ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Σχολή Θετικών Επιστημών και Τεχνολογίας

Συστήματα Κινητού και Διάχυτου Υπολογισμού (ΣΔΥ)

Διπλωματική Εργασία

Επεξηγήσιμη Τεχνητή Νοημοσύνη για Ανίχνευση Δικτυακών Επιθέσεων: Ανάπτυξη Εφαρμογής Κυβερνοασφάλειας με Επεξηγήσιμες Προβλέψεις

Ιωάννης Αλτανόπουλος, Ιωάννης Θεοδωρόπουλος

Επιβλέπων Καθηγητής: Δρ. Περικλής Χατζημίσιος



Shield IDS

Πάτρα, Ιούνιος 2026

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών (Ιωάννη Αλτανόπουλου, Ιωάννη Θεοδωρόπουλου) που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης οι συγγραφείς/δημιουργοί εκχωρούν στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οποιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας των συγγραφέων/δημιουργών ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέων/δημιουργών. Οι συγγραφείς/δημιουργοί διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



Επεξηγήσιμη Τεχνητή Νοημοσύνη για Ανίχνευση Δικτυακών Επιθέσεων: Ανάπτυξη Εφαρμογής Κυβερνοασφάλειας με Επεξηγήσιμες Προβλέψεις

Ιωάννης Αλτανόπουλος, Ιωάννης Θεοδωρόπουλος

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:
Δρ. Περικλής Χατζημίσιος
Καθηγητής, Διεθνές Πανεπιστήμιο της Ελλάδος (ΔΙ.ΠΑ.Ε.)

Συν-Επιβλέπων Καθηγητής:
Δρ. Κωνσταντίνος Χωριανόπουλος
Αναπληρωτής Καθηγητής, Ιόνιο Πανεπιστήμιο

Πάτρα, Ιούνιος 2026

«Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, θα θέλαμε να εκφράσουμε τις θερμές μας ευχαριστίες σε όσους συνέβαλαν, ο καθένας με τον δικό του τρόπο, στην πραγματοποίησή της.

Καταρχάς, οφείλουμε ένα μεγάλο ευχαριστώ στον επιβλέποντα καθηγητή μας, κ. Χατζημίσιο Περικλή, για την πολύτιμη καθοδήγηση, τις εύστοχες παρατηρήσεις και την εμπιστοσύνη που μας έδειξε καθ' όλη τη διάρκεια εκπόνησης της εργασίας. Οι συμβουλές και οι κατευθύνσεις του ήταν καθοριστικές για τη διαμόρφωση και την ολοκλήρωση τόσο του θεωρητικού όσο και του πρακτικού μέρους της διπλωματικής.

Θα θέλαμε επίσης να ευχαριστήσουμε τον κ. Χωριανόπουλο Κωνσταντίνο , για τις γνώσεις και τα εφόδια που μας παρείχε κατά την διάρκεια των σπουδών μας.

Ιδιαίτερες ευχαριστίες οφείλει ο καθένας μας στον άλλον διότι η συνεργασία μας κατά τη διάρκεια εκπόνησης αυτής της εργασίας βασίστηκε στην αμοιβαία στήριξη, την υπομονή και τον επιμερισμό της δουλειάς, στοιχεία που έκαναν το εγχείρημα αυτό πολύ πιο εφικτό και ουσιαστικό.

Τέλος, θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας, για την κατανόηση, την υπομονή και την ανιδιοτελή στήριξη που μας προσέφεραν σε όλη τη διάρκεια των σπουδών μας, ειδικά στις πιο απαιτητικές περιόδους της εκπόνησης της παρούσας εργασίας.»

Περίληψη

Η ραγδαία αύξηση των κυβερνοεπιθέσεων, σε συνδυασμό με τους περιορισμούς των παραδοσιακών συστημάτων ανίχνευσης βασισμένα σε υπογραφές (signature-based detection), καθιστά πλέον αναγκαία τη δημιουργία πιο έξυπνων και διαφανών λύσεων ασφαλείας. Η παρούσα διπλωματική εργασία παρουσιάζει το Shield IDS, ένα ολοκληρωμένο σύστημα ανίχνευσης εισβολών δικτύου, το οποίο συνδυάζει τη μηχανική μάθηση με την επεξηγήσιμη τεχνητή νοημοσύνη (Explainable Artificial Intelligence - XAI), προσφέροντας πλήρη λειτουργικότητα σε πραγματικές συνθήκες λειτουργίας του δικτύου.

Η μεθοδολογία ξεκινά με την αναλυτική διαδικασία προεπεξεργασίας των δεδομένων, η οποία περιλαμβάνει τη διαχείριση των διπλότυπων και ελλιπουσών (NaN) τιμών, την κωδικοποίηση των κατηγορικών μεταβλητών, την επιλογή των χαρακτηριστικών και την αφαίρεση των μη σημαντικών στηλών, ώστε να διασφαλιστεί ένα καθαρό και αξιόπιστο σύνολο (δεδομένων) εκπαίδευσης. Στη συνέχεια, εκπαιδεύεται ο ταξινομητής XGBoost, ο οποίος βελτιστοποιείται μέσω της μεθόδου RandomizedSearchCV, ενώ ιδιαίτερη έμφαση δίνεται επίσης στην αντιμετώπιση της ανισορροπίας των κλάσεων, εξετάζοντας τις τρεις τεχνικές εξισορρόπησης SMOTE, SMOTE-ENN και υβριδικής μεθόδου.

Η αξιολόγηση του συστήματος πραγματοποιήθηκε στα δύο αναγνωρισμένα σύνολα δεδομένων CIC-IDS2017 (το οποίο διαχωρίζεται στα υποσύνολα DoS, Web Attacks και PortScan) και UNSW-NB15. Συνολικά, μελετήθηκαν 21 διαφορετικοί συνδυασμοί δεδομένων και τεχνικών εξισορρόπησης. Τα αποτελέσματα έδειξαν εξαιρετική απόδοση, η οποία ισοφαρίζει ή ακόμα και ξεπερνά πιο περίπλοκα συστήματα που έχουν δημοσιευτεί στη σχετική βιβλιογραφία. Παράλληλα, επιβεβαιώθηκε ότι δεν υπάρχει μία καθολικά βέλτιστη τεχνική εξισορρόπησης, καθώς η αποτελεσματικότητα της καθεμίας τεχνικής εξαρτάται άμεσα από τα χαρακτηριστικά του εκάστοτε συνόλου δεδομένων.

Για την ενίσχυση της διαφάνειας, ενσωματώθηκε σε όλη τη ροή της εφαρμογής η μέθοδος προσθήκης εύστοχων επεξηγήσεων (SHapley Additive exPlanations - SHAP), με την οποία επιτρέπεται η ερμηνεία των αποφάσεων του μοντέλου. Παράλληλα, οι ανιχνευόμενες επιθέσεις αντιστοιχίζονται αυτόματα στα δύο βασικά επίπεδα διαχωρισμού (τεχνικές και τακτικές) της παγκόσμιας βάσης γνώσεων MITRE ATT&CK.

Το Shield IDS υλοποιήθηκε ως μια ολοκληρωμένη εφαρμογή με γραφικό περιβάλλον χρήστη (Graphical User Interface - GUI) που υποστηρίζει τον πλήρη κύκλο ζωής ενός

συστήματος ανίχνευσης εισβολών δικτύου IDS. Ξεκινώντας από την προεπεξεργασία και την εκπαίδευση, μέχρι τη ανίχνευση σε πραγματικό χρόνο, τον έλεγχο IP διευθύνσεων, την ανάλυση καταγραφών (logs) και την αυτοματοποιημένη δημιουργία αναφορών, διαθέτοντας παράλληλα ένα πίνακα αποτελεσμάτων, απευθυνόμενο στα στελέχη της εταιρείας για τη παρακολούθηση των βασικών στρατηγικών δεικτών. Οι ελεγχόμενες δοκιμές επίθεσης σε απομονωμένο τοπικό δικτυακό περιβάλλον επιβεβαίωσαν την αξιοπιστία και την ετοιμότητα του συστήματος σε πραγματικές συνθήκες.

Explainable Artificial Intelligence for Network Intrusion Detections: Development of a Cybersecurity Application with Explainable Predictions

Ioannis Altanopoulos, Ioannis Theodoropoulos

Abstract

The rapid increase in cyberattacks, combined with the limitations of traditional signature-based detection systems, has made the development of more intelligent and transparent security solutions a necessity. This thesis presents Shield IDS, a comprehensive network intrusion detection system that combines machine learning with Explainable Artificial Intelligence (XAI), offering full functionality under real network operational conditions.

The methodology begins with a detailed data preprocessing procedure, which includes handling duplicate and missing (NaN) values, encoding categorical variables, performing feature selection, and removing non-significant columns, in order to ensure a clean and reliable training dataset. Subsequently, an XGBoost classifier is trained and optimized using the RandomizedSearchCV method, with particular emphasis on addressing class imbalance by evaluating three balancing techniques: SMOTE, SMOTE-ENN, and a hybrid method.

The system was evaluated on two well-established datasets: CIC-IDS2017 (divided into DoS, Web Attacks, and PortScan subsets) and UNSW-NB15. In total, 21 different combinations of datasets and balancing techniques were studied. The results demonstrated excellent performance, matching or even surpassing those of more complex systems reported in the related literature. At the same time, it was confirmed that there is no universally optimal balancing technique, as the effectiveness of each one depends directly on the characteristics of the respective dataset.

To enhance transparency, the SHapley Additive exPlanations (SHAP) method was integrated throughout the entire application flow, enabling the interpretation of the model's decisions. In parallel, detected attacks are automatically mapped to the two main levels of categorization (techniques and tactics) of the global MITRE ATT&CK knowledge base.

Shield IDS was implemented as a complete application with a graphical user interface (GUI) that supports the full lifecycle of an IDS. Starting with preprocessing and training, through to real-time detection, IP address control, log analysis, and automated report generation, it also features a results dashboard for company executives to monitor key strategic indicators. Controlled attack tests in an isolated local network environment confirmed the system's reliability and readiness under real-world conditions.

Keywords

Intrusion Detection System, Machine Learning, XGBoost, Explainable AI (XAI), SHAP, Class Imbalance, cic-ids2017, UNSW-NB15

Περιεχόμενα

Περίληψη v
Abstract vii
Περιεχόμενα ix
Κατάλογος Εικόνων / Σχημάτων xii
Κατάλογος Πινάκων xxi
Συντομογραφίες & Ακρωνύμια xxii
1. Εισαγωγή 1
1.1. Γενικό Πλαίσιο 1
1.2. Αντικείμενο και Στόχοι της Εργασίας 2
1.3. Δομή της Διπλωματικής Εργασίας 3
2. Θεωρητικό υπόβαθρο 7
2.1. Συστήματα Ανίχνευσης Εισβολών (IDS) 7
2.1.1. Ορισμός και σκοπός των συστημάτων ανίχνευσης εισβολών. 7
2.1.2. Κυβερνοαπειλές και Επιθέσεις 9
2.2. Μηχανική Μάθηση στην Κυβερνοασφάλεια 12
2.2.1. Εισαγωγή στη Μηχανική Μάθηση 12
2.2.2. Θεωρία των δέντρων απόφασης και της ομαδικής μάθησης 15
2.2.3. Ο αλγόριθμος XGBoost 18
2.2.4. Πλεονεκτήματα του XGBoost 20
3. Σύνολο δεδομένων 22
3.1. Περιγραφή του συνόλου δεδομένων 23
3.2. Ανάλυση χαρακτηριστικών 25
3.3. Αφαίρεση μη σημαντικών στηλών 31
3.3.1. Διαδικασία και τεχνική αφαίρεσης 33
4. Προεπεξεργασία Δεδομένων 37
4.1. Φόρτωση του αρχικού συνόλου δεδομένων 42
4.2. Διαχείριση διπλότυπων εγγραφών 46
4.3. Διαχείριση ελλιπουσών τιμών (NaN) 48
4.4. Κωδικοποίηση κατηγορικών μεταβλητών 54
4.5. Επιλογή χαρακτηριστικών (feature selection) 57
4.6. Δημιουργία και αποθήκευση τελικού επεξεργασμένου συνόλου δεδομένων 63
5. Εκπαίδευση, αξιολόγηση κι επεξηγησιμότητα μοντέλου 65
5.1. Διαχωρισμός δεδομένων σε training και test σύνολο 69
5.2. Το πρόβλημα του "Data Imbalance", διαχείριση ανισορροπίας κλάσεων 74
5.2.1. Τεχνική SMOTE 75
5.2.2. Τεχνική SMOTEENN 80
5.2.3. Τεχνική Hybrid (Under sampling + SMOTE + ENN) 84
5.3. Βελτιστοποίηση και εκπαίδευση 88
5.3.1. Καθορισμός αντικειμενικής συνάρτησης: Διάκριση μεταξύ δυαδικής και πολυκλασικής ταξινόμησης 89
5.3.2. RandomizedSearchCV: Γιατί επιλέχθηκε αντί του GridSearchCV - Αναφορά υπέρ παραμέτρων 91
6. Αξιολόγηση απόδοσης 96

6.1. Μετρικές αξιολόγησης 96
6.2. Απεικόνιση αποτελεσμάτων 99
6.3. Επεξηγησιμότητα μοντέλου (XAI) 120
6.3.1. Ανάλυση σημαντικότητας χαρακτηριστικών 123
6.3.2. Ερμηνεία προβλέψεων 144
6.4. Συγκριτική ανάλυση αποτελεσμάτων εκπαίδευσης 221
6.4.1. Σύγκριση μετρικών με τη βιβλιογραφία 221
7. Κύρια Εφαρμογή και Λειτουργικές Ενότητες 227
7.1. Λειτουργίες μενού: Φόρτωση μοντέλων και έλεγχος σύλληψης 230
7.2. Executive Dashboard 232
7.2.1. Λειτουργίες φιλτραρίσματος και ενημέρωσης του Executive Dashboard 233
7.2.2. Στρατηγικοί δείκτες (KPIs) 234
7.2.3. Μοντέλο λήψης αποφάσεων: Παροχή προτάσεων βάσει της σοβαρότητας των επιθέσεων 241
7.2.4. Δημιουργία Εκτελεστικής Αναφοράς Ασφαλείας (Executive Security Report) 242
7.3. Κεντρικό dashboard 243
7.4. Live detection και παρακολούθηση σε πραγματικό χρόνο 255
7.5. Περιβάλλον του Explainability 260
7.5.1. Σύνοψη περιστατικού στο XAI περιβάλλον 262
7.5.2. Ερμηνεία χαρακτηριστικών μέσω SHAP και προφίλ ανά τύπο επίθεσης 263
7.5.3. Διασύνδεση με το MITRE ATT&CK: Αντιστοίχιση των ανιχνευμένων επιθέσεων με επίσημες τεχνικές και τακτικές κυβερνοεπιθέσεων 268
7.5.4. Σύγκριση Μοντέλων: Απεικόνιση της συμφωνίας/διαφωνίας μεταξύ CIC-IDS και UNSW-NB15 μοντέλων 270
7.6. Δημιουργία και διαχείριση αναφορών (Alerts & Reports) 272
7.7. Ανάλυση καταγραφών (Logs Analysis) 277
7.8. Έλεγχος και διαχείριση IP (IP Control) 282
7.8.1. Learning Mode: Αυτόματη δημιουργία whitelist μέσω παρακολούθησης της κίνησης 284
7.8.2. Διαχείριση Μαύρης/Λευκής Λίστας (blacklist/whitelist) για έλεγχο πρόσβασης IP 285
8. Αρχιτεκτονική και ανάπτυξη εφαρμογής 288
8.1. Συλλογή κι επεξεργασία δεδομένων 289
8.1.1. Ανάλυση βασικών βιβλιοθηκών 292
8.1.2. Μηχανισμός multi-threading - Χρήση του FastSniffThread για τη συλλογή πακέτων 295
8.1.3. Μοντέλο Δεδομένων: Χρήση του PandasModel για τη σύνδεση των DataFrames με τα UI Tables 297
8.2. Μονάδα Συλλογής και Επεξεργασίας Ροών (flow aggregator) 299
8.2.1. Διαδικασία packet capturing, η χρήση της βιβλιοθήκης pcapy για απευθείας πρόσβαση στην κάρτα δικτύου 300
8.2.2. Εξαγωγή και αντιστοίχιση χαρακτηριστικών: Πώς ο συσσωρευτής ροών μετατρέπει τα πακέτα σε χαρακτηριστικά (CIC και UNSW features) σε πραγματικό χρόνο και πως τροφοδοτούνται τα μοντέλα. 303
8.2.3. Διάγραμμα ροής δεδομένων 310

8.3. Πρόβλεψη & διάγνωση: Αναγνώριση μοτίβων 313
8.4. Υλοποίηση Λειτουργιών Εφαρμογής 319
8.4.1. Σχεδιασμός και Υλοποίηση Μενού 319
8.4.2. Executive Dashboard 327
8.4.3. Κεντρικό Dashboard 339
8.4.4. Live Detection και παρακολούθηση σε πραγματικό χρόνο 357
8.4.5. Explainability περιβάλλον 362
8.4.6. Δημιουργία και διαχείριση αναφορών (Alerts & Reports) 372
8.4.7. Ανάλυση καταγραφών (Logs Analysis) 377
8.4.8. Έλεγχος και διαχείριση IP (IP Control) 387
9. Κατανομή Εργασιών, Συμπεράσματα και Μελλοντικές Επεκτάσεις 395
9.1. Κατανομή Εργασιών 395
9.2. Συμπεράσματα 398
9.3. Μελλοντικές Επεκτάσεις 399
Βιβλιογραφία 401

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1 Κεντρική αρχική οθόνη Shield IDS 33
Εικόνα 2 Αρχική εφαρμογή και εισαγωγή συνόλου δεδομένων....34
Εικόνα 3 Επιλογή συνόλου δεδομένων για προεπεξεργασία. 42
Εικόνα 4 Επιλογή της ενότητας “Preprocessing Dataset” από το κεντρικό μενού για την έναρξη της διαδικασίας προεπεξεργασίαςς. 45
Εικόνα 5 Παράθυρο εντοπισμού στηλών που αντιστοιχούν σε διευθύνσεις IP ή αριθμούς θυρών (ports), με δυνατότητα επιλογής των στηλών που θα διαγραφούν πριν από την περαιτέρω προεπεξεργασίας. 45
Εικόνα 6 Μήνυμα επιβεβαίωσης της επιτυχούς αφαίρεσης στηλών. 45
Εικόνα 7 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset Wednesday-workingHours- Dos_cleaned....47
Εικόνα 8 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned 47
Εικόνα 9 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned....47
Εικόνα 10 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset UNSW-NB15_1 47
Εικόνα 11 Παράθυρο διαχείρισης NaN τιμών και επιλογής μεθόδου καθαρισμού από το dataset Wednesday-workingHours- Dos_cleaned....52
Εικόνα 12 Αποτελέσματα καθαρισμού NaN τιμών με διαγραφή γραμμών και με αντικατάσταση μέσω μέσου όρου ανά κλάση από το dataset Wednesday-workingHours-Dos_cleaned 52
Εικόνα 13 Παράθυρο διαχείρισης NaN τιμών και επιλογής μεθόδου καθαρισμού από το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned....53
Εικόνα 14 Αποτελέσματα καθαρισμού NaN τιμών με διαγραφή γραμμών και με αντικατάσταση μέσω μέσου όρου ανά κλάση από το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned 53
Εικόνα 15 Παράθυρο διαχείρισης NaN τιμών και επιλογής μεθόδου καθαρισμού από το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned 53
Εικόνα 16 Αποτελέσματα καθαρισμού NaN τιμών με διαγραφή γραμμών και με αντικατάσταση μέσω μέσου όρου ανά κλάση από το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned 53
Εικόνα 17 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset Wednesday-workingHours- Dos_cleaned....56
Εικόνα 18 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned....56
Εικόνα 19 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned 56
Εικόνα 20 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset UNSW-NB15_1 57
Εικόνα 21 Επιλογή της στήλης στόχου για το dataset Wednesday-workingHours-Dos_cleaned. 59

Εικόνα 22 Επιλεγμένα χαρακτηριστικά για το dataset Wednesday-workingHours-Dos_cleaned μετά τη διαδικασία feature selection με XGBoost για τις δύο μεθόδους διαχείρισης NaN τιμών 60
Εικόνα 23 Επιλογή της στήλης στόχου για το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned. 60
Εικόνα 24 Επιλεγμένα χαρακτηριστικά για το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned μετά τη διαδικασία feature selection με XGBoost για τις δύο μεθόδους διαχείρισης NaN τιμών 61
Εικόνα 25 Επιλογή της στήλης στόχου για το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned. 61
Εικόνα 26 Επιλεγμένα χαρακτηριστικά για το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned μετά τη διαδικασία feature selection με XGBoost για τις δύο μεθόδους διαχείρισης NaN τιμών 62
Εικόνα 27 Επιλογή της στήλης στόχου για το dataset UNSW-NB15_1 62
Εικόνα 28 Επιλεγμένα χαρακτηριστικά για το dataset UNSW-NB15_1 63
Εικόνα 29 Αρχική οθόνη εφαρμογής εκπαίδευσης 69
Εικόνα 30 Επιλογή στήλης στόχου για την εκκίνηση της εκπαίδευσης 73
Εικόνα 31 Παράθυρο για την ενημέρωση της προόδου της διαδικασίας 74
Εικόνα 32 Επιλογή της τεχνικής SMOTE από το γραφικό περιβάλλον 77
Εικόνα 33 1ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις. 77
Εικόνα 34 2ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE. Κατανομή της κλάσεων πριν και μετά το resampling. 78
Εικόνα 35 3ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 4 κλάσεις. 78
Εικόνα 36 10ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Smote. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις 79
Εικόνα 37 11ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE. Κατανομή της κλάσεων πριν και μετά το resampling. 79
Εικόνα 38 12ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling. 79
Εικόνα 39 19ος Συνδυασμός: Dataset UNSW-NB15_1.csv με μέθοδο εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling. 80
Εικόνα 40 4ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις. 81
Εικόνα 41 5ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή της κλάσεων πριν και μετά το resampling. 82

Εικόνα 42 6ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling. 82
Εικόνα 43 13ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις.....82
Εικόνα 44 14ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή της κλάσεων πριν και μετά το resampling. 83
Εικόνα 45 15ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling.....83
Εικόνα 46 20ος Συνδυασμός: Dataset UNSW-NB15_1.csv με μέθοδο εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling.....83
Εικόνα 47 7ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις.....86
Εικόνα 48 8ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης hybrid. Κατανομή της κλάσεων πριν και μετά το resampling. 86
Εικόνα 49 9ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling.....87
Εικόνα 50 16ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις. 87
Εικόνα 51 17ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Hybrid. Κατανομή της κλάσεων πριν και μετά το resampling. 87
Εικόνα 52 18ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling. 88
Εικόνα 53 21ος Συνδυασμός: Dataset UNSW-NB15_1.csv με μέθοδο εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling. 88
Εικόνα 54 Εντοπισμός δύο κλάσεων - Δυαδική ταξινόμηση και εύρεση βέλτιστων παραμέτρων.....95
Εικόνα 55 Εντοπισμός έξι κλάσεων - Πολυκλασική ταξινόμηση και εύρεση βέλτιστων παραμέτρων.....95
Εικόνα 56 Πίνακας σύγχυσης για τον 1ο συνδυασμό. 104
Εικόνα 57 Πίνακας σύγχυσης για τον 2ο συνδυασμό. 104
Εικόνα 58 Πίνακας σύγχυσης για τον 3ο συνδυασμό. 105
Εικόνα 59 Πίνακας σύγχυσης για τον 4ο συνδυασμό. 106
Εικόνα 60 Πίνακας σύγχυσης για τον 5ο συνδυασμό. 107
Εικόνα 61 Πίνακας σύγχυσης για τον 6ο συνδυασμό. 107
Εικόνα 62 Πίνακας σύγχυσης για τον 7ο συνδυασμό. 108
Εικόνα 63 Πίνακας σύγχυσης για τον 8ο συνδυασμό. 109
Εικόνα 64 Πίνακας σύγχυσης για τον 9ο συνδυασμό. 110

Εικόνα 65 Πίνακας σύγχυσης για τον 10ο συνδυασμό. 111
Εικόνα 66 Πίνακας σύγχυσης για τον 11ο συνδυασμό. 111
Εικόνα 67 Πίνακας σύγχυσης για τον 12ο συνδυασμό. 112
Εικόνα 68 Πίνακας σύγχυσης για τον 13ο συνδυασμό. 113
Εικόνα 69 Πίνακας σύγχυσης για τον 14ο συνδυασμό. 114
Εικόνα 70 Πίνακας σύγχυσης για τον 15ο συνδυασμό. 114
Εικόνα 71 Πίνακας σύγχυσης για τον 16ο συνδυασμό. 116
Εικόνα 72 Πίνακας σύγχυσης για τον 17ο συνδυασμό. 116
Εικόνα 73 Πίνακας σύγχυσης για τον 18ο συνδυασμό. 117
Εικόνα 74 Πίνακας σύγχυσης για τον 19ο συνδυασμό. 118
Εικόνα 75 Πίνακας σύγχυσης για τον 20ο συνδυασμό. 119
Εικόνα 76 Πίνακας σύγχυσης για τον 21ο συνδυασμό. 120
Εικόνα 77 Διάγραμμα Bar Plot για τον 1ο συνδυασμό. 124
Εικόνα 78 Διάγραμμα Bar Plot για τον 2ο συνδυασμό. 125
Εικόνα 79 Διάγραμμα Bar Plot για τον 3ο συνδυασμό. 126
Εικόνα 80 Διάγραμμα Bar Plot για τον 4ο συνδυασμό. 127
Εικόνα 81 Διάγραμμα Bar Plot για τον 5ο συνδυασμό. 128
Εικόνα 82 Διάγραμμα Bar Plot για τον 6ο συνδυασμό. 129
Εικόνα 83 Διάγραμμα Bar Plot για τον 7ο συνδυασμό. 130
Εικόνα 84 Διάγραμμα Bar Plot για τον 8ο συνδυασμό. 131
Εικόνα 85 Διάγραμμα Bar Plot για τον 9ο συνδυασμό. 132
Εικόνα 86 Διάγραμμα Bar Plot για τον 10ο συνδυασμό. 133
Εικόνα 87 Διάγραμμα Bar Plot για τον 11ο συνδυασμό. 134
Εικόνα 88 Διάγραμμα Bar Plot για τον 12ο συνδυασμό. 135
Εικόνα 89 Διάγραμμα Bar Plot για τον 13ο συνδυασμό. 136
Εικόνα 90 Διάγραμμα Bar Plot για τον 14ο συνδυασμό. 137
Εικόνα 91 Διάγραμμα Bar Plot για τον 15ο συνδυασμό. 138
Εικόνα 92 Διάγραμμα Bar Plot για τον 16ο συνδυασμό. 139
Εικόνα 93 Διάγραμμα Bar Plot για τον 17ο συνδυασμό. 140
Εικόνα 94 Διάγραμμα Bar Plot για τον 18ο συνδυασμό. 141
Εικόνα 95 Διάγραμμα Bart Plot για τον 19ο συνδυασμό. 142
Εικόνα 96 Διάγραμμα Bar Plot για τον 20ο συνδυασμό. 143
Εικόνα 97 Διάγραμμα Bar Plot για τον 21ο συνδυασμό. 144
Εικόνα 98 SHAP Summary Plot, όλες οι κλάσεις για τον 1ο συνδυασμό. 147
Εικόνα 99 SHAP Summary Plot, κλάση 0 για τον 1ο συνδυασμό. 147
Εικόνα 100 SHAP Summary Plot, κλάση 1 για τον 1ο συνδυασμό. 147
Εικόνα 101 SHAP Summary Plot, κλάση 2 για τον 1ο συνδυασμό. 147
Εικόνα 102 SHAP Summary Plot, κλάση 3 για τον 1ο συνδυασμό. 148
Εικόνα 103 SHAP Summary Plot, κλάση 4 για τον 1ο συνδυασμό. 148
Εικόνα 104 SHAP Summary Plot, κλάση 5 για τον 1ο συνδυασμό. 148
Εικόνα 105 SHAP Summary Plot, όλες οι κλάσεις για τον 2ο συνδυασμό. 149
Εικόνα 106 SHAP Summary Plot, όλες οι κλάσεις για τον 3ο συνδυασμό. 151
Εικόνα 107 SHAP Summary Plot, κλάση 0 για τον 3ο συνδυασμό. 151
Εικόνα 108 SHAP Summary Plot, κλάση 1 για τον 3ο συνδυασμό. 152
Εικόνα 109 SHAP Summary Plot, κλάση 2 για τον 3ο συνδυασμό. 152
Εικόνα 110 SHAP Summary Plot, κλάση 3 για τον 3ο συνδυασμό. 152

Εικόνα 111 SHAP Summary Plot, όλες οι κλάσεις για τον 4ο συνδυασμό. 154
Εικόνα 112 SHAP Summary Plot, κλάση 0 για τον 4ο συνδυασμό. 154
Εικόνα 113 SHAP Summary Plot, κλάση 1 για τον 4ο συνδυασμό. 155
Εικόνα 114 SHAP Summary Plot, κλάση 2 για τον 4ο συνδυασμό. 155
Εικόνα 115 SHAP Summary Plot, κλάση 3 για τον 4ο συνδυασμό. 155
Εικόνα 116 SHAP Summary Plot, κλάση 4 για τον 4ο συνδυασμό. 155
Εικόνα 117 SHAP Summary Plot, κλάση 5 για τον 4ο συνδυασμό. 156
Εικόνα 118 SHAP Summary Plot, όλες οι κλάσεις για τον 5ο συνδυασμό. 158
Εικόνα 119 SHAP Summary Plot, κλάση 0 για τον 5ο συνδυασμό. 158
Εικόνα 120 SHAP Summary Plot, κλάση 1 για τον 5ο συνδυασμό. 158
Εικόνα 121 SHAP Summary Plot, κλάση 2 για τον 5ο συνδυασμό. 158
Εικόνα 122 SHAP Summary Plot, κλάση 3 για τον 5ο συνδυασμό. 159
Εικόνα 123 SHAP Summary Plot, όλες οι κλάσεις για τον 6ο συνδυασμό. 160
Εικόνα 124 SHAP Summary Plot, όλες οι κλάσεις για τον 7ο συνδυασμό. 163
Εικόνα 125 SHAP Summary Plot, κλάση 0 για τον 7ο συνδυασμό. 163
Εικόνα 126 SHAP Summary Plot, κλάση 1 για τον 7ο συνδυασμό. 163
Εικόνα 127 SHAP Summary Plot, κλάση 2 για τον 7ο συνδυασμό. 163
Εικόνα 128 SHAP Summary Plot, κλάση 3 για τον 7ο συνδυασμό. 164
Εικόνα 129 SHAP Summary Plot, κλάση 4 για τον 7ο συνδυασμό. 164
Εικόνα 130 SHAP Summary Plot, κλάση 5 για τον 7ο συνδυασμό. 164
Εικόνα 131 SHAP Summary Plot, όλες οι κλάσεις για τον 8ο συνδυασμό. 167
Εικόνα 132 SHAP Summary Plot, κλάση 0 για τον 8ο συνδυασμό. 167
Εικόνα 133 SHAP Summary Plot, κλάση 1 για τον 8ο συνδυασμό. 167
Εικόνα 134 SHAP Summary Plot, κλάση 2 για τον 8ο συνδυασμό. 167
Εικόνα 135 SHAP Summary Plot, κλάση 3 για τον 8ο συνδυασμό. 168
Εικόνα 136 SHAP Summary Plot, όλες οι κλάσεις για τον 9ο συνδυασμό. 169
Εικόνα 137 Summary Plot, όλες οι κλάσεις για τον 10ο συνδυασμό.....173
Εικόνα 138 SHAP Summary Plot, κλάση 0 για τον 10ο συνδυασμό. 173
Εικόνα 139 SHAP Summary Plot, κλάση 1 για τον 10ο συνδυασμό. 173
Εικόνα 140 SHAP Summary Plot, κλάση 2 για τον 10ο συνδυασμό. 173
Εικόνα 141 SHAP Summary Plot, κλάση 3 για τον 10ο συνδυασμό. 174
Εικόνα 142 SHAP Summary Plot, κλάση 4 για τον 10ο συνδυασμό. 174
Εικόνα 143 SHAP Summary Plot, κλάση 5 για τον 10ο συνδυασμό. 174
Εικόνα 144 Summary Plot, όλες οι κλάσεις για τον 11ο συνδυασμό.....177
Εικόνα 145 SHAP Summary Plot, κλάση 0 για τον 11ο συνδυασμό. 177
Εικόνα 146 SHAP Summary Plot, κλάση 1 για τον 11ο συνδυασμό. 177
Εικόνα 147 SHAP Summary Plot, κλάση 2 για τον 11ο συνδυασμό. 177
Εικόνα 148 SHAP Summary Plot, κλάση 3 για τον 11ο συνδυασμό. 178
Εικόνα 149 Summary Plot, όλες οι κλάσεις για τον 12ο συνδυασμό.....179
Εικόνα 150 Summary Plot, όλες οι κλάσεις για τον 13ο συνδυασμό.....183
Εικόνα 151 SHAP Summary Plot, κλάση 0 για τον 13ο συνδυασμό. 183
Εικόνα 152 SHAP Summary Plot, κλάση 1 για τον 13ο συνδυασμό. 183
Εικόνα 153 SHAP Summary Plot, κλάση 2 για τον 13ο συνδυασμό. 183
Εικόνα 154 SHAP Summary Plot, κλάση 3 για τον 13ο συνδυασμό. 184
Εικόνα 155 SHAP Summary Plot, κλάση 4 για τον 13ο συνδυασμό. 184
Εικόνα 156 SHAP Summary Plot, κλάση 5 για τον 13ο συνδυασμό. 184

Εικόνα 157 Summary Plot, όλες οι κλάσεις για τον 14ο συνδυασμό.....187
Εικόνα 158 SHAP Summary Plot, κλάση 0 για τον 14ο συνδυασμό.....187
Εικόνα 159 SHAP Summary Plot, κλάση 1 για τον 14ο συνδυασμό.....187
Εικόνα 160 SHAP Summary Plot, κλάση 2 για τον 14ο συνδυασμό.....187
Εικόνα 161 SHAP Summary Plot, κλάση 3 για τον 14ο συνδυασμό.....188
Εικόνα 162 Summary Plot, όλες οι κλάσεις για τον 15ο συνδυασμό.....189
Εικόνα 163 Summary Plot, όλες οι κλάσεις για τον 16ο συνδυασμό.....193
Εικόνα 164 SHAP Summary Plot, κλάση 0 για τον 16ο συνδυασμό.....193
Εικόνα 165 SHAP Summary Plot, κλάση 1 για τον 16ο συνδυασμό.....193
Εικόνα 166 SHAP Summary Plot, κλάση 2 για τον 16ο συνδυασμό.....193
Εικόνα 167 SHAP Summary Plot, κλάση 3 για τον 16ο συνδυασμό.....194
Εικόνα 168 SHAP Summary Plot, κλάση 4 για τον 16ο συνδυασμό.....194
Εικόνα 169 SHAP Summary Plot, κλάση 5 για τον 16ο συνδυασμό.....194
Εικόνα 170 Summary Plot, όλες οι κλάσεις για τον 17ο συνδυασμό.....197
Εικόνα 171 SHAP Summary Plot, κλάση 0 για τον 17ο συνδυασμό.....197
Εικόνα 172 SHAP Summary Plot, κλάση 1 για τον 17ο συνδυασμό.....197
Εικόνα 173 SHAP Summary Plot, κλάση 2 για τον 17ο συνδυασμό.....197
Εικόνα 174 SHAP Summary Plot, κλάση 3 για τον 17ο συνδυασμό.....198
Εικόνα 175 Summary Plot, όλες οι κλάσεις για τον 18ο συνδυασμό.....199
Εικόνα 176 Summary Plot, όλες οι κλάσεις για τον 19ο συνδυασμό.....204
Εικόνα 177 SHAP Summary Plot, κλάση 0 για τον 19ο συνδυασμό.....204
Εικόνα 178 SHAP Summary Plot, κλάση 1 για τον 19ο συνδυασμό.....204
Εικόνα 179 SHAP Summary Plot, κλάση 2 για τον 19ο συνδυασμό.....204
Εικόνα 180 SHAP Summary Plot, κλάση 3 για τον 19ο συνδυασμό.....205
Εικόνα 181 SHAP Summary Plot, κλάση 4 για τον 19ο συνδυασμό.....205
Εικόνα 182 SHAP Summary Plot, κλάση 5 για τον 19ο συνδυασμό.....205
Εικόνα 183 SHAP Summary Plot, κλάση 6 για τον 19ο συνδυασμό.....205
Εικόνα 184 SHAP Summary Plot, κλάση 7 για τον 19ο συνδυασμό.....206
Εικόνα 185 SHAP Summary Plot, κλάση 8 για τον 19ο συνδυασμό.....206
Εικόνα 186 SHAP Summary Plot, κλάση 9 για τον 19ο συνδυασμό.....206
Εικόνα 187 Summary Plot, όλες οι κλάσεις για τον 20ο συνδυασμό.....211
Εικόνα 188 SHAP Summary Plot, κλάση 0 για τον 20ο συνδυασμό.....211
Εικόνα 189 SHAP Summary Plot, κλάση 1 για τον 20ο συνδυασμό.....211
Εικόνα 190 SHAP Summary Plot, κλάση 2 για τον 20ο συνδυασμό.....211
Εικόνα 191 SHAP Summary Plot, κλάση 3 για τον 20ο συνδυασμό.....212
Εικόνα 192 SHAP Summary Plot, κλάση 4 για τον 20ο συνδυασμό.....212
Εικόνα 193 SHAP Summary Plot, κλάση 5 για τον 20ο συνδυασμό.....212
Εικόνα 194 SHAP Summary Plot, κλάση 6 για τον 20ο συνδυασμό.....212
Εικόνα 195 SHAP Summary Plot, κλάση 7 για τον 20ο συνδυασμό.....213
Εικόνα 196 SHAP Summary Plot, κλάση 8 για τον 20ο συνδυασμό.....213
Εικόνα 197 SHAP Summary Plot, κλάση 9 για τον 20ο συνδυασμό.....213
Εικόνα 198 Summary Plot, όλες οι κλάσεις για τον 21ο συνδυασμό.....218
Εικόνα 199 SHAP Summary Plot, κλάση 0 για τον 21ο συνδυασμό.....218
Εικόνα 200 SHAP Summary Plot, κλάση 1 για τον 21ο συνδυασμό.....218
Εικόνα 201 SHAP Summary Plot, κλάση 2 για τον 21ο συνδυασμό.....218
Εικόνα 202 SHAP Summary Plot, κλάση 3 για τον 21ο συνδυασμό.....219

Εικόνα 203 SHAP Summary Plot, κλάση 4 για τον 21ο συνδυασμό. 219
Εικόνα 204 SHAP Summary Plot, κλάση 5 για τον 21ο συνδυασμό. 219
Εικόνα 205 SHAP Summary Plot, κλάση 6 για τον 21ο συνδυασμό. 219
Εικόνα 206 SHAP Summary Plot, κλάση 7 για τον 21ο συνδυασμό. 220
Εικόνα 207 SHAP Summary Plot, κλάση 8 για τον 21ο συνδυασμό. 220
Εικόνα 208 SHAP Summary Plot, κλάση 9 για τον 21ο συνδυασμό. 220
Εικόνα 209 Εφαρμογή Shield IDS 228
Εικόνα 210 Μενού φόρτωσης των μοντέλων 231
Εικόνα 211 Μενού εκκίνησης και παύσης της σύλληψης της δικτυακής κίνησης 231
Εικόνα 212 Οπτικοποίηση του Executive Dashboard 233
Εικόνα 213 Οπτικοποίηση του δείκτη CRI στο Executive Dashboard με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου 235
Εικόνα 214 Οπτικοποίηση του δείκτη Endpoint Threat Exposure με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου 237
Εικόνα 215 Οπτικοποίηση δείκτη MTTD με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου 238
Εικόνα 216 Οπτικοποίηση του δείκτη 7-Day Risk Forecast με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου 239
Εικόνα 217 Οπτικοποίηση των σημαντικών συμβάντων με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου 240
Εικόνα 218 Οπτικοποίηση των δράσεων 241
Εικόνα 219 Εκτελεστική Αναφορά Ασφαλείας σε μορφή PDF 242
Εικόνα 220 Κεντρικό Ταμπλό (Dashboard) 244
Εικόνα 221 Οπτικοποίηση ζωντανής ροής στο Dashboard 245
Εικόνα 222 Διάγραμμα πίτας για την κατανομή των πρωτοκόλλων 245
Εικόνα 223 Κάρτα Top Remote Hosts (Για λόγους προσωπικών δεδομένων οι IPs έχουν διαγραφεί) 247
Εικόνα 224 Κάρτα Alert Summary 247
Εικόνα 225 Αριστερά: Παρουσίαση του πιο πρόσφατου συμβάντος κακόβουλης δραστηριότητας στην κάρτα Recent Alerts, Δεξιά: Ενημέρωση ότι δεν υπάρχει ενεργή απειλή τη συγκεκριμένη χρονική στιγμή 248
Εικόνα 226 Αναδυόμενο παράθυρο με αναλυτικές πληροφορίες για την κακόβουλη δραστηριότητα 249
Εικόνα 227 Αριστερά: Σύντομη περιγραφή των παραγόντων που συνέβαλαν στην απόφαση Δεξιά: Ενημέρωση του συστήματος ότι η κίνηση είναι φυσιολογική 250
Εικόνα 228 Απεικόνιση της κάρτας Network Activity 251
Εικόνα 229 Εμφάνιση του συνολικού αριθμού των καταγεγραμμένων συμβάντων στην κάρτα Logs του dashboard 251
Εικόνα 230 Κάρτα System Usage που εμφανίζει σε πραγματικό χρόνο τη χρήση των πόρων του συστήματος (CPU, μνήμη RAM και δραστηριότητα δίσκου) 253
Εικόνα 231 Απεικόνιση της κάρτας Emergency Control, η οποία παρέχει δυνατότητα άμεσης απομόνωσης και επαναφοράς του συστήματος μαζί με το μήνυμα επιβεβαίωσης. 253
Εικόνα 232 Απεικόνιση των καρτών Detected Attacks, οι οποίες παρουσιάζουν συγκεντρωτικά τον αριθμό των ανιχνευμένων επιθέσεων ανά κατηγορία 254

Εικόνα 233 Αριστερά: Οπτικοποίηση Detection latency , Δεξιά: Πληροφοριακό παράθυρο της κάρτας Detection Latency, το οποίο επεξηγεί τον τρόπο υπολογισμού της καθυστέρησης ανίχνευσης μιας επίθεσης 255
Εικόνα 234 Καρτέλα Live Detection. Εμφάνιση ροών με κανονική και κακόβουλη κίνηση 260
Εικόνα 235 Περιβάλλον Επεξηγησιμότητας.....261
Εικόνα 236 Εμφάνιση σύνοψης όταν η δικτυακή κίνηση είναι κανονική.....262
Εικόνα 237 Εμφάνιση σύνοψης όταν η ροή έχει χαρακτηριστεί ως κακόβουλη.....262
Εικόνα 238 Απεικόνιση τιμών Shap για την επίθεση τύπου Exploits μέσω ραβδογράμματος 265
Εικόνα 239 Απεικόνιση τιμών Shap για την επίθεση τύπου Dos μέσω ραβδογράμματος 265
Εικόνα 240 Απεικόνιση τιμών Shap για την επίθεση τύπου Exploits για το UNSW-NB15 και τύπου Dos για το CIC-IDS2017.....265
Εικόνα 241 Απεικόνιση τιμών Shap για την επίθεση τύπου Port Scan μέσω ραβδογράμματος 265
Εικόνα 242 Απεικόνιση τιμών XAI προφίλ για την επίθεση τύπου Generic μέσω ραβδογράμματος 265
Εικόνα 243 Απεικόνιση τιμών Shap για την επίθεση τύπου Backdoor μέσω ραβδογράμματος 265
Εικόνα 244 Απεικόνιση τιμών Shap για την επίθεση τύπου Fuzzers μέσω ραβδογράμματος 265
Εικόνα 245 Απεικόνιση τιμών Shap για την επίθεση τύπου Reconnaissance μέσω ραβδογράμματος 266
Εικόνα 246 Απεικόνιση τιμών Shap για την επίθεση τύπου Web Attack μέσω ραβδογράμματος 266
Εικόνα 247 Εμφάνιση τιμών Shap για την επίθεση τύπου Exploits. 266
Εικόνα 248 Εμφάνιση τιμών Shap για την επίθεση τύπου Dos.....267
Εικόνα 249 Εμφάνιση τιμών Shap για την επίθεση τύπου Exploits για το UNSW-NB15 και τύπου Dos για το CIC-IDS2017.....267
Εικόνα 250 Εμφάνιση τιμών Shap για την επίθεση τύπου PortScan. 267
Εικόνα 251 Εμφάνιση τιμών XAI προφίλ για την επίθεση τύπου Generic. 267
Εικόνα 252 Εμφάνιση τιμών Shap για την επίθεση τύπου Backdoor. 267
Εικόνα 253 Εμφάνιση τιμών Shap για την επίθεση τύπου Fuzzers. 268
Εικόνα 254 Εμφάνιση τιμών Shap για την επίθεση τύπου Reconnaissance. 268
Εικόνα 255 Εμφάνιση τιμών Shap για την επίθεση τύπου Web Attack 268
Εικόνα 256 MITRE ATT&CK περιεχόμενο για ανιχνευθείσα επίθεση τύπου Exploits. .269
Εικόνα 257 Περιγραφή της επίθεσης Exploits για αρχάριους χρήστες και αντιστοίχισή της με το MITRE ATT&CK. 270
Εικόνα 258 Παράδειγμα διαφωνίας μεταξύ των μοντέλων CIC-IDS2017 και UNSW-NB15 272
Εικόνα 259 Απεικόνιση όταν η κίνηση είναι φυσιολογική 272
Εικόνα 260 Απεικόνιση όταν έχουμε διαφορετικό είδος ανίχνευσης επίθεσης.....272
Εικόνα 261 Καρτέλα Alerts & Reports.....276
Εικόνα 262 Ενότητα Logs Analysis. Στις φωτογραφίες εμφανίζονται όλα τα χαρακτηριστικά του Πίνακα.....281
Εικόνα 263 Παράδειγμα εξαγόμενου αρχείου Excel από την εφαρμογή 281

Εικόνα 264 Διεπαφή χρήστη IP Control 282
Εικόνα 265 Απεικόνιση λειτουργίας εκμάθησης όταν είναι ανενεργή και όταν είναι ενεργοποιημένη 285
Εικόνα 266 Απεικόνιση της λευκής και μαύρης λίστας καθώς και τον κουμπιών αφαίρεσης IP διευθύνσεων 287
Εικόνα 267 Μπάρα αναζήτησης καταχωρημένων διεύθυνσεων IP 287
Εικόνα 268 Κουμπιά καταχώρησης μια διεύθυνσης IP είτε στην λευκή λίστα είτε στην μαύρη λίστα 287

Σχήμα 1 Διάγραμμα ροής επεξεργασίας δικτυακής κίνησης για ανίχνευση κακόβουλων ροών 311

Κατάλογος Πινάκων

Πίνακας 1 Ανάλυση χαρακτηριστικών των cicids 2017 και unsbw 15 30
Πίνακας 2 Χαρακτηριστικά που αφαιρέθηκαν από το CIC-IDS2017 και αιτιολόγηση αφαίρεσης. 32
Πίνακας 3 Συνοπτικός πίνακας μεθόδων προεπεξεργασίας. 40
Πίνακας 4 Διπλότυπες εγγραφές ανά σύνολο και υποσύνολο δεδομένων 47
Πίνακας 5 Σύνολο δυνατών συνδυασμών (προεπεξεργασία - εκπαίδευση) 68
Πίνακας 6 Συγκεντρωτικά αποτελέσματα μετρικών για κάθε συνδυασμό. 101
Πίνακας 7 Σύγκριση Friday-WorkingHours-Afternoon 223
Πίνακας 8 Σύγκριση Thursday-WorkingHours-Morning-WebAttacks 223
Πίνακας 9 Σύγκριση Wednesday-workingHours 225
Πίνακας 10 Σύγκριση UNSW-NB15 226
Πίνακας 11 Καταστάσεις ανά τιμή score 237
Πίνακας 12 Μέθοδοι άντλησης και τροποποίησης πληροφοριών 298

Συντομογραφίες & Ακρωνύμια

ACK	Acknowledgement: Σημαία επιβεβαίωσης λήψης δεδομένων - πακέτων
AI	Artificial Intelligence: Τεχνητή Νοημοσύνη
API	Application Programming Interface: Διεπαφή προγραμματισμού εφαρμογών
CPU	Central Processing Unit: Κεντρική Μονάδα Επεξεργασίας του υπολογιστή, υπεύθυνη για την εκτέλεση των εντολών και την επεξεργασία των δεδομένων
CRI	Cyber Resilience Index: Δείκτης Κυβερνο-ανθεκτικότητας που αξιολογεί την ικανότητα ενός συστήματος να αντέχει σε κυβερνοεπιθέσεις
FIN	Finish: Σημαία φυσιολογικού τερματισμού σύνδεσης
FPR	False Positive Rate: Ποσοστό ψευδώς θετικών αποτελεσμάτων κατά την ανίχνευση απειλών
ICMP	Internet Control Message Protocol: Πρωτόκολλο μηνυμάτων ελέγχου Διαδικτύου
IDS	Intrusion Detection System: Σύστημα Ανίχνευσης Εισβολών σε δίκτυα ή συστήματα
Inf	Infinity: Άπειρες τιμές
I/O	Input/Output: Είσοδου/ΕξόδουIP Internet Protocol: Πρωτόκολλο Δικτύου που χρησιμοποιείται για τη δρομολόγηση και μεταφορά πακέτων δεδομένων μεταξύ συσκευών σε ένα δίκτυο
IPv4	Internet Protocol version 4: Πρωτόκολλο Διαδικτύου έκδοσης 4
KPIs	Key Performance Indicators: Δείκτες Απόδοσης που χρησιμοποιούνται για την αξιολόγηση της αποτελεσματικότητας ενός συστήματος
LIME	Local Interpretable Model-agnostic Explanations
ML	Machine Learning: Μηχανική Μάθηση
MTTD	Mean Time To Detect: Μέσος Χρόνος Ανίχνευσης μιας απειλής ή επίθεσης
MVC	Model View Controller: Αρχιτεκτονικό πρότυπο λογισμικού
NaN	Not a Number: μη αριθμητικές ή μη διαθέσιμες τιμές
NIDS	Network Intrusion Detection System: Σύστημα Ανίχνευσης Εισβολών Δικτύου

OS	Operating System: Λειτουργικό Σύστημα
OSPF	Open Shortest Path First: Πρωτόκολλο δρομολόγησης που χρησιμοποιείται για τον υπολογισμό της βέλτιστης διαδρομής μεταφοράς δεδομένων σε ένα δίκτυο
PSH	Push: Σημαία διαταγής άμεσης παράδοσης δεδομένων (ανεξαρτήτου buffer)
RAM	Random Access Memory: Κύρια μνήμη του συστήματος που χρησιμοποιείται για την προσωρινή αποθήκευση δεδομένων και εντολών κατά τη διάρκεια εκτέλεσης των εφαρμογών
RST	Reset: Σημαία άμεσου τερματισμού σύνδεσης κατά τον εντοπισμό σφάλματος
RTT	Round-Trip Time: Πεδίο ελέγχου, υπολογίζει το χρόνο ενός πακέτου από την πηγή στον προορισμό και πάλι πίσω (επιστροφή επιβεβαίωσης)
SHAP	SHapley Additive exPlanations
SYN	Synchronization: Σημαία εκκίνησης και συγχρονισμού σύνδεσης
TCP	Transmission Control Protocol: Πρωτόκολλο μεταφοράς δεδομένων
TTL	Time To Live: Πεδίο ελέγχου από το οποίο καθορίζεται ο μέγιστος αριθμός αλμάτων (hops) ενός πακέτου πριν απορριφθεί
UDP	User Datagram Protocol: Πρωτόκολλο μεταφοράς δεδομένων που δίνει προτεραιότητα στην ταχύτητα αντί της αξιοπιστίας
UI	User Interface: Γραφικό περιβάλλον
URG	Urgent: Σημαία που υποδεικνύει επείγοντα δεδομένα (για άμεση προώθηση)
XAI	Explainable Artificial Intelligence: Επεξηγήσιμη τεχνητή νοημοσύνη που επιτρέπει την κατανόηση και ερμηνεία των αποφάσεων των μοντέλων

1. Εισαγωγή

1.1. Γενικό Πλαίσιο

Για να είναι αποτελεσματική και σύγχρονη η κυβερνοασφάλεια ενός οργανισμού, πρέπει να υπάρχει ένα αξιόπιστο δίκτυο συνεργασίας και αμοιβαίας εμπιστοσύνης ανάμεσα στα επιμέρους στοιχεία που αποτελούν τον οργανισμό (στους ανθρώπους, τις ομάδες και τους οργανισμούς) που συμμετέχουν στη λειτουργία του [1]. Το κάθε ένα από αυτά τα στοιχεία, καλείται να αναπτύξει και να εφαρμόσει τέτοιους μηχανισμούς ασφάλειας έτσι ώστε τα υπόλοιπα μέρη του δικτύου να μπορούν να βασιστούν σε αυτούς τους μηχανισμούς ως προς την πρόληψη και αντιμετώπιση των κυβερνοεπιθέσεων [1]. Τα σύγχρονα Συστήματα Ανίχνευσης Εισβολών με Μηχανική Μάθηση (Machine Learning - Intrusion Detection Systems - ML-IDSs), ως ένα από τα σημαντικά τεχνολογικά στοιχεία σε αυτό το πλέγμα, αξιοποιούν προηγμένη υπολογιστική ισχύ προκειμένου να αναλύσουν τεράστιους όγκους ιστορικών δεδομένων και να μάθουν πρότυπα κανονικής δικτυακής συμπεριφοράς, για να είναι σε θέση να εντοπίζουν συνδέσεις που αποκλίνουν από αυτά τα πρότυπα ως πιθανή κακόβουλη δραστηριότητα [2].

Ωστόσο, ενώ τα μοντέλα ML είναι εξαιρετικά χρήσιμα στην αναγνώριση και την αποτροπή επιθέσεων, συχνά δεν παρέχουν καμία ουσιαστική εξήγηση ως προς το λόγο ή τους λόγους για τους οποίους μια δικτυακή κίνηση ταξινομήθηκε ως κανονική ή ως κακόβουλη [3]. Τα IDSs ορίζονται ως μηχανισμοί που αυξάνουν την ασφάλεια ενός συστήματος αποτρέποντας μη εξουσιοδοτημένη εισβολή ή κακόβουλη πρόσβαση [4], κατά συνέπεια πρόκειται για μια λειτουργία που για να αποδώσει πλήρως, προϋποθέτει ότι οι αναλυτές ασφαλείας θα μπορούν να κατανοήσουν και να επικυρώσουν τη λογική πίσω από κάθε ανίχνευση. Με την αύξηση της επιχειρηματικής αυτοματοποίησης και του αριθμού των συνδεδεμένων συσκευών στο διαδίκτυο, ο ρυθμός εμφάνισης περιστατικών ασφαλείας αυξάνεται με ταχείς ρυθμούς [5], καθιστώντας επιτακτική την ανάγκη για ML-IDSs που να συνδυάζουν αξιόπιστη ανίχνευση με ικανότητα ερμηνείας των αποφάσεών τους.

Η πρακτική αξία ενός ML-IDS δεν παρουσιάζεται μόνο στην ανίχνευση επιθέσεων αλλά εξαρτάται επίσης τόσο από την ικανότητα ερμηνείας των αποφάσεών του όσο και από την αξιόπιστη ανίχνευση όλων των κατηγοριών επίθεσης που εκπροσωπούνται σε ένα σύνολο δεδομένων. Η ταυτόχρονη επίτευξη αυτών των δύο διαστάσεων, σε συνδυασμό με τη

συστηματική αξιολόγηση σε αναγνωρισμένα σύνολα δεδομένων που αποτελούν σημείο αναφοράς (benchmark datasets), παραμένει ένα δυσεπίλυτο ερευνητικό κενό.

1.2. Αντικείμενο και Στόχοι της Εργασίας

Η παρούσα εργασία στοχεύει στη γεφύρωση αυτού του κενού (όπως προαναφέρθηκε στην ενότητα 1.1) μέσω της ανάπτυξης του Shield IDS, ενός ολοκληρωμένου συστήματος ανίχνευσης δικτυακών εισβολών που συνδυάζει υψηλή ακρίβεια ταξινόμησης με ερμηνεύσιμες προβλέψεις. Στον πυρήνα του συστήματος βρίσκεται ο αλγόριθμος μέγιστης ενίσχυσης κλίσης (eXtreme Gradient Boosting - XGBoost), ένας αποδεδειγμένα αποτελεσματικός ταξινομητής για δεδομένα δικτυακής κυκλοφορίας, σε συνδυασμό με τη μέθοδο προσθήκης εύστοχων επεξηγήσεων (SHapley Additive exPlanations - SHAP) που παρέχει επεξηγησιμότητα σε επίπεδο χαρακτηριστικών για κάθε μεμονωμένη πρόβλεψη. Για την αντιμετώπιση της ανισορροπίας των κλάσεων, ενός από τα πιο κρίσιμα προβλήματα στα πραγματικά σύνολα δεδομένων δικτυακής κυκλοφορίας, αξιολογούνται συστηματικά οι τρεις εναλλακτικές τεχνικές εξισορρόπησης συνθετικής υπερδειγματοληψίας μειοψηφιακών κλάσεων, συνδυαστικά με την επεξεργασία πλησιέστερων γειτόνων (Synthetic Minority Over-sampling Technique and Edited Nearest Neighbours - SMOTE, SMOTE-ENN) και ενός υβριδικού σύνολο (Υποδειγματοληψία + SMOTE + ENN). Η αξιολόγηση πραγματοποιείται στα δύο ευρέως αναγνωρισμένα σύνολα δεδομένων της βιβλιογραφίας, το CIC-IDS2017[1] και το UNSW-NB15[2], καλύπτοντας πλήθος κατηγοριών επιθέσεων, από κατανεμημένες και μη άρνησης υπηρεσιών (Denial και Distributed Denial of Services - DoS και DDoS) έως PortScan, Web Attacks, Backdoors και Reconnaissance.

Πέραν του ερευνητικού πυρήνα, η εργασία αυτή δεν περιορίζεται σε ένα ακαδημαϊκό πρωτότυπο, καθώς το Shield IDS υλοποιείται ως πλήρης εφαρμογή με γραφικό περιβάλλον χρήστη (Graphical User Interface - GUI) που μπορεί να αναπτυχθεί πρακτικά σε πραγματικά περιβάλλοντα. Η εφαρμογή ενσωματώνει ζωντανή σύλληψη και ανίχνευση της δικτυακής κυκλοφορίας σε πραγματικό χρόνο, πίνακα ελέγχου στελεχών (executive dashboard) με καίριους δείκτες απόδοσης (Key Performance Indicators – KPIs) και μοντέλο υποστήριξης αποφάσεων (decision support), κεντρικό πίνακα ελέγχου για παρακολούθηση κυκλοφορίας και ανιχνευμένων απειλών, αναλυτικό περιβάλλον επεξηγησιμότητας μέσω

[1] https://www.unb.ca/cic/datasets/ids-2017.html
[2] https://research.unsw.edu.au/projects/unsw-nb15-dataset

SHAP (γραφήματα ράβδων, συνοπτικά γραφήματα και ανάλυση ανά κλάση επίθεσης), αντιστοίχιση ανιχνευμένων επιθέσεων με το πλαίσιο MITRE ATT&CK, καθώς και εργαλεία διαχείρισης IP και αναφορών ασφαλείας

Συνοπτικά, οι βασικοί στόχοι της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ML-IDS υψηλής ακρίβειας με χρήση XGBoost (ο οποίος έχει αξιολογηθεί σε πολυκλασικά σενάρια), η συστηματική διερεύνηση και η σύγκριση τριών τεχνικών εξισορρόπησης κλάσεων, όπως και η ενσωμάτωση του SHAP για ερμηνεύσιμες προβλέψεις σε επίπεδο χαρακτηριστικών ανά κατηγορία επίθεσης. Επιπλέον, η παρούσα εργασία αποσκοπεί στην αξιολόγηση και σύγκριση αποτελεσμάτων με τη σχετική βιβλιογραφία σε δύο σύνολα δεδομένων (αξιολογημένα ως σημεία αναφοράς), καθώς επίσης και στην υλοποίηση λειτουργικής εφαρμογής με ανίχνευση σε πραγματικό χρόνο, πίνακες ελέγχου, επεξηγησιμότητα και MITRE ATT&CK αντιστοίχιση.

1.3. Δομή της Διπλωματικής Εργασίας

Η εργασία οργανώνεται σε εννέα κεφάλαια, καθένα από τα οποία καλύπτει ένα συγκεκριμένο στάδιο της ερευνητικής και αναπτυξιακής διαδικασίας.

- Κεφάλαιο 2 - Θεωρητικό υπόβαθρο: Το δεύτερο κεφάλαιο θέτει τα θεωρητικά θεμέλια της εργασίας σε δύο βασικούς άξονες. Αρχικά παρουσιάζονται τα Συστήματα Ανίχνευσης Εισβολών (IDSs) και πιο συγκεκριμένα ο ορισμός, ο σκοπός, οι διαφορετικές κατηγορίες τους (NIDS, HIDS, PIDS, Hybrid) και οι τεχνικές ανίχνευσης (signature-based, anomaly-based) που χρησιμοποιούν. Στη συνέχεια, αναλύονται οι κατηγορίες των κυβερνοαπειλών που εκπροσωπούνται στα σύνολα δεδομένων που χρησιμοποιούνται, δηλαδή οι DoS, DDoS, Web Attacks, PortScan για το CIC-IDS2017 και οι Backdoor, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, Worms για το UNSW-NB15. Στον δεύτερο άξονα παρουσιάζονται οι αρχές της Μηχανικής Μάθησης στην κυβερνοασφάλεια, η θεωρία των δέντρων απόφασης (decision trees) και ομαδικής μάθησης (ensemble learning), και ο αλγόριθμος XGBoost με τα κύρια πλεονεκτήματά του.
- Κεφάλαιο 3 - Σύνολα δεδομένων: Το τρίτο κεφάλαιο παρουσιάζει αναλυτικά τα δύο σύνολα δεδομένων που χρησιμοποιούνται στην εργασία. Συγκεκριμένα, για το CIC-IDS2017 και το UNSW-NB15, αναλύεται η προέλευση και η μεθοδολογία δημιουργίας τους, η κατανομή των κλάσεων και των κατηγοριών επίθεσης, και η

σημασία κάθε χαρακτηριστικού. Στη συνέχεια τεκμηριώνεται η διαδικασία αφαίρεσης των μη-σημαντικών στηλών, παρουσιάζοντας την τεχνική και την αιτιολογία κάθε αφαίρεσης.

- Κεφάλαιο 4 - Προεπεξεργασία δεδομένων: Το τέταρτο κεφάλαιο περιγράφει αναλυτικά κάθε στάδιο της διαδικασίας προεπεξεργασίας δηλαδή την φόρτωση και αρχική εξερεύνηση του συνόλου δεδομένων, τον εντοπισμό και τη διαχείριση των διπλότυπων εγγραφών, την διαχείριση των ελλιπουσών τιμών (NaN), την κωδικοποίηση των κατηγορικών μεταβλητών (label encoding), την επιλογή των χαρακτηριστικών (feature selection) και τη δημιουργία του τελικού επεξεργασμένου συνόλου δεδομένων. Για κάθε στάδιο παρουσιάζεται τόσο η τεχνική υλοποίηση όσο και η θεωρητική αιτιολόγηση της επιλογής.
- Κεφάλαιο 5 - Εκπαίδευση και αξιολόγηση μοντέλου: Το πέμπτο κεφάλαιο εστιάζει στην εκπαίδευση του ταξινομητή XGBoost. Αρχικά περιγράφεται ο διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης και αξιολόγησης ενώ στη συνέχεια αναλύεται εκτενώς το πρόβλημα της ανισορροπίας των κλάσεων (data imbalance) και οι τρεις τεχνικές που εφαρμόζονται για την αντιμετώπισή του SMOTE, SMOTE-ENN και το υβριδικό σύνολο (υπό - δειγματοληψία + SMOTE + ENN). Ακολούθως, παρουσιάζεται η διαδικασία βελτιστοποίησης των υπέρ-παραμέτρων μέσω του RandomizedSearchCV και η επιλογή της αντικειμενικής συνάρτησης (δυαδική ή πολυκλασική ταξινόμηση).
- Κεφάλαιο 6 - Αποτελέσματα, επεξηγησιμότητα και συγκριτική ανάλυση: Το έκτο κεφάλαιο παρουσιάζει τα αποτελέσματα αξιολόγησης για το σύνολο των 21 συνδυασμών που προέκυψαν (σύνολο δεδομένων × τεχνική εξισορρόπησης). Αναλύονται οι μετρικές απόδοσης (ορθότητας, ακρίβειας, ανάκλησης, F1 βαθμολογία, ποσοστό ψευδώς θετικών (False Positive Rate - FPR) και παρουσιάζονται οι πίνακες σύγχυσης (confusion matrices). Στη συνέχεια, το κεφάλαιο αφιερώνεται στην επεξηγησιμότητα μέσω SHAP, αναφέρονται δηλαδή τα ραβδογράμματα σημαντικότητας χαρακτηριστικών, τα συνοπτικά διαγράμματα και τα διαγράμματα με πρόσημο μέσο όρο (bar, summary & signed mean plots), με ανάλυση ανά κλάση επίθεσης για κάθε συνδυασμό. Το κεφάλαιο ολοκληρώνεται με συγκριτική ανάλυση των αποτελεσμάτων ως προς τη σχετική ερευνητική βιβλιογραφία, πραγματοποιώντας σύγκριση με εργασίες που χρησιμοποιούν παρεμφερείς μεθόδους και σύνολα δεδομένων.

- Κεφάλαιο 7 - Κύρια εφαρμογή και λειτουργικές ενότητες: Το έβδομο κεφάλαιο παρουσιάζει αναλυτικά τις λειτουργικές ενότητες της κύριας εφαρμογής Shield IDS. Περιγράφεται ο πίνακας ελέγχου στελεχών με τους KPIs, το μοντέλο υποστήριξης αποφάσεων (decision support) για παροχή αξιοποιήσιμης πληροφορίας (actionable intelligence) βάσει σοβαρότητας της επίθεσης, και η δημιουργία εκτελεστικής αναφοράς ασφαλείας. Ακολουθεί ο κεντρικός πίνακας (main dashboard) ελέγχου παρακολούθησης της κυκλοφορίας, η ενότητα ανίχνευσης σε πραγματικό χρόνο (live detection) για ανίχνευση σε πραγματικό χρόνο, και το περιβάλλον επεξηγησιμότητας (explainability) με ανάλυση SHAP, σύνοψη του περιστατικού, αντιστοίχιση με το MITRE ATT&CK και σύγκριση των μοντέλων CIC-IDS2017 και UNSW-NB15. Το κεφάλαιο ολοκληρώνεται με τις ενότητες διαχείρισης αναφορών (alerts & reports) και διαχείρισης IP (blacklist/whitelist).
- Κεφάλαιο 8 - Αρχιτεκτονική και τεχνική υλοποίηση: Το όγδοο κεφάλαιο εξετάζει την τεχνική αρχιτεκτονική του Shield IDS. Παρουσιάζεται η δομή της εφαρμογής βασισμένη στο αρχιτεκτονικό πρότυπο MVC (Model-View-Controller), το οποίο διαχωρίζει με σαφήνεια τα δεδομένα, την παρουσίασή τους και τη λογική ελέγχου, επιτρέποντας έτσι τη καλύτερη οργάνωση, την επεκτασιμότητα και την συντήρηση της εφαρμογής. Στη συνέχεια αναλύεται ο μηχανισμός πολλαπλών νημάτων (FastSniffThread) για ασύγχρονη σύλληψη των δικτυακών πακέτων μέσω pcapy, καθώς και ο συσσωρευτής ροών (flow aggregator) που μετατρέπει τα ακατέργαστα πακέτα σε δομημένες αμφίδρομες ροές. Επιπλέον, περιγράφεται η φόρτωση και χρήση των προ-εκπαιδευμένων μοντέλων XGBoost, η ενσωμάτωση της SHAP ερμηνείας στη ροή πρόβλεψης, καθώς και ο μηχανισμός λευκής / μαύρης λίστας που επιτρέπει τη χειροκίνητη παράκαμψη των αποτελεσμάτων. Τέλος, αναλύονται (τεχνικά και προγραμματιστικά) οι επτά καρτέλες της εφαρμογής, όπως αυτές παρουσιάστηκαν στο κεφάλαιο 7.
- Κεφάλαιο 9 - Συμπεράσματα και μελλοντικές επεκτάσεις: Το ένατο και τελευταίο κεφάλαιο αρχικά παρουσιάζει την κατανομή εργασιών μεταξύ των δύο συγγραφέων, ενώ στη συνέχεια συνοψίζει τα κύρια ευρήματα της εργασίας, αξιολογεί τον βαθμό επίτευξης των στόχων που τέθηκαν στο παρόν κεφάλαιο, και αναλύει τους περιορισμούς της προσέγγισης. Το κεφάλαιο ολοκληρώνεται με προτάσεις για μελλοντικές επεκτάσεις, όπως η δημιουργία και αξιολόγηση σε πρωτότυπο σύνολο

δεδομένων παραγμένο από πραγματική δικτυακή κίνηση μέσω του ίδιου του Shield IDS, η ενσωμάτωση ομοσπονδιακής μάθησης (federated learning), η επέκταση σε συστήματα πρόληψης εισβολών (Intrusion Prevention Systems - IPSs), η υποστήριξη IPv6 καθώς η τρέχουσα υλοποίηση περιορίζεται σε IPv4, καθώς και η διερεύνηση τεχνικών αυξητικής μάθησης (online/incremental learning) που θα επιτρέπουν στα μοντέλα να προσαρμόζονται σε νέες κατηγορίες επιθέσεων χωρίς πλήρη επανεκπαίδευση.

2. Θεωρητικό υπόβαθρο

2.1. Συστήματα Ανίχνευσης Εισβολών (IDS)

2.1.1. Ορισμός και σκοπός των συστημάτων ανίχνευσης εισβολών.

Το IDS αποτελεί έναν κρίσιμο μηχανισμό ασφαλείας που σχεδιάστηκε για να προστατεύει πληροφοριακά συστήματα και δίκτυα από μη εξουσιοδοτημένες ή κακόβουλες εισβολές. Ουσιαστικά, ένα IDS ορίζεται ως μια συσκευή ή εφαρμογή λογισμικού που συλλέγει και αναλύει πληροφορίες σε ένα δίκτυο, προκειμένου να εντοπίσει πιθανές παραβιάσεις ασφαλείας, που αφορούν τόσο εξωτερικές επιθέσεις (intrusions) όσο και εσωτερικές κακόβουλες ενέργειες (misuse) [1]. Η ανίχνευση εισβολών αποτελεί αναπόσπαστο τμήμα της κυβερνοασφάλειας, καθώς επιτρέπει τον εντοπισμό κακόβουλης δικτυακής δραστηριότητας πριν αυτή θέσει σε κίνδυνο τη διαθεσιμότητα, ακεραιότητα ή εμπιστευτικότητα της πληροφορίας [1].

Τα IDS αναπτύχθηκαν ραγδαία τόσο στην έρευνα όσο και στη βιομηχανία, ως ανταπόκριση του αυξανόμενου αριθμού των κυβερνοεπιθέσεων εναντίον κυβερνητικών φορέων και εμπορικών επιχειρήσεων παγκοσμίως. Το ετήσιο κόστος αντιμετώπισης του κυβερνοεγκλήματος αυξάνεται συνεχώς, με τις πλέον καταστροφικές περιπτώσεις να αφορούν κακόβουλες εσωτερικές ενέργειες, επιθέσεις DoS και επιθέσεις μέσω διαδικτύου [1]. Ενδεικτικό της σοβαρότητας του φαινομένου είναι το γεγονός ότι κάθε χρόνο πάνω από 71 εκατομμύρια άτομα πλήττονται από κυβερνοεγκλήματα, ενώ μια κυβερνοεπίθεση εκδηλώνεται κατά μέσο όρο κάθε 39 δευτερόλεπτα [2]. Για την αντιμετώπιση αυτών των απειλών, οι οργανισμοί αναπτύσσουν ένα πολυεπίπεδο αμυντικό πλαίσιο που περιλαμβάνει τείχη προστασίας (firewalls), λογισμικό προστασίας από ιούς (antivirus) και κεντρικά συστήματα ανίχνευσης εισβολών [1].

Ο κύριος σκοπός ενός IDS είναι η συνεχής παρακολούθηση της κυκλοφορίας που διέρχεται από το δίκτυο, ο εντοπισμός ανώμαλης ή ύποπτης συμπεριφοράς καθώς και η έγκαιρη ειδοποίηση του διαχειριστή ασφαλείας [2]. Επιπλέον, το IDS μπορεί επίσης να παρέχει πολύτιμες πληροφορίες σχετικά με το πώς συμβαίνουν οι επιθέσεις αλλά και με τον τρόπο αποτροπής τους στο μέλλον. Αναλύοντας δεδομένα από προηγούμενες επιθέσεις, οι οργανισμοί μπορούν να εντοπίσουν τρωτά σημεία στα δίκτυά τους και να τα αντιμετωπίσουν προτού γίνουν αντικείμενο εκμετάλλευσης [3]. Η σπουδαιότητα των IDSs στην ασφάλεια δικτύων είναι αδιαμφισβήτητη. Με την αυξανόμενη εξάρτηση από το

διαδίκτυο σε όλες τις πτυχές της ζωής μας, η ανάγκη για αποτελεσματικά και αποδοτικά μέτρα ασφάλειας δικτύων συνεχίζει να αυξάνεται [3].

Τα IDSs κατηγοριοποιούνται με βάση δύο βασικά κριτήρια. Το πρώτο είναι ανάλογα με τη θέση τους στη δομή του υπολογιστικού συστήματος και το δεύτερο με βάση την μέθοδο ανίχνευσης που χρησιμοποιούν. Ως προς την τοποθεσία, διακρίνονται στις ακόλουθες κατηγορίες [4], [5].

Η πρώτη κατηγορία είναι τα Δικτυακά IDSs (Network-based IDSs - NIDSs) τα οποία εγκαθίστανται σε σημαντικά σημεία του δικτύου και παρακολουθούν το σύνολο της κυκλοφορίας που διέρχεται, συγκρίνοντάς την με βάση δεδομένων γνωστών επιθέσεων. Μόλις εντοπιστεί ύποπτη δραστηριότητα, αποστέλλεται αμέσως ειδοποίηση στον διαχειριστή. Τα NIDSs μπορούν να υλοποιηθούν τόσο ως υλικό (hardware) όσο και ως λογισμικό (software) [6].

Η δεύτερη κατηγορία είναι τα IDSs Βασισμένα στον Κόμβο (Host-based IDSs - HIDSs) τα οποία εγκαθίστανται σε μεμονωμένους υπολογιστές / κόμβους και παρακολουθούν αποκλειστικά τα εισερχόμενα και εξερχόμενα πακέτα του συγκεκριμένου κόμβου. Σε περίπτωση εντοπισμού ανωμαλίας, λαμβάνεται ένα «στιγμιότυπο» (snapshot) της κατάστασης του συστήματος και συγκρίνεται με προηγούμενες καταστάσεις [4]**.** Τα HIDS λειτουργούν ως τελευταία γραμμή άμυνας, αναλύοντας αρχεία καταγραφής (logs), κλήσεις συστήματος και εκχωρήσεις μνήμης [5].

Η τρίτη κατηγορία είναι τα IDSs Βασισμένα στο Πρωτόκολλο (Protocol-based IDS - PIDSs) τα οποία εστιάζουν στην ανάλυση των πρωτοκόλλων επικοινωνίας που χρησιμοποιούνται εντός του συστήματος. Πράκτορες (agents) αναπτύσσονται σε διακομιστές ιστού (web servers) για την εξέταση των πρωτοκόλλων, και σε περίπτωση απόκλισης από την αναμενόμενη συμπεριφορά παράγονται άμεσα ειδοποιήσεις [4].

Η τέταρτη κατηγορία είναι τα Υβριδικά IDSs (Hybrid IDSs) τα οποία συνδυάζουν τις παραπάνω μεθόδους, παρουσιάζοντας υψηλότερες επιδόσεις ανίχνευσης σε σχέση με την ανεξάρτητη χρήση κάθε επιμέρους μεθόδου [4].

Όπως προαναφέρθηκε, πέρα από την τοποθεσία ανάπτυξης, τα IDSs διαφοροποιούνται επιπλέον σε δύο διαφορετικές κατηγορίες ως προς τη μέθοδο ανίχνευσης που εφαρμόζουν [6], [7].

Η πρώτη κατηγορία αφορά τα Signature-based IDSs (βασισμένα σε υπογραφές) τα οποία για τον εντοπισμό κακόβουλης κυκλοφορίας αξιοποιούν ιστορικές βάσεις δεδομένων γνωστών επιθέσεων με χαρακτηριστικά μοτίβα όπως ο αριθμός bytes [4]. Είναι αποτελεσματικά στον εντοπισμό γνωστών απειλών, παρουσιάζουν όμως σοβαρό μειονέκτημα, καθώς αδυνατούν να αναγνωρίσουν νέες, άγνωστες επιθέσεις [1], [7]. Επιπλέον, η προσθήκη νέων κανόνων είναι χρονοβόρα και απαιτεί χειροκίνητη παρέμβαση των διαχειριστών του συστήματος [6].

Η πρώτη κατηγορία αφορά τα Anomaly-based IDSs (βασισμένα σε ανωμαλίες) τα οποία ορίζουν ένα πρότυπο κανονικής συμπεριφοράς δικτύου και εντοπίζουν αποκλίσεις από αυτό. Για τον εντοπισμό ανωμαλιών, χρησιμοποιούνται τεχνικές ML, οι οποίες επιτρέπουν τον εντοπισμό ακόμη και νέων, άγνωστων επιθέσεων [4], [5]. Ωστόσο, τα anomaly-based IDSs παρουσιάζουν συχνά υψηλότερα ποσοστά ψευδώς θετικών (false positives) αποτελεσμάτων, γεγονός που επιβαρύνει το έργο των αναλυτών ασφαλείας [7].

Παρά τη σημαντική τους συνεισφορά, τα παραδοσιακά IDSs αντιμετωπίζουν μια σειρά από σοβαρούς περιορισμούς. Η πολυπλοκότητά τους εξαρτάται από τον αριθμό των μοτίβων επιθέσεων και των διαθέσιμων υπογραφών, ενώ το δυναμικό περιβάλλον του διαδικτύου καθιστά εξαιρετικά δύσκολη τη δημιουργία ενός πλήρους συνόλου κανόνων για κάθε πιθανό τύπο επίθεσης [6]**.** Αυτές οι αδυναμίες οδήγησαν στην υιοθέτηση τεχνικών ML, οι οποίες αναλύονται στην ενότητα 2.2.1.

2.1.2. Κυβερνοαπειλές και Επιθέσεις

Η ανάπτυξη αποτελεσματικών IDSs προϋποθέτει βαθιά κατανόηση των κατηγοριών κυβερνοεπιθέσεων που στοχεύουν τα σύγχρονα δίκτυα. Στη βιβλιογραφία υπάρχουν διάφορα δημόσια διαθέσιμα σύνολα δεδομένων για αξιολόγηση των IDSs, μεταξύ των οποίων τα πλέον διαδεδομένα τα οποία είναι το NSL-KDD, το UNSW-NB15 και το CIC-IDS2017, καθένα από τα οποία καλύπτει διαφορετικές κατηγορίες επιθέσεων [5], [6]**.** Στην παρούσα εργασία επιλέχθηκαν τα σύνολα δεδομένων CIC-IDS2017 και UNSW-NB15, των οποίων η αναλυτική περιγραφή παρουσιάζεται στο κεφάλαιο 3. Στην παρούσα ενότητα, για τη καλύτερη κατανόηση του τρόπου λειτουργίας της κεντρικής εφαρμογής (Shield IDS), παρουσιάζεται το θεωρητικό υπόβαθρο κάθε κατηγορίας επίθεσης που περιέχεται σε αυτά.

Οι επιθέσεις DoS αποτελούν κακόβουλες απόπειρες καταστροφής της διαθεσιμότητας διακομιστή ή πόρου δικτύου, καθιστώντας τον προσωρινά ή μόνιμα απρόσιτο στους

νόμιμους χρήστες [8]**.** Χαρακτηριστικό γνώρισμά τους είναι ότι προέρχονται από μία μόνο πηγή, η οποία κατακλύζει τον στόχο με υπερβολικό όγκο αιτημάτων μέχρι να εξαντληθούν πλήρως οι διαθέσιμοι πόροι του [9]. Στο CIC-IDS2017 περιλαμβάνονται οι παρακάτω τέσσερις παραλλαγές της επίθεσης DoS.

- Η DoS Hulk (HTTP Unbearable Load King) αποτελεί μια από τις πλέον σημαντικές μορφές DoS επίθεσης επιπέδου εφαρμογής. Ο επιτιθέμενος αποστέλλει έναν τεράστιο όγκο εικονικής κυκλοφορίας HTTP στον web server, με αποτέλεσμα την αδυναμία εξυπηρέτησης νόμιμων χρηστών [10].
- Η DoS GoldenEye αποτελεί επίθεση τύπου DoS που στοχεύει web servers και εφαρμογές δικτύου, προκαλώντας διακοπή της διαθεσιμότητας των υπηρεσιών για τους νόμιμους χρήστες [11].
- Η DoS Slowloris αποτελεί επίθεση αργού τύπου (slow attack) η οποία λειτουργεί διατηρώντας ανοιχτές συνδέσεις με τον διακομιστή, αποστέλλοντας αργά πακέτα με σκοπό να εξαντλήσει τους πόρους του διακομιστή διαχρονικά. Η φύση της την καθιστά ιδιαίτερα δύσκολα ανιχνεύσιμη από τα IDS λόγω των διαφορετικών μοτίβων κυκλοφορίας που παρουσιάζει σε σχέση με τις κλασικές (volumetric) επιθέσεις [12].
- Η DoS Slowhttptest είναι επίθεση αργής HTTP DoS που εστιάζει στο επίπεδο εφαρμογής. Εισάγει σκόπιμες καθυστερήσεις στις αποκρίσεις του web server, επηρεάζοντας τους νόμιμους χρήστες και τις υπηρεσίες [10].
- Οι επιθέσεις DDoS αποτελούν εξελιγμένη μορφή των DoS και διαφέρουν ουσιαστικά ως προς την προέλευσή τους καθώς, σε αντίθεση με τις DoS που εκκινούν από έναν μόνο κόμβο, οι DDoS επιθέσεις εκδηλώνονται ταυτόχρονα από πολλαπλές πηγές, συνήθως μέσω δικτύων botnets που ελέγχονται από τον επιτιθέμενο [13]. Μια κακόβουλη οντότητα κατακλύζει έναν διακομιστή ή έναν ιστότοπο με υπερβολικά δεδομένα, προκαλώντας διακοπές υπηρεσιών ή ακόμη και κατάρρευση του διακομιστή [13]. Ταξινομούνται σε δύο βασικές κατηγορίες, τις Exploitation-Based επιθέσεις, στις οποίες αξιοποιείται η IP του θύματος για αποστολή απαντητικών πακέτων (reply) μέσω διακομιστών αντανάκλασης (reflection servers) εκμεταλλευόμενες πρωτόκολλα όπως TCP και UDP, και τις Reflection-Based επιθέσεις, που αποκρύπτουν την ταυτότητα του επιτιθέμενου χειριζόμενες υπηρεσίες διαδικτύου όπως DNS, NTP και SNMP [13]. Η ολοένα αυξανόμενη πολυπλοκότητα των DDoS επιθέσεων, όπου οι επιτιθέμενοι αξιοποιούν πλέον νέες στρατηγικές

εκμεταλλευόμενοι πρωτόκολλα επιπέδου εφαρμογής βασισμένα σε TCP/UDP, καθιστά την αντιμετώπισή τους ιδιαίτερα απαιτητική [13].

- Η σάρωση θυρών αποτελεί τεχνική δικτυακής αναγνώρισης κατά την οποία ο επιτιθέμενος αποστέλλει πακέτα σε ένα εύρος θυρών του στόχου, με σκοπό τον εντοπισμό ανοιχτών θυρών και των υπηρεσιών που εκτελούνται σε αυτές [14]. Το CIC-IDS2017 περιλαμβάνει σενάριο PortScan από το αρχείο Friday-WorkingHours-Afternoon, το οποίο αξιοποιείται ευρέως στη βιβλιογραφία για την ανάπτυξη και αξιολόγηση μοντέλων ανίχνευσης σαρώσεων θυρών [14].
- Οι επιθέσεις ιστού εκμεταλλεύονται αδυναμίες ασφαλείας σε web εφαρμογές για να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση, συμπεριλαμβανομένων επιθέσεων SQL injection, εισαγωγής εντολών και της δημιουργίας σεναρίων μεταξύ των ιστοσελίδων [9]. Στο CIC-IDS2017 εμπεριέχονται τρεις τέτοιου είδους διαφορετικοί τύποι επιθέσεων, οι Brute Force, SQL Injection και XSS.
- Η Brute Force η οποία αφορά συστηματική επίθεση εικασίας κωδικών πρόσβασης που στοχεύει μηχανισμούς αυθεντικοποίησης. Ο επιτιθέμενος δοκιμάζει διαδοχικά όλους τους δυνατούς συνδυασμούς κωδικών μέχρι να εντοπιστεί ο σωστός, αποκτώντας έτσι παράνομο έλεγχο πάνω στη συσκευή-στόχο [9]. Η επίθεση αυτή χαρακτηρίζεται από επαναλαμβανόμενα μοτίβα προσπαθειών σύνδεσης (login attempts) που αποκαλύπτουν συμπεριφορά εικασίας διαπιστευτηρίων [15]. Η SQL Injection αποτελεί επίθεση web εφαρμογών κατά την οποία κακόβουλος κώδικας SQL εισάγεται στο σύστημα λόγω ανεπαρκούς υλοποίησης, με στόχο την απόκτηση μη εξουσιοδοτημένης πρόσβασης [11]. Η Cross-Site Scripting (XSS) αποτελεί επίθεση web εφαρμογών κατά την οποία εισάγονται κακόβουλα scripts με σκοπό την απόκτηση μη εξουσιοδοτημένης πρόσβασης [9]. Το CIC-IDS2017 περιέχει σενάρια και των τριών αυτών τύπων web attack στο αρχείο Thursday-WorkingHours-Morning [16].

Το UNSW-NB15 περιλαμβάνει τις ακόλουθες εννέα κατηγορίες επιθέσεων που αντιπροσωπεύουν σύγχρονες απειλές δικτύου, συνδυάζοντας πραγματική κανονική κυκλοφορία με συνθετική κακόβουλη δραστηριότητα [8].

- Τα Fuzzers αποτελούν απόπειρα αναστολής λειτουργίας προγράμματος ή δικτύου μέσω τυχαία παραγόμενων δεδομένων εισόδου [8].

- Η Analysis περιλαμβάνει επιθέσεις σάρωσης θυρών, spam και διείσδυσης μέσω HTML αρχείων [8].
- Το Backdoors είναι τεχνική απόκτησης πρόσβασης σε υπολογιστικό σύστημα με παράκαμψη των κανονικών μηχανισμών ασφαλείας [8].
- Η DoS είναι η κακόβουλη απόπειρα καταστροφής της διαθεσιμότητας διακομιστή ή πόρου δικτύου, καθιστώντας τον προσωρινά απρόσιτο, συνήθως διακόπτοντας ή αναστέλλοντας τις υπηρεσίες ενός κόμβου συνδεδεμένου στο διαδίκτυο [8].
- Στην Exploits ο επιτιθέμενος εκμεταλλεύεται γνωστή αδυναμία ασφαλείας σε λειτουργικό σύστημα ή εφαρμογή λογισμικού για να αποκτήσει μη εξουσιοδοτημένη πρόσβαση ή να προκαλέσει κατάρρευση του συστήματος [8].
- Η Generic είναι τεχνική που λειτουργεί κατά όλων των block ciphers με δεδομένο μέγεθος block και κλειδιού, ανεξάρτητα από τη δομή τους [8].
- Η Reconnaissance περιλαμβάνει παθητικές και ενεργητικές τεχνικές σάρωσης δικτύου για τη συλλογή πληροφοριών σχετικά με ευάλωτες συσκευές, υπηρεσίες και διαμορφώσεις δικτύου [9].
- Η Shellcode είναι μικρό τμήμα κώδικα που χρησιμοποιείται ως payload στην εκμετάλλευση αδυναμιών λογισμικού [8].
- Η Worms είναι κακόβουλος κώδικας που αναπαράγεται αυτόνομα και εξαπλώνεται σε άλλους υπολογιστές μέσω δικτύου, εκμεταλλευόμενος αδυναμίες ασφαλείας στο σύστημα-στόχο χωρίς να απαιτείται παρέμβαση χρήστη [8].

2.2. Μηχανική Μάθηση στην Κυβερνοασφάλεια

2.2.1. Εισαγωγή στη Μηχανική Μάθηση

Η Μηχανική Μάθηση (Machine Learning — ML) αποτελεί κλάδο της Τεχνητής Νοημοσύνης που έχει αναδειχθεί ως μετασχηματιστικό εργαλείο στον τομέα της κυβερνοασφάλειας. Αξιοποιώντας την ικανότητα να μαθαίνει από δεδομένα και να βελτιώνεται επαναληπτικά χωρίς ρητό προγραμματισμό, οι τεχνικές ML μπορούν να εντοπίζουν αποτελεσματικά νέες και εξελιγμένες κυβερνοαπειλές [17], [18]. Οι αλγόριθμοί της αναλύουν μοτίβα και ανωμαλίες σε τεράστια σύνολα δεδομένων κυκλοφορίας δικτύου, επιτρέποντας τον εντοπισμό ύποπτων δραστηριοτήτων με αξιοσημείωτη ακρίβεια και αποδοτικότητα [18], [19].

Τα μοντέλα ML διακρίνονται στις παρακάτω τρεις βασικές κατηγορίες ανάλογα με τον τρόπο εκπαίδευσής τους [20], [21].

(α) Στην εποπτευόμενη μάθηση (supervised learning), σύμφωνα με την οποία τα μοντέλα εκπαιδεύονται σε επισημειωμένα δεδομένα (labeled data) όπου η ετικέτα κλάσης, π.χ. «κανονικό» ή «επίθεση», είναι γνωστή εκ των προτέρων. Τα δεδομένα με ετικέτα αποτελούν απαραίτητη προϋπόθεση, ενώ η υψηλή ακρίβεια ανίχνευσης είναι το κύριο πλεονέκτημά της. Ωστόσο, η ακρίβεια ανίχνευσης για άγνωστες απειλές απαιτεί περαιτέρω επαλήθευση [21]. Αλγόριθμοι όπως τα δέντρα απόφασης, τυχαίο δάσος (Random Forest), υποστήριξης διανυσμάτων (Support Vector Machines - SVM) και XGBoost αποτελούν δημοφιλείς αλγόριθμους εποπτευόμενης μάθησης που έχουν εφαρμοστεί ευρέως σε IDSs [22], [23], [24], [25].

(β) Στη μη εποπτευόμενη μάθηση (unsupervised learning), η οποία δεν απαιτεί επισημειωμένα δεδομένα και εξειδικεύεται στον εντοπισμό άγνωστων απειλών. Ωστόσο, παρουσιάζει χαμηλότερη ακρίβεια ανίχνευσης σε σχέση με την εποπτευόμενη μάθηση, ενώ η διαμόρφωση του κατωφλίου για την ταξινόμηση μεταξύ κανονικής και κακόβουλης κυκλοφορίας είναι επί τούτω (ad hoc) [21]. Μέθοδοι όπως το Isolation Forest, ο Local Outlier Factor (LOF) και οι αλγόριθμοι clustering έχουν αναπτυχθεί λόγω της ικανότητάς τους να εντοπίζουν ανωμαλίες χωρίς επισημειωμένα δεδομένα [26].

(γ) Στην ημι-εποπτευόμενη μάθηση (semi-supervised learning), η οποία τοποθετείται μεταξύ των δύο προηγούμενων κατηγοριών, συνδυάζοντας μικρή ποσότητα επισημειωμένων με μεγάλη ποσότητα μη επισημειωμένων δεδομένων κατά την εκπαίδευση. Οι ημι-εποπτευόμενες μέθοδοι παρουσιάζουν γενικά χαμηλότερα ποσοστά ψευδώς θετικών (false positives) σε σχέση με τις πλήρως μη εποπτευόμενες [20].

Όπως αναφέραμε και παραπάνω παραδοσιακές τεχνικές ML όπως τα τυχαία δάση, η λογιστική παλινδρόμηση (Logistic Regression), τα δέντρα απόφασης και SVM έχουν χρησιμοποιηθεί εκτενώς σε IDSs [26]. Ωστόσο, η εισαγωγή της Βαθιάς Μάθησης (Deep Learning) έφερε ισχυρότερες προσεγγίσεις, όπως τα επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks - RNN), τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNN) και τα δίκτυα αμφίδρομης μακροχρόνιας – βραχυπρόσθεσμης μνήμης (Bidirectional Long Short-Term Memory - Bi-LSTM), τα οποία αποδείχθηκαν εξαιρετικά αποτελεσματικά στην ανίχνευση ανώμαλης συμπεριφοράς [26]. Πιο

συγκεκριμένα, τα CNN εξελίσσονται στην αναγνώριση χωρικών μοτίβων, τα LSTM στην ανάλυση χρονικών εξαρτήσεων και τα GCN στον εντοπισμό σχεσιακών μοτίβων μεταξύ των κόμβων του δικτύου [27]. Παρά την αποτελεσματικότητά τους, τα μοντέλα DL αντιμετωπίζουν συχνά δυσκολίες με δεδομένα υψηλής διαστατικότητας καθώς απαιτούν αυστηρή προεπεξεργασία, χρειάζονται μεγαλύτερους όγκους εκπαιδευτικών δεδομένων όπως περισσότερους και υπολογιστικούς πόρους [12], [26].

Η ανάπτυξη αποτελεσματικών IDS μοντέλων βασισμένα σε ML συνοδεύεται από σειρά σημαντικών προκλήσεων. Αρχικά, η υπερπροσαρμογή (overfitting) όπου αποτελεί διαδεδομένο πρόβλημα και κατά την οποία το μοντέλο μαθαίνει αποτελεσματικά από τα εκπαιδευτικά δεδομένα αλλά αποτυγχάνει να γενικεύσει σε νέα δεδομένα, επειδή αποστηθίζει το σύνολο εκπαίδευσης αντί να αναγνωρίζει γενικά μοτίβα [5]. Έπειτα, η ανισορροπία κλάσεων (class imbalance) η οποία επίσης αποτελεί ένα από τα σοβαρότερα προβλήματα στα σύνολα δεδομένων ανίχνευσης εισβολών, καθώς η κανονική κυκλοφορία υπερτερεί κατά πολύ αριθμητικά έναντι της κακόβουλης, γεγονός που οδηγεί σε μεροληπτικά μοντέλα που ευνοούν τη συχνότερη κλάση [5], [28]. Για την αντιμετώπισή της χρησιμοποιούνται τεχνικές όπως η τεχνική SMOTE, η υποδειγματοληψία (undersampling) και η μάθηση με ευαισθησία στο κόστος [28]. Επίσης, η προεπεξεργασία δεδομένων είναι ένα καθοριστικό βήμα, καθώς τα διαθέσιμα σύνολα δεδομένων παρουσιάζουν συχνά ελλιπείς τιμές [22], ασυνέπειες τύπων δεδομένων και τιμές σε μεγάλη κλίμακα που δυσχεραίνουν την εκπαίδευση [5]. Τεχνικές όπως ο καθαρισμός δεδομένων (data cleaning), η κωδικοποίηση (encoding) και η κανονικοποίηση (normalization) εφαρμόζονται για τη βελτίωση της ποιότητας των δεδομένων εισόδου [5].

Παρά τις εντυπωσιακές επιδόσεις τους, τα IDSs βασισμένα σε ML/DL αντιμετωπίζουν ένα ιδιαίτερα σημαντικό δίλημμα μεταξύ της ακρίβειας και της ερμηνευσιμότητας [19]. Τα black-box μοντέλα, αν και εξαιρετικά ακριβή, δεν παρέχουν κατανοητές εξηγήσεις για τις αποφάσεις τους, υπονομεύοντας την εμπιστοσύνη των αναλυτών ασφαλείας [26]. Η εφαρμογή της ML στα IDSs αντιπροσωπεύει ένα σημαντικό βήμα προόδου για την ανάπτυξη δυναμικών στρατηγικών ασφαλείας, καθώς μπορούν να προλαμβάνουν και να εξουδετερώνουν απειλές σε πραγματικό χρόνο, διασφαλίζοντας την ακεραιότητα και εμπιστευτικότητα των πληροφοριακών συστημάτων [18]. Αυτή η ανάγκη για διαφάνεια αποτέλεσε ουσιαστικά το κύριο κίνητρο για την ανάπτυξη της Ερμηνεύσιμης Τεχνητής Νοημοσύνης (Explainable Artificial Intelligence - XAI).

2.2.2. Θεωρία των δέντρων απόφασης και της ομαδικής μάθησης

Το δέντρο απόφασης αποτελεί έναν μη-παραμετρικό αλγόριθμο εποπτευόμενης μάθησης για εργασίες ταξινόμησης και παλινδρόμησης [1], [22]. Η λειτουργία του βασίζεται στον επαναληπτικό διαχωρισμό των δεδομένων εισόδου σε μικρότερα υποσύνολα, βάσει συγκεκριμένων συνθηκών που ορίζονται - εφαρμόζονται στις τιμές των χαρακτηριστικών του. Σε κάθε βήμα, ο αλγόριθμος επιλέγει ένα χαρακτηριστικό και ένα κατώφλι που διαχωρίζει καλύτερα τα δεδομένα, δημιουργώντας κλάδους που οδηγούν σε διαφορετικά αποτελέσματα. Η διαδικασία αυτή συνεχίζεται μέχρι τα δεδομένα να διαχωριστούν σε διακριτές ομάδες, σχηματίζοντας μια ιεραρχική δομή δέντρου [22]. Το μοντέλο κατασκευάζεται από πάνω προς τα κάτω (top-down) από τον ριζικό κόμβο (root node), με διαδοχικούς κόμβους απόφασης και φύλλα (leaf nodes). Οι ριζικοί κόμβοι αντιστοιχούν στα πλέον σημαντικά χαρακτηριστικά, ενώ τα φύλλα παρέχουν την τελική ταξινόμηση [1].

Το δέντρο απόφασης παράγει κανόνες ταξινόμησης με βάση συγκεκριμένα κριτήρια διαχωρισμού. Ένας από τους καλύτερους αλγόριθμους κατασκευής είναι ο ID3 (Iterative Dichotomiser 3), ο οποίος κατασκευάζει το δέντρο top-down, με τη μέθοδο άπληστης (greedy) αναζήτησης στα εκπαιδευτικά δεδομένα, χρησιμοποιώντας τη στατιστική μετρική κέρδους πληροφορίας (information gain) για την επιλογή του χαρακτηριστικού που θα ελεγχθεί σε κάθε κόμβο [1]. Γενικότερα, για την επιλογή του βέλτιστου χαρακτηριστικού διαχωρισμού οι δύο βασικές μετρικές είναι η εντροπία και το κέρδος πληροφορίας[1], [29].

Η εντροπία (entropy) αποτελεί μέτρο της θεωρίας πληροφορίας που εκφράζει - χαρακτηρίζει την ασυνέπεια ενός αυθαίρετου συνόλου παραδειγμάτων. Εάν το χαρακτηριστικό - στόχος λαμβάνει c διαφορετικές τιμές, η εντροπία S ορίζεται ως:

$$E(S) = \sum_{i=1}^{c} -P_i \, log_2 \, p_i \quad (1)$$

όπου p_iείναι η πιθανότητα του S να ανήκει στην κλάση i [1].

Το κέρδος πληροφορίας (information gain) χρησιμοποιείται για την κατάταξη των χαρακτηριστικών και την κατασκευή του δέντρου, τοποθετώντας σε κάθε κόμβο το χαρακτηριστικό με το υψηλότερο κέρδος [1]. Υπολογίζει την αναμενόμενη μείωση της εντροπίας μετά τον διαχωρισμό των δεδομένων βάσει ενός χαρακτηριστικού. Το κέρδος πληροφορίας Gain (S, A) ενός χαρακτηριστικού A ορίζεται ως:

$$Gain(S, A) = E(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} E(S_v) \quad (2)$$

Εναλλακτικά, ο αλγόριθμος CART χρησιμοποιεί τον Δείκτη Gini ως μέτρο ασυνέπειας για την αξιολόγηση της ποιότητας ενός διαχωρισμού, ποσοτικοποιώντας έτσι την πιθανότητα εσφαλμένης ταξινόμησης ενός τυχαία επιλεγμένου στοιχείου. Ένα χαρακτηριστικό γνώρισμα του CART είναι η αποκλειστική χρήση των δυαδικών διαχωρισμών (binary splits), που απλοποιεί τη δομή του δέντρου και βελτιώνει την ερμηνευσιμότητα, ενώ παράλληλα εφαρμόζει «κλάδεμα κόστους-πολυπλοκότητας» (cost-complexity pruning) για αποφυγή της υπερπροσαρμογής [22].

Τα δέντρα απόφασης, παρότι θεωρούνται εγγενώς ερμηνεύσιμα μοντέλα καθώς η ιεραρχική τους δομή επιτρέπει απλή οπτικοποίηση και κατανόηση της διαδικασίας λήψης αποφάσεων, εκτιμώνται ιδιαίτερα για την ερμηνευσιμότητά τους [1], [22]. Παρέχουν ευελιξία και μπορούν να διαχειριστούν αποτελεσματικά τόσο αριθμητικά όσο και κατηγορικά δεδομένα, ωστόσο, ένα μεμονωμένο δέντρο μπορεί να παρουσιάσει υπερπροσαρμογή, γεγονός που οδήγησε στην ανάπτυξη των συνολικών (ensemble) μεθόδων [22].

Η ομαδική μάθηση αποτελεί μια τεχνική κατά την οποία πολλαπλά μοντέλα εκπαιδεύονται και συνδυάζονται για τη βελτίωση της συνολικής απόδοσης της πρόβλεψης. Ενώ τα μεμονωμένα μοντέλα ενδέχεται να παρουσιάζουν περιορισμούς, ο συνδυασμός των προβλέψεών τους μέσω κατάλληλης μεθόδου συνάθροισης (όπως το boosting) μπορεί να βελτιώσει σημαντικά την ακρίβεια και την ευρωστία [22]. Γενικά, ένα σύνολο αδύναμων μαθητών (weak learners) εκφράζεται ως:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F}. \quad (3)$$

όπου K είναι ο αριθμός των δέντρων στο σύνολο και f_k είναι η συνάρτηση εντός του χώρου υπόθεσης. Το F συμβολίζει το σύνολο όλων των δυνατών μοντέλων CART και $\hat{y}_i$ αντιπροσωπεύει την προβλεπόμενη τιμή της πραγματικής ετικέτας του x_i διαμέσου όλων των δέντρων του συνόλου [22]. Η αντικειμενική συνάρτηση ενός ensemble learner γράφεται ως:

$$\mathcal{L} = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k). \quad (4)$$

όπου $\omega(f_k)$ είναι ο όρος ομαλοποίησης (regularization) που μετρά την πολυπλοκότητα του δέντρου f_k, και $l(y_i, \hat{y}_i)$ είναι η συνάρτηση απώλειας που ποσοτικοποιεί το σφάλμα ή την απόκλιση μεταξύ της πραγματικής τιμής y και της προβλεπόμενης τιμής $\hat{y}$ [22].

Ο αλγόριθμος συσσώρευσης (Bootstrap Aggregation ή Bagging) χρησιμοποιεί τυχαία υποσύνολα εκπαιδευτικών δεδομένων με επανατοποθέτηση και συνδυάζει τις προβλέψεις για ταξινόμηση μέσω ψηφοφορίας πλειοψηφίας ή μέσου όρου παλινδρόμησης [29], [30]. Ο αλγόριθμος τυχαίου δάσους αποτελεί το κλασικό παράδειγμα συνόλου bagging που συνδυάζει πολλαπλά δέντρα απόφασης εκπαιδευμένα σε τυχαία υποσύνολα δεδομένων και χαρακτηριστικών, με την τελική πρόβλεψη να καθορίζεται από ψηφοφορία πλειοψηφίας [30], [31]. Είναι ένας από τους πλέον διαδεδομένους αλγόριθμους για επίλυση προβλημάτων πολλαπλής ταξινόμησης και πρόβλεψης, συνδυάζοντας τυχαίο υποχώρο (random subspace) και συνδυαστική μάθηση με δειγματοληψία bootstrap (bagging ensemble learning) [29]. Τα τυχαία δάση είναι γνωστά για την ικανότητά τους να μειώνουν την υπερπροσαρμογή [31], ενώ παράλληλα επιτυγχάνουν υψηλότερη ακρίβεια, καθώς η χρήση περισσότερων δέντρων μειώνει τη γενίκευση και τον θόρυβο [2].

Η ενίσχυση (boosting) αποτελεί τεχνική ML κατά την οποία εκπαιδεύονται διαδοχικά πολλαπλά μοντέλα με στόχο τη βελτίωση της συνολικής ακρίβειας πρόβλεψης και τη δημιουργία ενός πιο ισχυρού συστήματος [32]. Αποτελείται από τρία βασικά συστατικά, το πρώτο εκ των οποίων είναι η συνάρτηση απώλειας που αξιολογεί πόσο καλά το μοντέλο κάνει προβλέψεις. Το δεύτερο αφορά τον αδύναμο μαθητή που ταξινομεί δεδομένα με χαμηλή απόδοση και υψηλό ποσοστό σφάλματος, ενώ το τρίτο το προσθετικό μοντέλο που τροποποιεί σε διαδοχική και επαναληπτική μέθοδο προσθέτοντας τους προηγούμενους αδύναμους μαθητές έναν προς έναν, βελτιώνοντας το τελικό μοντέλο ενώ μειώνει το σφάλμα μεροληψίας σε κάθε κύκλο [29]. Οι κυριότεροι αλγόριθμοι Boosting που χρησιμοποιούνται στη βιβλιογραφία για εφαρμογές IDS είναι ο AdaBoost, ο Light Gradient Boosting και ο XGBoost.

Ο αλγόριθμος προσαρμοστικής ενίσχυσης (Adaptive Boosting - AdaBoost) αποτελεί ensemble-based αλγόριθμο ML κατάλληλο για εργασίες ταξινόμησης, ο οποίος λειτουργεί αναθέτοντας αρχικά ίσα βάρη σε όλα τα σημεία εκπαίδευσης. Σε κάθε επανάληψη, τα εσφαλμένα ταξινομημένα παραδείγματα λαμβάνουν αυξημένα βάρη, ώστε τα επόμενα μοντέλα να εστιάζουν περισσότερο στις δυσκολότερες περιπτώσεις και η διαδικασία συνεχίζεται μέχρι να επιτευχθεί το χαμηλότερο σφάλμα. Βασικό μειονέκτημά του είναι η

απαίτηση για σύνολο δεδομένων υψηλής ποιότητας, ελεύθερο από θορυβώδη δεδομένα [29].

Ο αλγόριθμος ελαφριάς ενίσχυσης (Light Gradient Boosting - LGBM) αποτελεί πλαίσιο ενίσχυσης κλίσης (gradient boosting) που χρησιμοποιεί μέθοδο μάθησης με βάση το δέντρο (tree-based). Το διακριτό χαρακτηριστικό του έναντι άλλων tree-based αλγόριθμων είναι ο τρόπος ανάπτυξης των δέντρων καθώς, ενώ οι συμβατικοί αλγόριθμοι αναπτύσσουν τα δέντρα οριζόντια (level-wise), ο LGBM τα αναπτύσσει κατακόρυφα (leaf-wise), επιλέγοντας σε κάθε βήμα το φύλλο με τη μικρότερη απώλεια (delta loss). Αυτή η προσέγγιση επιτρέπει υψηλότερη μείωση της απώλειας σε σχέση με τη οριζόντια ανάπτυξη για τον ίδιο αριθμό φύλλων. Ο χαρακτηρισμός «Light» αναφέρεται στη μειωμένη υπολογιστική επιβάρυνση, στην ικανότητα επεξεργασίας μεγάλων όγκων δεδομένων και στις ελάχιστες απαιτήσεις μνήμης. Ωστόσο, καθώς αυξάνεται ο αριθμός των χαρακτηριστικών, η αποδοτικότητα και η κλιμακωσιμότητα παραμένουν ανεπαρκείς. Επιπλέον, η κατακόρυφη ανάπτυξη οδηγεί σε δέντρα μεγαλύτερης πολυπλοκότητας, γεγονός που μπορεί να προκαλέσει υπερπροσαρμογή [29].

Ο XGBoost αποτελεί κλιμακούμενο ensemble-based αλγόριθμο ML, σχεδιασμένο για δομημένα ή πινακοποιημένα δεδομένα, που επιλύει διάφορα προβλήματα ML γρήγορα και αποτελεσματικά. Κατασκευάζει δέντρα απόφασης διαδοχικά, αναθέτοντας βάρη στις ανεξάρτητες μεταβλητές, οι οποίες τροφοδοτούνται στο επόμενο δέντρο με αυξημένα βάρη για τις εσφαλμένες ταξινομήσεις, με αποτέλεσμα τη δημιουργία ενός ισχυρότερου συνδυαστικού μοντέλου [29]. Λόγω της κεντρικής σημασίας του για την παρούσα εργασία, αναλύεται διεξοδικά στην επόμενη ενότητα.

2.2.3. Ο αλγόριθμος XGBoost

Ο XGBoost αποτελεί αλγόριθμο ML βασισμένο σε δέντρα απόφασης που αξιοποιεί τη δομή ενίσχυσης κλίσης [14]. Όπως αναφέρθηκε στην προηγούμενη ενότητα, για τα προβλήματα που περιλαμβάνουν δομημένα ή πινακοποιημένα δεδομένα, οι υπολογισμοί βασισμένοι σε δέντρα απόφασης θεωρούνται κορυφαίοι.

Ο XGBoost επεκτείνει τη γενική αντικειμενική συνάρτηση των μηχανών ενίσχυσης κλίσης (Gradient Boosting Machines – GBMs) ενσωματώνοντας τον ειδικό όρο ομαλοποίησης (regularization) $\omega(f_t)$ ο οποίος ορίζεται ως:

$$\omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j{}^2 \quad (5)$$

όπου T είναι ο αριθμός των φύλλων του δέντρου, ω_j το βάρος του φύλλου j, και γ, λ είναι παράμετροι ομαλοποίησης για τον αριθμό των φύλλων και των βαρών αντίστοιχα. Ένας αλγόριθμος GBM που ενσωματώνει αυτόν τον όρο ομαλοποίησης ονομάζεται XGBoost [22].

Η αντικειμενική συνάρτηση του XGBoost σε κάθε βήμα t γράφεται ως:

$$L^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t{}^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j{}^2 \quad (6)$$

όπου g_i και h_i είναι τα πρώτης και δεύτερης τάξης στατιστικά της συνάρτησης απώλειας. Αξίζει να σημειωθεί ότι η γενική αντικειμενική συνάρτηση των ensemble μοντέλων (εξίσωση 4 της προηγούμενης ενότητας) δεν είναι άμεσα εφαρμόσιμη στα GBMs, καθώς η ταυτόχρονη βελτιστοποίηση όλων των δέντρων είναι υπολογιστικά αδύνατη [22].

Κατά την εκπαίδευση, κάθε φορά που ένα χαρακτηριστικό χρησιμοποιείται για διαχωρισμό κόμβου, υπολογίζεται το gain score για αυτό τον διαχωρισμό :

$$\text{gain} = \frac{1}{2}\left[\frac{G_L{}^2}{H_L + \lambda} + \frac{G_R{}^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (7)$$

Ο πρώτος και ο δεύτερος όρος αντιπροσωπεύουν τα scores του αριστερού και δεξιού νέου φύλλου αντίστοιχα, ο τρίτος όρος είναι το score του αρχικού φύλλου και ο τέταρτος αντιπροσωπεύει τον παράγοντα ομαλοποίησης. Εάν το συνολικό κέρδος (gain) είναι μικρότερο από τον παράγοντα ομαλοποίησης, δεν προστίθεται ο κλάδος επιτρέποντας έτσι στο κλάδεμα δέντρου (tree pruning) να εκτελείται ταυτόχρονα με την εκπαίδευση. Ο XGBoost αθροίζει τις βαθμολογίες του κέρδους (gain scores) για κάθε χαρακτηριστικό σε όλα τα ενισχυμένα δέντρα απόφασης, παρέχοντας έτσι μέτρο σπουδαιότητας χαρακτηριστικών (feature importance). Υψηλότερο κέρδος υποδηλώνει ότι ένα χαρακτηριστικό συνεισφέρει περισσότερο στη βελτίωση της απόδοσης του μοντέλου [22].

Τα gain scores του XGBoost διαφέρουν ουσιαστικά από τα κλασικά μέτρα θεωρίας πληροφορίας (εντροπία, δείκτης Gini) ως προς τη δομή, τη θεωρητική βάση και την πρακτική εφαρμογή καθώς τα gain scores προκύπτουν από τη βελτιστοποίηση της

ομαλοποιημένης συνάρτησης απώλειας του XGBoost, ενώ τα μέτρα ασυνέπειας εξαρτώνται από τις συχνότητες κλάσεων στους κόμβους [22].

Η ανωτερότητα του XGBoost στον τομέα των IDSs έχει τεκμηριωθεί εκτενώς στη βιβλιογραφία. Αποτελεί συνδυαστικό μοντέλο κατασκευασμένο με πολλαπλά δέντρα απόφασης για να καλύψει τις αδυναμίες των μεμονωμένων δέντρων με χαμηλή ακρίβεια [16]. Σε συγκριτική αξιολόγηση με τα LinearSVC, SGD, δέντρα απόφασης και τυχαίο δάσος στο σύνολο δεδομένων CIC-IDS2017, το XGBoost επέδειξε ανώτερη απόδοση επιτυγχάνοντας ορθότητα (accuracy), ακρίβεια πρόβλεψης (precision), ανάκληση (recall) και F1- score 99.9%, με εξαιρετικά χαμηλό ποσοστό ψευδώς αρνητικών (False Negative Rate – FNR) 0.0004 και ποσοστό ψευδώς θετικών (False Positive Rate – FPR) 0.0016, αναδεικνύοντας την αποτελεσματικότητά του στην ελαχιστοποίηση ψευδών συναγερμών [19]. Σε αξιολόγηση για ανίχνευση DDoS επιθέσεων, το XGBoost πέτυχε ορθότητα 99.07% με μέση βαθμολογία διασταυρούμενης επικύρωσης (Cross-Validation Score) 97.36% [13], όπως επίσης, σε συνδυασμό με την σπουδαιότητα των χαρακτηριστικών μέσω XGBoost, η ορθότητα βελτιώθηκε από 89.89% σε 98.13% [24].

2.2.4. Πλεονεκτήματα του XGBoost

Ο XGBoost υπερτερεί των παραδοσιακών αλγόριθμων ML λόγω μιας σειράς σημαντικών πλεονεκτημάτων, με σημαντικότερα από όλα τη ταχύτητά του και τη παράλληλη επεξεργασία. Πιο συγκεκριμένα, είναι περίπου δέκα φορές ταχύτερος από τις προηγούμενες μεθόδους ταξινόμησης, ενώ μέσω της δυνατότητας χρήσης πολλαπλών πυρήνων επιτρέπει την παράλληλη επεξεργασία ώστε να έχει ακόμα πιο υψηλή απόδοση [14]. Σε σύγκριση με τα παραδοσιακά μοντέλα ML (όπως τυχαία δάση ή SVM), απαιτεί το χαμηλότερο χρόνο εκπαίδευσης και παρέχει καλύτερο έλεγχο μέσω της ρύθμισης των υπερπαραμέτρων του (hyperparameter tuning) [12]. Επίσης, με τη βοήθεια της ομαλοποίησης και της τεχνικής του κλαδέματος δέντρου, αφαιρώντας τμήματα που προσφέρουν ελάχιστες ή μηδενικές περιπτώσεις ταξινόμησης μέχρι το μέγιστο βάθος του δέντρου χωρίς να τερματίζει μόνο μετά από αρνητική κλίση [14], μετριάζει αποτελεσματικά την υπερπροσαρμογή, καθιστώντας τον κατάλληλο για IDS με μη ισορροπημένα σύνολα δεδομένων [12], [33].

Πλεονέκτημα αποτελεί επίσης και η διαχείριση των ελλιπών τιμών και αραιών δεδομένων. Ο XGBoost αποτελεί αλγόριθμο με επίγνωση της αραιότητας (sparsity-aware) για ακανόνιστα - αραιά δεδομένα, επομένως και μπορεί να διαχειριστεί ελλιπείς τιμές σε αντίθεση με άλλους όπως ο αλγόριθμος λογιστικής παλινδρόμησης (Logistic Regression)

[14]. Ακόμα, η διασταυρούμενη επικύρωση αποτελεί μια ιδιαίτερα σημαντική παράμετρο που υπάρχει ήδη ενσωματωμένη στο πλαίσιο (framework) του αλγόριθμου, χωρίς να απαιτείται εγκατάσταση εξωτερικού πακέτου. Παράλληλα, αναφορικά με τις μετρικές αξιολόγησης και την αντικειμενική συνάρτηση, ο ορισμός και η παραμετροποίησή τους έγκειται στην ευχέρεια του χρήστη [14].

Η κλιμακωσιμότητα και η αποδοτικότητα είναι ιδιαίτερα σημαντικά χαρακτηριστικά και ο XGBoost αποτελεί ένα κλιμακούμενο πλαίσιο που υποστηρίζει την πλειοψηφία των σεναρίων της μηχανικής μάθησης, συμπεριλαμβανομένης της ανίχνευσης πλημμυρισμού δεδομένων (data flooding) όπως στις επιθέσεις DDoS [14]. Σε σύγκριση με τα DL μοντέλα, ο XGBoost παρέχει ισορροπία μεταξύ της υψηλής απόδοσης ανίχνευσης και της χαμηλότερης υπολογιστικής επιβάρυνσης [12].

Παρά τα σημαντικά πλεονεκτήματα που προσφέρει, ο XGBoost παρουσιάζει περιορισμό στην ερμηνευσιμότητα των αποτελεσμάτων του, καθώς η αρχιτεκτονική του βασίζεται σε μια ensemble προσέγγιση, όπου η τελική πρόβλεψη προκύπτει από τον συνδυασμό πολλών δέντρων απόφασης. Αν και αυτή η διαδικασία συμβάλλει στην υψηλή ακρίβεια του μοντέλου, καθιστά ταυτόχρονα δύσκολη την κατανόηση του τρόπου με τον οποίο κάθε χαρακτηριστικό επηρεάζει τη λήψη της τελικής απόφασης [22]. Ως αποτέλεσμα, ο XGBoost συχνά χαρακτηρίζεται ως μοντέλο «μαύρου κουτιού» (black-box model), ιδιαίτερα σε εφαρμογές όπου η διαφάνεια και η δυνατότητα αιτιολόγησης των προβλέψεων είναι σημαντικές.

Για την αντιμετώπιση αυτού του προβλήματος έχουν αναπτυχθεί διάφορες τεχνικές του πεδίου της Εξηγήσιμης Τεχνητής Νοημοσύνης (Explainable Artificial Intelligence – XAI), οι οποίες επιτρέπουν την καλύτερη κατανόηση και ερμηνεία της συμπεριφοράς σύνθετων μοντέλων μηχανικής μάθησης. Μεταξύ αυτών, ιδιαίτερα διαδεδομένη είναι η μέθοδος SHAP), η οποία βασίζεται στη θεωρία παιγνίων και παρέχει ποσοτικές εκτιμήσεις σχετικά με τη συνεισφορά κάθε χαρακτηριστικού στην τελική πρόβλεψη του μοντέλου. Μέσω του SHAP καθίσταται δυνατή η ανάλυση τόσο της συνολικής συμπεριφοράς του μοντέλου όσο και των επιμέρους προβλέψεων, ενισχύοντας σημαντικά τη διαφάνεια και την αξιοπιστία των αποτελεσμάτων. Η μέθοδος αυτή παρουσιάζεται και αναλύεται διεξοδικά σε επόμενο κεφάλαιο της παρούσας εργασίας [19].

3. Σύνολο δεδομένων

Η επιλογή του κατάλληλου συνόλου δεδομένων αποτελεί έναν από τους πιο σημαντικούς παράγοντες για την ανάπτυξη αξιόπιστων συστημάτων ανίχνευσης εισβολών βασισμένων σε τεχνικές μηχανικής μάθησης. Όπως αναφέρεται και στη σχετική βιβλιογραφία, τα δημόσια σύνολα δεδομένων που διατίθενται για συστήματα IDS συχνά εμπεριέχουν ασύμβατες ή κατηγορικές μεταβλητές που αδυνατούν να επεξεργαστούν συγκεκριμένοι αλγόριθμοι, με αποτέλεσμα να παρεμποδίζεται η εκπαίδευση και η διαδικασία λήψης αποφάσεων του μοντέλου [5]. Επιπλέον, τα δημόσια διαθέσιμα σύνολα δεδομένων για συστήματα IDS συχνά εμπεριέχουν ανακολουθίες, ακραίες τιμές ή χαρακτηριστικά μηδενικής διακύμανσης, τα οποία δεν συνεισφέρουν στη διαδικασία ταξινόμησης και απαιτούν κατάλληλη προεπεξεργασία πριν από τη χρήση τους [22].

Για τις ανάγκες της παρούσας εργασίας επιλέχθηκαν δύο από τα πλέον καθιερωμένα και ευρέως χρησιμοποιούμενα σύνολα δεδομένων στον τομέα ανίχνευσης δικτυακών εισβολών, το CIC-IDS2017 και το UNSW-NB15. Η επιλογή αυτή δεν ήταν τυχαία, καθώς και τα δύο αυτά σύνολα εμφανίζονται κατ' επανάληψη στη σύγχρονη βιβλιογραφία ως σημεία αναφοράς για την αξιολόγηση των μοντέλων μηχανικής μάθησης στον τομέα της κυβερνοασφάλειας [5], ενώ παράλληλα καλύπτουν διαφορετικές προελεύσεις, μεθοδολογίες παραγωγής και κατηγορίες επιθέσεων, γεγονός που επιτρέπει την αξιολόγηση του συστήματος σε ετερογενή και ρεαλιστικά σενάρια.

Το CIC-IDS2017 δημιουργήθηκε από το Canadian Institute for Cybersecurity (CIC) του Πανεπιστημίου του New Brunswick (UNB). Η παραγωγή των δεδομένων βασίστηκε στο σύστημα B-Profile, το οποίο προσομοίωσε την αφηρημένη συμπεριφορά 25 χρηστών που αλληλεπιδρούσαν με πρωτόκολλα όπως HTTPS, HTTP, SSH, FTP και ηλεκτρονικό ταχυδρομείο [5]. Για την υποδομή του συνόλου δεδομένων αξιοποιήθηκαν διάφορα στοιχεία του δικτύου όπως firewalls, modems, switches και routers, ενώ τα λειτουργικά συστήματα που χρησιμοποιήθηκαν ήταν Ubuntu, Mac OS X και Windows. Μέσω του εργαλείου CICFlowMeter εξήχθησαν συνολικά 80 χαρακτηριστικά ροής δικτύου, που διαχωρίζουν την κανονική από την κακόβουλη κίνηση, καλύπτοντας μεταξύ άλλων κατηγορίες επιθέσεων όπως DoS, Brute Force, Web Attack, Botnet και PortScan [5]. Το CIC-IDS2017 είναι οργανωμένο ανά ημέρα εργασίας, από Δευτέρα έως Παρασκευή, με κάθε ημέρα να αντιστοιχεί σε συγκεκριμένα σενάρια επιθέσεων, γεγονός που επιτρέπει την

επιλογή στοχευμένων υποσυνόλων ανάλογα με τις κατηγορίες επιθέσεων που αφορούν την εκάστοτε εφαρμογή.

Το UNSW-NB15 δημιουργήθηκε από το Cyber Range Lab του Πανεπιστημίου της Νέας Νότιας Ουαλίας (UNSW Canberra), υπό την αιγίδα του Australian Cyber Security Center (ACSC) [5]. Για την παραγωγή των ακατέργαστων πακέτων δικτύου χρησιμοποιήθηκε το εργαλείο IXIA Perfect Storm, το οποίο προσομοίωσε τόσο πραγματική καθημερινή κίνηση όσο και συνθετικά, σύγχρονα σενάρια επιθέσεων [6]. Για την εξαγωγή των χαρακτηριστικών αξιοποιήθηκαν τα εργαλεία Argus και Bro-IDS, ενώ για την καταγραφή της κίνησης χρησιμοποιήθηκε το tcpdump, παράγοντας συνολικά 100 GB ακατέργαστων δεδομένων [5]. Το σύνολο δεδομένων περιλαμβάνει 49 χαρακτηριστικά κατανεμημένα σε έξι κατηγορίες: ροής, βασικά, περιεχομένου, χρόνου, γενικού σκοπού και σύνδεσης (flow features, basic features, content features, time features, general purpose features, connection features), ενώ οι κατηγορίες επιθέσεων που καλύπτουν είναι οι Analysis, DoS, Exploits, Fuzzers, Backdoor, Generic, Shellcode, Reconnaissance και Worms [5]. Το σύνολο εκπαίδευσης περιλαμβάνει 175.341 εγγραφές και το σύνολο ελέγχου 82.332, με αξιοσημείωτα ισορροπημένη κατανομή μεταξύ φυσιολογικής και κακόβουλης κίνησης [6].

Αξίζει να σημειωθεί ότι, παρά τις διαφορές στον τρόπο παραγωγής τους, τα δύο σύνολα δεδομένων παρουσιάζουν σημαντικές ομοιότητες στη δομή των χαρακτηριστικών τους. Συγκεκριμένα, περίπου το 47% των χαρακτηριστικών του UNSW-NB15 εντοπίζεται και στο CIC-IDS2017, με τις κατηγορίες flow features και basic features να παρουσιάζουν τη μεγαλύτερη αλληλεπικάλυψη, 100% και 85% αντίστοιχα [5]. Αυτού του είδους η μερική συνάφεια καθιστά τα δύο σύνολα συμπληρωματικά, επιτρέποντας τη σύγκριση και την αξιολόγηση μοντέλων σε ένα ευρύτερο πλαίσιο.

3.1. Περιγραφή του συνόλου δεδομένων

Το CIC-IDS2017 είναι ένα από τα πλέον αξιόπιστα και ευρέως αναγνωρισμένα σύνολα δεδομένων στον χώρο της ανίχνευσης δικτυακών εισβολών και διατίθεται δημόσια για έρευνα και ανάπτυξη συστημάτων IDS [22]. Το σύνολο δεδομένων παρέχει πραγματικές ροές δικτύου που περιλαμβάνουν τόσο φυσιολογική (Benign) όσο και κακόβουλη κίνηση, καλύπτοντας με αυτό το τρόπο ένα ευρύ φάσμα κατηγοριών επιθέσεων όπως DoS, DDoS, Brute Force, Web Attack, PortScan, BotNet, Infiltration και Heartbleed [5].

Το σύνολο του CIC-IDS2017 είναι δομημένο ανά ημέρα εργασίας, από Δευτέρα έως Παρασκευή, με κάθε ημέρα να αντιστοιχεί σε διαφορετικά σενάρια επιθέσεων [16]. Αυτή η οργάνωση ανά ημέρα δεν είναι τυχαία, καθώς κάθε αρχείο CSV αντιπροσωπεύει ένα θεματικά αυτόνομο υποσύνολο, στο οποίο τα σενάρια επίθεσης είναι απομονωμένα, επιτρέποντας έτσι στον ερευνητή να επιλέξει στοχευμένα τις κατηγορίες επιθέσεων που τον ενδιαφέρουν χωρίς να επηρεάζεται από άσχετες κατηγορίες. Η πρακτική αυτή είναι ευρέως αποδεκτή, καθώς όπως αναφέρεται και στη σχετική βιβλιογραφία, [16] μπορεί να χρησιμοποιηθεί αποκλειστικά το Thursday-morning για Web Attacks, οι [14] επιλέγουν μόνο το Friday-PortScan για την ανάλυση επιθέσεων Port Scan, ενώ στο [22] εστιάζει στο Wednesday για επιθέσεις DoS. Αυτή, η κατά ημέρα διαχωρισμένη δομή, αποφεύγει την ανάμειξη ετερόκλητων σεναρίων επίθεσης σε ένα ενιαίο αρχείο, μειώνοντας τον κίνδυνο εισαγωγής θορύβου κατά την εκπαίδευση. Αυτός είναι και ο λόγος για τον οποίο στην παρούσα εργασία δεν χρησιμοποιήθηκε όλο το σύνολο δεδομένων, αλλά τρία επιμέρους CSV αρχεία, καθένα από τα οποία αντιστοιχεί σε μία συγκεκριμένη κατηγορία επιθέσεων που αποτελεί στόχο ανίχνευσης του Shield IDS.

Αναλυτικότερα, χρησιμοποιήθηκαν τα εξής τρία αρχεία:

1. Wednesday-workingHours_cleaned.csv: Το αρχείο αυτό αντιστοιχεί στην Τετάρτη 5 Ιουλίου 2017 και αποτελεί το μεγαλύτερο υποσύνολο του συνόλου από πλευράς όγκου δεδομένων. Περιλαμβάνει κανονική κίνηση δικτύου (Benign) καθώς και τις παραλλαγές επιθέσεων τύπου άρνησης υπηρεσιών (DoS) DoS Hulk (231.073 εγγραφές), DoS GoldenEye (10.293), DoS Slowloris (5.796) και DoS Slowhttptest (5.499), έναντι 440.031 εγγραφών φυσιολογικής κίνησης [22]. Το αρχείο αυτό είναι το βασικό σύνολο δεδομένων αναφοράς για την εκπαίδευση του μοντέλου στην αναγνώριση επιθέσεων DoS.

2. Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv: Το αρχείο αυτό αντιστοιχεί στο πρωινό τμήμα της Πέμπτης 6 Ιουλίου 2017 και περιέχει κανονική κίνηση δικτύου μαζί με επιθέσεις τύπου Web Attack και πιο συγκεκριμένα Brute Force (1.507 εγγραφές), XSS Cross-Site Scripting (652 εγγραφές) και SQL Injection (21 εγγραφές), έναντι 168.186 εγγραφών Benign κίνησης [16]. Αξίζει να σημειωθεί ότι ο συγκεκριμένος τύπος επιθέσεων παρουσιάζει έντονη ανισορροπία κλάσεων, γεγονός που καθιστά ιδιαίτερα απαιτητική την εκπαίδευση μοντέλων για την κατηγορία αυτή.

3. Friday-WorkingHours-Afternoon-PortScan_cleaned.csv: Το αρχείο αυτό αντιστοιχεί στο απογευματινό τμήμα της Παρασκευής 7 Ιουλίου 2017 και συνδυάζει κανονική κίνηση δικτύου με επιθέσεις τύπου Port Scan. Το αρχείο περιλαμβάνει συνολικά 127.537 εγγραφές κανονικής κίνησης και 158.930 εγγραφές επιθέσεων PortScan, καθιστώντας το σχετικά ισορροπημένο σε επίπεδο κατανομής κλάσεων [14].

Όπως προαναφέρθηκε, το UNSW-NB15 περιλαμβάνει 49 χαρακτηριστικά τα οποία διαχωρίζονται στις κατηγορίες ροής, βασικών, περιεχομένου, χρόνου, γενικού σκοπού και σύνδεσης [5], ενώ καλύπτονται εννέα διακριτές οι κατηγορίες επιθέσεων Analysis, DoS, Exploits, Fuzzers, Backdoor, Generic, Shellcode, Reconnaissance και Worms, σε συνδυασμό με κανονική κίνηση δικτύου [5]. Το σύνολο εκπαίδευσης αποτελείται από 175.341 εγγραφές και το σύνολο ελέγχου από 82.332 εγγραφές, με σχετικά ισορροπημένη κατανομή μεταξύ φυσιολογικής και κακόβουλης κίνησης [6].

Ιδιαίτερα σημαντικό πλεονέκτημα του UNSW-NB15 είναι η εκτενής τεκμηρίωση των χαρακτηριστικών του και η ύπαρξη μετα-δεδομένων που διευκολύνουν την ανάλυση και την επιλογή των χαρακτηριστικών [26]. Επιπλέον, το συγκεκριμένο σύνολο χρησιμοποιείται ευρέως στη βιβλιογραφία για την αξιολόγηση της γενικευσιμότητας των μοντέλων IDS, καθώς τα σενάρια επίθεσης που περιέχει διαφέρουν αρκετά από αυτά του CIC-IDS2017, επιτρέποντας έτσι τη σύγκριση της απόδοσης του μοντέλου σε εντελώς διαφορετικά περιβάλλοντα [5].

3.2. Ανάλυση χαρακτηριστικών

Τα χαρακτηριστικά του CIC-IDS2017 εξήχθησαν μέσω του εργαλείου CICFlowMeter, το οποίο αναλύει αμφίδρομες (bidirectional) ροές δικτύου και συλλέγει πληροφορίες όπως οι διευθύνσεις IP προέλευσης και προορισμού, οι θύρες επικοινωνίας, η διάρκεια, το μήκος και ο αριθμός των πακέτων, ανώ παράλληλα υπολογίζει και στατιστικά μεγέθη όπως η τυπική απόκλιση του χρόνου άφιξης των πακέτων και οι μετρήσεις των σημαιών (flag counts) [5]. Συνολικά το σύνολο δεδομένων περιλαμβάνει 80 χαρακτηριστικά ροής, τα οποία μπορούν να κατηγοριοποιηθούν σε τέσσερις βασικές ομάδες ανάλογα με τη φύση της πληροφορίας που αποτυπώνουν. Η πρώτη ομάδα αφορά τα χαρακτηριστικά όγκου και μήκους των πακέτων, τα οποία καταγράφουν τον συνολικό αριθμό και το μέγεθος των πακέτων που ανταλλάσσονται και στις δύο κατευθύνσεις της ροής. Στην κατηγορία αυτή ανήκουν χαρακτηριστικά όπως τα Total Fwd Packets και Total Bwd Packets (συνολικός

αριθμός πακέτων προς τα εμπρός και προς τα πίσω), TotLen Fwd Pkts / TotLen Bwd Pkts (συνολικό μέγεθος σε bytes), Max Pkt Len, Min Pkt Len, Pkt Len Mean και Pkt Len Std (μέγιστο, ελάχιστο, μέσο και τυπική απόκλιση μήκους πακέτων), καθώς και Avg Pkt Size, Avg Fwd Seg Size και Avg Bwd Seg Size (μέσα μεγέθη τμημάτων). Η δεύτερη ομάδα περιλαμβάνει τα χαρακτηριστικά χρονισμού και ρυθμού της ροής, τα οποία αποτυπώνουν τη χρονική συμπεριφορά της κίνησης και αποτελούν ιδιαίτερα σημαντικούς δείκτες για την αναγνώριση της κακόβουλης δραστηριότητας. Χαρακτηριστικά όπως τα Fwd IAT Mean, Fwd IAT Std, Fwd IAT Min (μέσος χρόνος, τυπική απόκλιση και ελάχιστος χρόνος μεταξύ των διαδοχικών πακέτων προς τα εμπρός), Bwd IAT Mean και Bwd IAT Std (αντίστοιχα για την αντίστροφη κατεύθυνση), καθώς και Flow Bytes/s (ρυθμός μεταφοράς των bytes ανά δευτερόλεπτο) ανήκουν σε αυτή την κατηγορία. Η σημασία τους είναι ιδιαίτερα υψηλή, καθώς όπως παρουσιάζεται και σε μελέτες της βιβλιογραφίας, χαρακτηριστικά όπως τα Flow IAT Mean, Flow IAT Max και Flow IAT Std κατατάσσονται μεταξύ των πιο επιδραστικών για την ταξινόμηση επιθέσεων [16]. Η τρίτη ομάδα καλύπτει τα χαρακτηριστικά των σημαιών TCP (TCP flags), τα οποία αντικατοπτρίζουν τον τύπο και την κατάσταση των συνδέσεων. Στα χαρακτηριστικά αυτά περιλαμβάνονται τα SYN Flag Count (πακέτα έναρξης σύνδεσης), ACK Flag Count (πακέτα επιβεβαίωσης), FIN Flag Count (πακέτα τερματισμού σύνδεσης), PSH Flag Count (άμεση προώθηση δεδομένων) και URG Flag Count (πακέτα προτεραιότητας). Η ανάλυση αυτών των σημαιών είναι χρήσιμη για την αναγνώριση μη φυσιολογικής συμπεριφοράς σύνδεσης, όπως συμβαίνει σε επιθέσεις DoS που αξιοποιούν μαζικές αιτήσεις SYN. Η τέταρτη ομάδα περιλαμβάνει τα χαρακτηριστικά παραθύρου και υποροών (subflows), όπως τα Init_Win_bytes_fwd και Init_Win_bytes_bwd (αρχικό μέγεθος παραθύρου TCP), Subflow Fwd Pkts, Subflow Bwd Pkts, Subflow Fwd Bytes και Subflow Bwd Bytes (μέσος αριθμός πακέτων και bytes ανά υποροή), καθώς και act_data_pkt_fwd (αριθμός πακέτων με ωφέλιμο φορτίο). Τα χαρακτηριστικά ενεργούς και αδρανούς κατάστασης όπως τα Active Mean, Active Std, Active Min και Idle Mean αποτυπώνουν τη δυναμική συμπεριφορά της ροής σε επίπεδο χρόνου [22].

Τα χαρακτηριστικά του UNSW-NB15 εξήχθησαν μέσω των εργαλείων Argus και Bro-IDS και είναι οργανωμένα σε έξι διακριτές κατηγορίες [5], οι οποίες αποτυπώνουν διαφορετικές πτυχές της κίνησης δικτύου σε σύγκριση με το CIC-IDS2017.

Τα χαρακτηριστικά ροής (5 στο σύνολο) περιλαμβάνουν βασικές πληροφορίες ροής όπως srcip (διεύθυνση IP πηγής), dstip (διεύθυνση IP προορισμού), sport / dsport (θύρες επικοινωνίας πηγής και προορισμού) και proto (πρωτόκολλο μεταφοράς). Τα βασικά χαρακτηριστικά (13 συνολικά) αποτυπώνουν στατιστικά χαρακτηριστικά της ροής, όπως sbytes / dbytes (bytes πηγής και προορισμού), Sload / Dload (ρυθμός μετάδοσης πηγής και προορισμού), Dpkts (πακέτα προορισμού), sloss / dloss (χαμένα πακέτα πηγής και προορισμού) και state (κατάσταση σύνδεσης). Τα χαρακτηριστικά περιεχομένου (8 συνολικά) σχετίζονται με πληροφορίες επιπέδου εφαρμογής, όπως service (αναγνωρισμένη υπηρεσία δικτύου π.χ. http, ftp, ssh) και is_sm_ips_ports (ταύτιση διευθύνσεων πηγής-προορισμού). Τα χαρακτηριστικά χρόνου (9 χαρακτηριστικά) περιλαμβάνουν χρονικά χαρακτηριστικά όπως Timestamp, flow_start_ts, ackdat (χρόνος SYN-ACK), tcprtt (TCP Round Trip Time) και detect_ts / detection_delay. Τέλος, τα χαρακτηριστικα γενικού σκοπού και σύνδεσης (12 χαρακτηριστικά συνολικά) εμπεριέχουν πρόσθετα περιγραφικά στοιχεία της σύνδεσης [5].

Αξίζει να επισημανθεί ότι, σε αντίθεση με το CIC-IDS2017, το UNSW-NB15 περιλαμβάνει ρητά χαρακτηριστικά επιπέδου εφαρμογής (π.χ. service) και χρονικής ανίχνευσης (π.χ. detection_delay), γεγονός που προσφέρει μια διαφορετική οπτική για την ανάλυση της κακόβουλης συμπεριφοράς. Παράλληλα, περίπου το 47% των χαρακτηριστικών του UNSW-NB15 εμφανίζει αντιστοιχία με χαρακτηριστικά του CIC-IDS2017, με τις κατηγορίες χαρακτηριστικών ροής και βασικών χαρακτηριστικών να παρουσιάζουν την υψηλότερη αλληλεπικάλυψη, 100% και 85% αντίστοιχα [5].

Για την πλήρη κατανόηση όλων των χαρακτηριστικών που χρησιμοποιήθηκαν, παρατίθεται αναλυτικός κατάλογος (Πίνακας 1) με την ονομασία και την περιγραφή κάθε χαρακτηριστικού και των δύο συνόλων δεδομένων.

Σύντομη Ονομασία	Πλήρης Ονομασία	Περιγραφή / Σημασία
ACK Flag Count	Acknowledgment Flag Count	Ο αριθμός των πακέτων με τη σημαία ACK (επιβεβαίωση λήψης).
ackdat	Acknowledgment Data Time	Ο χρόνος μεταξύ του SYN-ACK και του ACK στο TCP handshake (ο χρόνος που χρειάζεται ο προορισμός για να απαντήσει).
act_data_pkt_fwd	Actual Data Packets Forward	Ο αριθμός των πακέτων στην κατεύθυνση προς τα εμπρός που περιέχουν τουλάχιστον 1 byte ωφέλιμου φορτίου (TCP payload).
Active Min	Minimum Active Time	Ο ελάχιστος χρόνος που η ροή ήταν ενεργή πριν γίνει αδρανής.
Active Mean	Mean Active Time	Ο μέσος χρόνος που η ροή ήταν ενεργή πριν γίνει αδρανής.
Active Std	Active Time Standard Deviation	Η τυπική απόκλιση του χρόνου που η ροή ήταν ενεργή.
Avg Bwd Seg Size	Average Backward Segment Size	Το μέσο μέγεθος των τμημάτων στην κατεύθυνση προς τα πίσω.
Avg Fwd Seg Size	Average Forward Segment Size	Το μέσο μέγεθος των τμημάτων (segments) στην κατεύθυνση προς τα εμπρός.
Avg Pkt Size	Average Packet Size	Το μέσο μέγεθος όλων των πακέτων στη ροή.
Bwd IAT Mean	Backward Inter-Arrival Time Mean	Ο μέσος χρόνος μεταξύ της άφιξης δύο διαδοχικών πακέτων προς τα πίσω.
Bwd IAT Std	Backward Inter-Arrival Time Std Dev	Η τυπική απόκλιση του χρόνου μεταξύ των πακέτων προς τα πίσω.
Bwd Pkt Len Max	Backward Packet Length Maximum	Το μέγιστο μήκος πακέτου στην κατεύθυνση προς τα πίσω.
Bwd Pkt Len Mean	Backward Packet Length Mean	Η μέση τιμή του μήκους των πακέτων προς τα πίσω.
Bwd Pkt Len Std	Backward Packet Length Standard Deviation	Η τυπική απόκλιση του μήκους των πακέτων προς τα πίσω.
dbytes	Destination Bytes	Συνολικός αριθμός bytes που στάλθηκαν από τον προορισμό προς την πηγή.
detect_ts	Detection Timestamp	Η χρονική στιγμή κατά την οποία το σύστημα ανίχνευσης (π.χ. IDS) εντόπισε τη δραστηριότητα.
detection_delay	Detection Delay	Η καθυστέρηση (σε χρόνο) μεταξύ της έναρξης της δραστηριότητας και της ανίχνευσής της.
Dload	Destination Load	Ο ρυθμός μετάδοσης του προορισμού σε bits ανά δευτερόλεπτο.
dloss	Destination Packets Loss	Ο αριθμός των πακέτων που χάθηκαν κατά τη μετάδοση από τον προορισμό.

Dpkts	Destination Packets	Ο συνολικός αριθμός πακέτων που στάλθηκαν από τον προορισμό προς την πηγή.
dsport	Destination Port	Ο αριθμός θύρας (port) στον οποίο στοχεύει η κίνηση στον προορισμό.
dstip	Destination IP Address	Η διεύθυνση IP του προορισμού (πού κατευθύνεται η κίνηση).
FIN Flag Count	Finish Flag Count	Ο αριθμός των πακέτων με τη σημαία FIN (αίτημα τερματισμού σύνδεσης).
Flow Bytes/s	Flow Bytes per Second	Ο ρυθμός μεταφοράς δεδομένων (throughput) σε bytes ανά δευτερόλεπτο.
flow_start_ts	Flow Start Timestamp	Η ακριβής ώρα έναρξης της συγκεκριμένης ροής δεδομένων (flow).
Fwd Header Len	Forward Header Length	Το συνολικό μέγεθος των επικεφαλίδων (headers) στην κατεύθυνση προς τα εμπρός.
Fwd IAT Min	Forward Inter-Arrival Time Minimum	Ο ελάχιστος χρόνος μεταξύ της άφιξης δύο διαδοχικών πακέτων προς τα εμπρός.
Fwd IAT Std	Forward Inter-Arrival Time Std Dev	Η τυπική απόκλιση του χρόνου μεταξύ των πακέτων προς τα εμπρός.
Fwd Pkt Len Mean	Forward Packet Length Mean	Η μέση τιμή του μήκους των πακέτων προς τα εμπρός.
Idle Mean	Idle Time Mean	Ο μέσος χρόνος που η ροή παρέμεινε σε κατάσταση αδράνειας.
Init_Win_bytes_bwd	Initial Window Bytes Backward	Ο αριθμός των bytes που στάλθηκαν στο αρχικό παράθυρο προς τα πίσω.
Init_Win_bytes_fwd	Initial Window Bytes Forward	Ο αριθμός των bytes που στάλθηκαν στο αρχικό παράθυρο (window size) προς τα εμπρός.
is_sm_ips_ports	If Source Equals Destination (IP and port)	Υποδεικνύει αν οι διευθύνσεις και πόρτες προέλευσης – προορισμού είναι ίδιες.
Max Pkt Len	Maximum Packet Length	Το μέγεθος του μεγαλύτερου πακέτου που παρατηρήθηκε.
Min Pkt Len	Minimum Packet Length	Το μέγεθος του μικρότερου πακέτου που παρατηρήθηκε.
Pkt Len Mean	Packet Length Mean	Η μέση τιμή του μήκους των πακέτων.
Pkt Len Std	Packet Length Standard Deviation	Η τυπική απόκλιση του μήκους των πακέτων (δείχνει τη μεταβλητότητα στο μέγεθος).
proto	Protocol	Το πρωτόκολλο μεταφοράς που χρησιμοποιείται (π.χ. TCP, UDP, ICMP).
PSH Flag Count	Push Flag Count	Ο αριθμός των πακέτων με τη σημαία PSH (άμεση προώθηση δεδομένων στην εφαρμογή).

sbytes	Source Bytes	Συνολικός αριθμός bytes που στάλθηκαν από την πηγή προς τον προορισμό.
service	Service	Η συγκεκριμένη υπηρεσία δικτύου που αναγνωρίστηκε (π.χ. http, ftp, ssh, dns).
Sload	Source Load	Ο ρυθμός μετάδοσης της πηγής σε bits ανά δευτερόλεπτο.
sloss	Source Packets Loss	Ο αριθμός των πακέτων που χάθηκαν κατά τη μετάδοση από την πηγή.
sport	Source Port	Ο αριθμός θύρας (port) που χρησιμοποιεί η πηγή για τη σύνδεση.
srcip	Source IP Address	Η διεύθυνση IP της πηγής (από πού ξεκινά η κίνηση).
state	State	Η κατάσταση της σύνδεσης τη στιγμή της καταγραφής (π.χ. FIN, CON, REQ, RST).
Subflow Bwd Bytes	Subflow Backward Bytes	Ο μέσος αριθμός bytes σε μια υποροή στην κατεύθυνση προς τα πίσω.
Subflow Bwd Pkts	Subflow Backward Packets	Ο μέσος αριθμός πακέτων υποροής στην κατεύθυνση προς τα πίσω.
Subflow Fwd Pkts	Subflow Forward Packets	Ο μέσος αριθμός πακέτων υποροής στην κατεύθυνση προς τα εμπρός
Subflow Fwd Bytes	Bytes Subflow Forward Bytes	Ο μέσος αριθμός bytes σε μια υποροή στην κατεύθυνση προς τα εμπρός
SYN Flag Count	Synchronize Flag Count	Ο αριθμός των πακέτων με τη σημαία SYN (έναρξη σύνδεσης).
tcprtt	TCP Round Trip Time	Ο συνολικός χρόνος που απαιτείται για την ολοκλήρωση ενός κύκλου επικοινωνίας TCP (άθροισμα setup time και latency).
Timestamp	Timestamp	Η ημερομηνία και η ώρα καταγραφής του συγκεκριμένου συμβάντος ή πακέτου.
Total Bwd Packets	Total Backward Packets	Συνολικός αριθμός πακέτων στην κατεύθυνση προς τα πίσω. (προορισμός -> πηγή).
Total Fwd Packets	Total Forward Packets	Συνολικός αριθμός πακέτων στην κατεύθυνση προς τα εμπρός (πηγή -> προορισμός).
TotLen Bwd Pkts	Total Length of Bwd Packets	Το συνολικό μέγεθος (σε bytes) όλων των πακέτων στην κατεύθυνση προς τα πίσω.
TotLen Fwd Pkts	Total Length of Fwd Packets	Το συνολικό μέγεθος (σε bytes) όλων των πακέτων στην κατεύθυνση προς τα εμπρός.
URG Flag Count	Urgent Flag Count	Ο αριθμός των πακέτων με τη σημαία URG (δεδομένα προτεραιότητας).

Πίνακας 1 Ανάλυση χαρακτηριστικών των cicids 2017 και unsbw 15

3.3. Αφαίρεση μη σημαντικών στηλών

Ένα ιδιαίτερα σημαντικό βήμα πριν από την εκπαίδευση οποιουδήποτε μοντέλου μηχανικής μάθησης είναι η αφαίρεση των χαρακτηριστικών που δεν προσφέρουν χρήσιμη πληροφορία για τη διαδικασία ταξινόμησης. Τέτοια χαρακτηριστικά, όπως για παράδειγμα όσα φέρουν μηδενική διακύμανση (zero-variance features), λαμβάνουν την ίδια τιμή σε όλες τις εγγραφές του συνόλου δεδομένων και συνεπώς αδυνατούν να συμβάλουν στη διαχωριστική ικανότητα του μοντέλου [22]. Ειδικότερα στα δενδρικά μοντέλα μηχανικής μάθησης, χαρακτηριστικά μηδενικής διακύμανσης δεν συμμετέχουν στον διαχωρισμό κόμβων, καθώς αφενός αυξάνουν το υπολογιστικό κόστος χωρίς αντίστοιχο όφελος και αφετέρου ενδέχεται να παραμορφώσουν την εκτιμώμενη σημαντικότητα των υπολοίπων χαρακτηριστικών [22]. Στο πλαίσιο της παρούσας εργασίας, η αφαίρεση των μη σημαντικών στηλών πραγματοποιήθηκε μέσω του ειδικού εργαλείου Remove Columns που αναπτύχθηκε στην εφαρμογή Shield IDS.

Για τα τρία CSV αρχεία του CIC-IDS2017 που χρησιμοποιήθηκαν αφαιρέθηκαν συνολικά τα παρακάτω 11 χαρακτηριστικά, τα οποία εντοπίστηκαν είτε ως χαρακτηριστικά μηδενικής διακύμανσης είτε ως αναγνωριστικά πεδία χωρίς πληροφοριακή αξία για την εκπαίδευση. Η επιλογή αυτή βασίστηκε στη μεθοδολογία που ακολουθεί η σχετική βιβλιογραφία [16].

- Dst Port (Destination Port): Η θύρα προορισμού αποτελεί αναγνωριστικό πεδίο σύνδεσης και όχι περιγραφικό χαρακτηριστικό της συμπεριφοράς της ροής. Αφαιρείται διότι η συμπερίληψή του σε περιπτώσεις συγκεκριμένων θυρών που απλώς συνδέονται με συγκεκριμένα πρωτόκολλα μπορεί να οδηγήσει σε υπερπροσαρμογή, χωρίς να αντικατοπτρίζει την πραγματική κακόβουλη συμπεριφορά [16].
- Bwd PSH Flags: Μετρά τον αριθμό πακέτων με τη σημαία PSH στην αντίστροφη κατεύθυνση. Στο συγκεκριμένο σύνολο δεδομένων λαμβάνει μία και μόνο τιμή σε όλες τις εγγραφές, καθιστώντας το χαρακτηριστικό μηδενικής διακύμανσης και άρα πλήρως αχρείαστο [16], [22].
- Fwd URG Flags: Υπολογίζει τα πακέτα με σημαία URG (urgent) στην κατεύθυνση προς τα εμπρός. Εμφανίζει επίσης μηδενική διακύμανση στο σύνολο δεδομένων και αφαιρείται για τον ίδιο λόγο [16], [22].
- Fwd Header Length (Fwd Header Len): Αναφέρεται στο συνολικό μέγεθος των επικεφαλίδων της προς τα εμπρός κατεύθυνσης. Παρουσιάζει μηδενική ή σχεδόν μηδενική διακύμανση και ταυτόχρονα αλληλεπικαλύπτεται πληροφοριακά με άλλα χαρακτηριστικά μήκους πακέτων [16].

- CWE Flag Count: Μετρά τα πακέτα με τη σημαία CWE (Congestion Window Reduced Echo). Λαμβάνει μία σταθερή τιμή σε όλο το σύνολο δεδομένων, καθιστώντας το χαρακτηριστικό πλήρως αδιάφορο για την εκπαίδευση [16], [22].
- Fwd Avg Bytes/Bulk (Fwd Bytes/b Avg), Fwd Avg Packets/Bulk (Fwd Pkts/b Avg), Fwd Avg Bulk Rate (Fwd Blk Rate Avg): Τα τρία αυτά χαρακτηριστικά αφορούν τα στατιστικά μαζικής μεταφοράς δεδομένων (bulk transfer) της προς τα εμπρός κατεύθυνσης. Στα συγκεκριμένα υποσύνολα του συνόλου δεδομένων λαμβάνουν σταθερή τιμή μηδέν σε όλες τις εγγραφές, καθώς δεν παρατηρείται μαζική μεταφορά, με αποτέλεσμα να μην προσφέρουν καμία πληροφορία για τη διαδικασία ταξινόμησης [16], [22].
- Bwd Avg Bytes/Bulk (Bwd Bytes/b Avg), Bwd Avg Packets/Bulk (Bwd Pkts/b Avg), Bwd Avg Bulk Rate (Bwd Blk Rate Avg): Αντίστοιχα με τα παραπάνω, τα τρία αυτά χαρακτηριστικά αφορούν τη μαζική μεταφορά της αντίστροφης όμως κατεύθυνσης και παρουσιάζουν επίσης μηδενική διακύμανση [16], [22].

Στον πίνακα που ακολουθεί παρουσιάζονται συγκεντρωτικά τα αφαιρούμενα χαρακτηριστικά με τη σύντομη αιτιολόγηση αφαίρεσής τους.

Χαρακτηριστικό	**Λόγος Αφαίρεσης**
Dst Port	Αναγνωριστικό πεδίο, κίνδυνος υπερπροσαρμογής
Bwd PSH Flags	Μηδενική διακύμανση
Fwd URG Flags	Μηδενική διακύμανση
Fwd Header Length	Μηδενική/χαμηλή διακύμανση, πλεονασμός
CWE Flag Count	Μηδενική διακύμανση
Fwd Avg Bytes/Bulk	Σταθερή τιμή μηδέν, μηδενική διακύμανση
Fwd Avg Packets/Bulk	Σταθερή τιμή μηδέν, μηδενική διακύμανση
Fwd Avg Bulk Rate	Σταθερή τιμή μηδέν, μηδενική διακύμανση
Bwd Avg Bytes/Bulk	Σταθερή τιμή μηδέν, μηδενική διακύμανση
Bwd Avg Packets/Bulk	Σταθερή τιμή μηδέν, μηδενική διακύμανση
Bwd Avg Bulk Rate	Σταθερή τιμή μηδέν, μηδενική διακύμανση

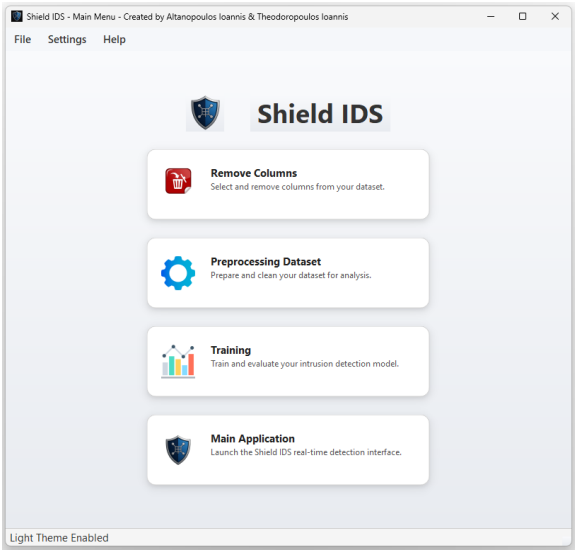
Πίνακας 2 Χαρακτηριστικά που αφαιρέθηκαν από το CIC-IDS2017 και αιτιολόγηση αφαίρεσης.

Μετά την αφαίρεση των 11 αυτών χαρακτηριστικών, κάθε CSV αρχείο του CIC-IDS2017 διατηρεί 69 χαρακτηριστικά (από τα αρχικά 80), τα οποία αποτελούν τη βάση για τα επόμενα στάδια της προεπεξεργασίας και της εκπαίδευσης του μοντέλου.

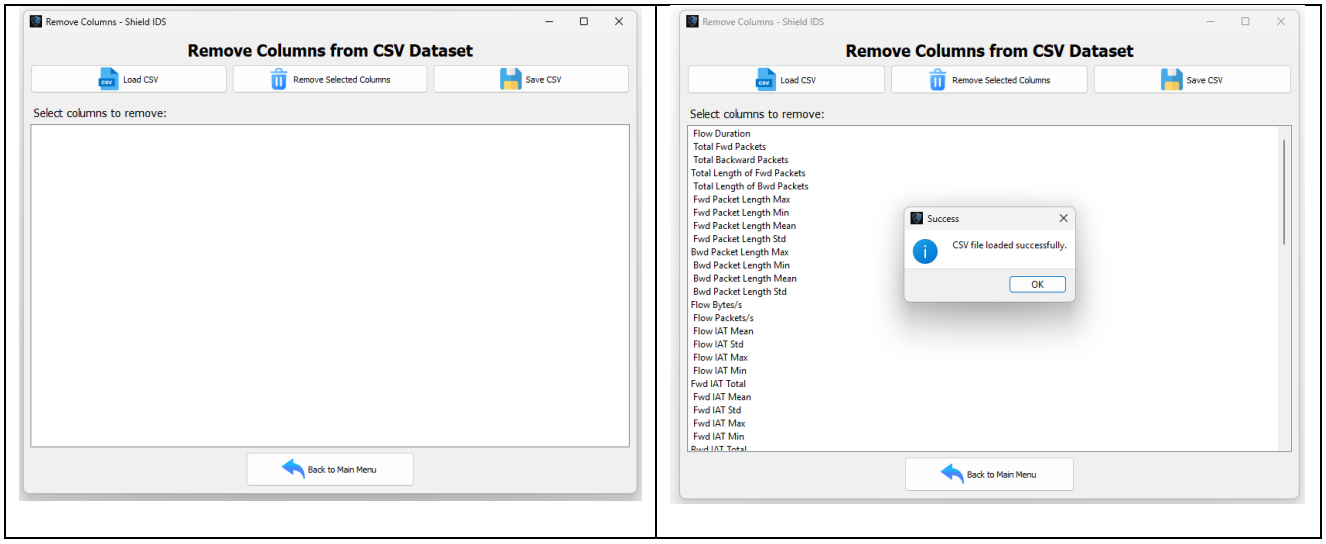
3.3.1. Διαδικασία και τεχνική αφαίρεσης

Όπως αναφέρθηκε παραπάνω, η αφαίρεση των μη σημαντικών στηλών πραγματοποιήθηκε μέσω του ειδικού εργαλείου Remove Columns που αναπτύχθηκε στην εφαρμογή Shield IDS κι επιλέγεται από την κεντρική αρχική οθόνη (όπως παρουσιάζεται στην Εικόνα 1).

Το εργαλείο αυτό επιτρέπει τη φόρτωση ενός CSV αρχείου (χαρακτηριστικό απόσπασμα στην Εικόνα 2), την επιλογή των στηλών που πρέπει να αφαιρεθούν μέσω διαδραστικής λίστας πολλαπλής επιλογής, και την αποθήκευση του καθαρισμένου αρχείου. Η λογική του υλοποιείται στον κώδικα της κλάσης RemoveColumnsApp μέσω της μεθόδου remove_columns(), η οποία χρησιμοποιεί τη συνάρτηση drop() της βιβλιοθήκης pandas για την αφαίρεση των επιλεγμένων στηλών από το DataFrame.



Εικόνα 1 Κεντρική αρχική οθόνη Shield IDS



Εικόνα 2 Αρχική εφαρμογή και εισαγωγή συνόλου δεδομένων

Η χρήση τριών διαφορετικών μεθόδων για τις αντίστοιχες λειτουργίες των κουμπιών (load_csv, remove_columns και save_csv, αποσπάσματα των οποίων ακολουθούν στην συνέχεια) αποτελεί βασική σχεδιαστική πρακτική του αντικειμενοστραφούς προγραμματισμού. Κάθε μέθοδος είναι απομονωμένη και εκτελεί αποκλειστικά ένα διακριτό έργο, έτσι η πρώτη διαχειρίζεται την ανάγνωση και την αρχική απεικόνιση των δεδομένων, η δεύτερη εκτελεί τους απαραίτητους μετασχηματισμούς και το φιλτράρισμα στη μνήμη, ενώ η τρίτη αναλαμβάνει την ασφαλή εξαγωγή και αποθήκευση του τελικού αποτελέσματος στο δίσκο. Αυτός ο διαχωρισμός προσφέρει μέγιστη διαμερισματοποίηση (modularity) στον κώδικα, καθιστώντας τον ευανάγνωστο, εύκολο στη συντήρηση και την αποσφαλμάτωση (debugging), καθώς τυχόν πρόβλημα σε μία λειτουργία (π.χ. ένα σφάλμα κατά την αποθήκευση) μπορεί να εντοπιστεί και να διορθωθεί τοπικά, χωρίς να επηρεάσει ή να περιπλέξει την υπόλοιπη ροή της εφαρμογής.

```
def load_csv(self):
      # Παράθυρο διαλόγου για επιλογή CSV και αποθήκευση διαδρομής
      path, _ = QtWidgets.QFileDialog.getOpenFileName(self, "Select CSV
            file", "", "CSV Files (*.csv)")
      # Αν ακυρωθεί η επιλογή
      if not path:
            return # Τερματισμός διαδικασίας

      # Διάβασμα CSV και αποθήκευση στη μεταβλητή self.df ως DataFrame
      self.df = pd.read_csv(path)
      # Εκκαθάριση από προηγούμενες στήλες
      self.columns_list.clear()
      # Μετατροπή ονόματων στηλών σε λίστα Python και προσθήκη στο GUI
      self.columns_list.addItems(self.df.columns.tolist())

      # Ενημερωτικό μήνυμα επιτυχίας
      QtWidgets.QMessageBox.information(self, "Success", "CSV file
            loaded successfully.")

def remove_columns(self):
      # Αν δεν έχει φορτωθεί κάποιο αρχείο
      if self.df is None:
            QtWidgets.QMessageBox.warning(self, "Warning", "No CSV file
```

```
                        has been loaded.") # Προειδοποιητικό μήνυμα
        return # Τερματισμός διαδικασίας

        # Δημιουργία λίστας με τα ονόματα στηλών που επιλέχθησαν
        selected = [item.text() for item in
              self.columns_list.selectedItems()]

        if not selected: # Αν δεν επιλέχθηκε στήλη
              QtWidgets.QMessageBox.information(self, "Information", "No
                    columns selected.") # Προειδοποιητικό μήνυμα
        return # Τερματισμός διαδικασίας

        # Διαγραφή επιλεγμένων στηλών από το DataFrame
        self.df.drop(columns=selected, inplace=True) # inplace: μόνιμα
        # Καθαρισμός τρέχουσας λίστα στηλών GUI
        self.columns_list.clear()
        # Προσθήκη εναπομείναντων στήλών στη λίστα του GUI
        self.columns_list.addItems(self.df.columns.tolist())

        # Μήνυμα επιτυχούς διαγραφής
        QtWidgets.QMessageBox.information(self, "Success", "Selected
                    columns were removed successfully.")

def save_csv(self):
        if self.df is None: # Αν δεν υπάρχουν δεδομένα προς αποθήκευση
              QtWidgets.QMessageBox.warning(self, "Warning", "There is no
                          CSV file to save.") # Προειδοποιητικό μήνυμα
              return # Τερματισμός διαδικασίας

        # Παράθυρο επιλογής τοποθεσίας και ονομασίας αποθήκευσης
        path, _ = QtWidgets.QFileDialog.getSaveFileName(self, "Save CSV
                    file", "", "CSV Files (*.csv)")
        # Αν ακυρωθεί η αποθήκευση
        if not path:
              return # Τερματισμός διαδικασίας

        try: # Μπλοκ αποθήκευσης
              # Αποθήκευση DataFrame σε αρχείο CSV στη διαδρομή
              self.df.to_csv(path, index=False)
```

```
            # Μήνυμα επιτυχούς αποθήκευσης
            QtWidgets.QMessageBox.information(self, "Success", "CSV file
                  saved successfully.")

      except Exception as e: # Σφάλμα κατά την αποθήκευση
         # Εμφάνιση μηνύματος αν αποτύχει η εγγραφή στο δίσκο
         QtWidgets.QMessageBox.critical(self, "Error", f"Failed to save
                  CSV file:\n{e}")
```

4. Προεπεξεργασία Δεδομένων

Η προεπεξεργασία δεδομένων αποτελεί ένα σημαντικό στάδιο στην ανάπτυξη συστημάτων μηχανικής μάθησης για την ανίχνευση εισβολών καθώς η ποιότητα του αρχικού συνόλου δεδομένων επηρεάζει άμεσα την ακρίβεια, τη σταθερότητα και την αποδοτικότητα των μοντέλων που παράγονται. Η βιβλιογραφία αναφέρει ότι τα σύνολα δεδομένων όπως τα UNSW-NB15 και CIC-IDS2017 συχνά περιέχουν ελλιπείς τιμές, τιμές που σημειώνονται ως NaN (μη αριθμητικές ή μη διαθέσιμες τιμές), άπειρες τιμές (Inf), διπλότυπες εγγραφές, ακραίες αριθμητικές τιμές και κατηγορικά χαρακτηριστικά. Τα παραπάνω δημιουργούν δυσκολίες στην εκπαίδευση των αλγορίθμων, αυξάνουν το υπολογιστικό κόστος και μπορούν να οδηγήσουν σε παραπλανητικά αποτελέσματα [5], [6]. Έτσι η εφαρμογή κατάλληλων τεχνικών προεπεξεργασίαςς θεωρείται αναγκαία πριν από οποιοδήποτε στάδιο μοντελοποίησης.

Πρόσφατες μελέτες δείχνουν ότι τεχνικές όπως ο καθαρισμός δεδομένων (data cleaning), η κωδικοποίηση κατηγορικών μεταβλητών, η κανονικοποίηση και η επιλογή χαρακτηριστικών επηρεάζουν ουσιαστικά την απόδοση των IDS, ενώ η μεταξύ τους αλληλεξάρτηση είναι συχνά ισχυρή ιδίως σε σύνολα δεδομένων που είναι βασισμένα σε ροές. Η βιβλιογραφία τονίζει επίσης ότι η επίδραση της προεπεξεργασίας δεν είναι ίδια για όλα τα προβλήματα, ούτε για όλα τα μοντέλα, γεγονός που καθιστά απαραίτητη την επιλογή μεθόδων που να ανταποκρίνονται στα χαρακτηριστικά του συνόλου δεδομένων [5], [6].

Βάσει των παραπάνω δημιουργήσαμε μια εφαρμογή την οποία ονομάσαμε Προεπεξεργασία του Συνόλου Δεδομένων (Preprocessing Dataset) και στην οποία χρησιμοποιούμε ως πρώτη τεχνική τη διαχείριση των διπλότυπων εγγραφών. Οι επαναλαμβανόμενες γραμμές μπορούν να παραμορφώσουν την εικόνα του συνόλου δεδομένων και να επηρεάσουν αρνητικά τη διαδικασία μάθησης. Σε σχετική μελέτη για την ανίχνευση DDoS αναφέρεται ρητά ότι οι διπλότυπες εγγραφές μπορούν να αλλοιώσουν τα συνολικά στατιστικά και να οδηγήσουν σε ανακριβή συμπεράσματα [13]. Αντίστοιχα για τα NIDS σύνολα δεδομένων έχει επισημανθεί ότι οι διπλότυπες εγγραφές ενδέχεται να οδηγήσουν σε λανθασμένες προβλέψεις [6]. Για τον λόγο αυτό πραγματοποιήθηκε αφαίρεση των διπλότυπων εγγραφών στο βασικό στάδιο καθαρισμού, με στόχο τη δημιουργία ενός πιο αξιόπιστου και αντιπροσωπευτικού συνόλου εκπαίδευσης.

Επίσης, υλοποιήθηκε συγκεκριμένη τεχνική για τον καθαρισμό των ελλιπουσών και προβληματικών τιμών. Η σχετική βιβλιογραφία για IDS αναφέρει ότι η παρουσία ελλιπουσών, ασυνεπών ή ακραίων τιμών στα δεδομένα, όπως τιμές NaN, καθώς και άπειρες ή εξαιρετικά μεγάλες τιμές, αποτελεί συχνό και σημαντικό πρόβλημα στα δημόσια σύνολα δεδομένων που χρησιμοποιούνται για ανίχνευση εισβολών, όπως τα UNSW-NB15 και CIC-IDS2017 [5]. Η εφαρμογή κατάλληλων τεχνικών καθαρισμού δεδομένων, όπως η μέθοδος της μεμονωμένης αναπλήρωσης (single imputation) με τη χρήση μέσης τιμής ή η διαγραφή σειρών με ελλιπή δεδομένα (listwise deletion), συμβάλλει ουσιαστικά στην ενίσχυση της ποιότητας των δεδομένων [5]. Στην εφαρμογή και συγκεκριμένα στην τεχνική επιλογής των χαρακτηριστικών που θα δούμε στην συνέχεια υλοποιήθηκε επιπλέον η τεχνική περιορισμού τιμών (Clip), κατά την οποία οι τιμές NaN και Inf αντικαθίστανται με μηδέν [34], ενώ οι ακραίες τιμές περιορίζονται σε προκαθορισμένο εύρος. Αυτή η προσέγγιση είναι γνωστή ως Winsorization και αποτελεί καθιερωμένη τεχνική σε συστήματα IDS που αξιοποιούν αλγορίθμους όπως XGBoost και τυχαίου δάσους [35]. Ως αποτέλεσμα, βελτιώνεται η σταθερότητα της διαδικασίας εκπαίδευσης των μοντέλων και αυξάνεται η αξιοπιστία των προβλέψεών τους, καθώς η ελαχιστοποίηση των θορυβωδών ή ελλιπουσών δεδομένων επιτρέπει στα μοντέλα να μαθαίνουν από πιο καθαρές και αντιπροσωπευτικές πληροφορίες [5].

Η επόμενη μέθοδος που υιοθετήθηκε αφορά την κωδικοποίηση των κατηγορικών μεταβλητών. Τα περισσότερα μοντέλα μηχανικής μάθησης απαιτούν αριθμητικά δεδομένα ως είσοδο και δεν μπορούν να επεξεργαστούν απευθείας κατηγορικές τιμές ή τιμές που περιέχουν σύμβολα [5], [6]. Για τον λόγο αυτό, στην προεπεξεργασία εφαρμόζεται κωδικοποίηση ετικετών (Label Encoding), ώστε οι κατηγορικές μεταβλητές, όπως οι στήλες που αποτυπώνουν κατηγορίες επιθέσεων ή ετικέτες κλάσεων, να μετατραπούν σε αριθμητική μορφή, που είναι συμβατή με αλγορίθμους όπως αυτοί των δέντρων αποφάσεων, τυχαίου δάσους και XGBoost [36]. Συνήθως εφαρμόζεται στην στήλη Label ή Attack των συνόλων δεδομένων η οποία περιλαμβάνει τον τύπο επιθέσεων. Η συγκεκριμένη επιλογή προτιμάται έναντι της one-hot encoding όταν επιδιώκεται αποφυγή της δημιουργίας νέων διαστάσεων, καθώς το label encoding δεν δημιουργεί νέες στήλες και συνεπώς συμβάλλει στη μείωση του υπολογιστικού κόστους [5], [6]. Παράλληλα, σχετικές μελέτες που συνδέουν τη μηχανική μάθηση με XAI τεχνικές, όπως SHAP και LIME,

δείχνουν ότι η σαφέστερη αναπαράσταση των χαρακτηριστικών διευκολύνει και την ερμηνεία της συμπεριφοράς του μοντέλου [5], [13], [36].

Ιδιαίτερα σημαντική είναι επίσης η επιλογή των χαρακτηριστικών, καθώς η ύπαρξη μεγάλου αριθμού μεταβλητών δεν συνεπάγεται απαραίτητα καλύτερη απόδοση. Η μείωση της διαστατικότητας των δεδομένων αποτελεί κρίσιμο βήμα που συμβάλλει τόσο στην αποδοτικότερη εκπαίδευση των μοντέλων όσο και στη βελτίωση της ακρίβειάς τους [5]. Μέθοδοι επιλογής χαρακτηριστικών, όπως οι αλγόριθμοι βελτιστοποίησης σωματιδίων (PSO) και η επιλογή βασισμένη στη συσχέτιση, έχουν αποδειχθεί ικανές να αυξήσουν τον ρυθμό αληθών θετικών και τη συνολική απόδοση των συστημάτων ανίχνευσης εισβολών [5]. Η επιλεκτική διατήρηση των πιο σημαντικών γνωρισμάτων μειώνει το υπολογιστικό κόστος και αποτρέπει το φαινόμενο της υπερ εκπαίδευσης, προάγοντας ταυτόχρονα την ευκολότερη ερμηνευσιμότητα των αποτελεσμάτων, στοιχείο ιδιαίτερα σημαντικό ειδικά σε περιβάλλοντα με περιορισμένους πόρους, όπως οι συσκευές Διαδικτύου των Πραγμάτων [5], [22]. Η επιλογή των κορυφαίων 20 έως 25 χαρακτηριστικών, που βασίζεται σε τεχνικές όπως η αξιολόγηση με τυχαία δάση [37], σε σύνολο μεθόδων όπως ο Extra Tree Classifier, καθώς και στην επιλογή σημαντικότητας βάσει κέρδους (gain-based importance) του XGBoost, επιτυγχάνει μια ισορροπία μεταξύ της μέγιστης δυνατής ακρίβειας και της αποδοτικότητας των υπολογισμών [13], [22]. Παράλληλα, αυτή η προσέγγιση ενισχύει τη δυνατότητα επεξηγηματικής ανάλυσης μέσω προηγμένων μεθόδων XAI, επιτρέποντας την καλύτερη κατανόηση των παραγόντων που επηρεάζουν τις αποφάσεις του μοντέλου [13], [22].

Η ανάγκη για προσεκτική προεπεξεργασία ενισχύεται και από την ευρύτερη βιβλιογραφία της μηχανικής μάθησης, πέρα από το πεδίο των IDS. Μελέτες σε σύνολα δεδομένων που δεν είναι ισορροπημένα δείχνουν πως όταν η κατανομή των κλάσεων είναι άνιση τα μοντέλα τείνουν να παρουσιάζουν μεροληψία προς την πλειοψηφική κλάση, γεγονός που μειώνει την αξιοπιστία τους και αναδεικνύει τη σημασία παρεμβάσεων στο επίπεδο των δεδομένων πριν από την εκπαίδευση [28]. Επίσης έρευνες στην XAI υπογραμμίζουν ότι η εμπιστοσύνη στα black-box μοντέλα δεν εξαρτάται μόνο από την υψηλή επίδοση, αλλά και από το κατά πόσο οι αποφάσεις τους μπορούν να εξηγηθούν με σαφήνεια μέσω τεχνικών όπως SHAP, LIME, Dalex και άλλων προσεγγίσεων [13], [30], [38]. Έχει φανεί ότι η πολυπλοκότητα του συνόλου δεδομένων, όπως η διαστατικότητα, η εντροπία κλάσεων, ο αριθμός γνωρισμάτων και η ισορροπία των κλάσεων, επηρεάζει όχι μόνο τη

γενικευσιμότητα των μοντέλων αλλά και τη μορφή των αποτελεσμάτων ερμηνευσιμότητας που προκύπτουν από διαφορετικές τεχνικές επεξήγησης [38]. Ακόμη και στη βιβλιογραφία αξιολόγησης ταξινομητών τονίζεται ότι η επίδοση ενός classifier δεν πρέπει να ερμηνεύεται μονοδιάστατα, καθώς μπορεί να επηρεάζεται από παραμέτρους της διαδικασίας που δεν αποτυπώνονται πλήρως στις κλασικές μετρικές [37]. Όλα αυτά ενισχύουν τη θέση ότι η σωστή προετοιμασία των δεδομένων αποτελεί αναγκαία προϋπόθεση όχι μόνο για καλύτερη ακρίβεια, αλλά και για πιο αξιόπιστη, συγκρίσιμη και ερμηνεύσιμη ανάλυση [13], [28], [30], [38].

Οι επιλεγμένες μέθοδοι (όπως παρουσιάζει ο Πίνακας 3) υιοθετήθηκαν, καθώς διασφαλίζουν καθαρά, πλήρη και ισορροπημένα δεδομένα για την εκπαίδευση των μοντέλων, ελαχιστοποιούν το υπολογιστικό κόστος μέσω της κατάλληλης κωδικοποίησης και της μείωσης των χαρακτηριστικών, βελτιώνουν την ακρίβεια και τη συνολική αποδοτικότητα των συστημάτων ανίχνευσης εισβολών και παράλληλα, ενισχύουν τη διατηρήσιμη ερμηνευσιμότητα μέσω τεχνικών XAI, διευκολύνοντας την ανθρώπινη κατανόηση των αποφάσεων του μοντέλου.

Στην εφαρμογή μας (αρχική οθόνη της οποίας παρουσιάζεται στην Εικόνα 3), μετά την διαγραφή των διπλότυπων, για τον καθαρισμό NaN τιμών επιλέγουμε την κατάλληλη τεχνική με βάση το ποσοστό των γραμμών που περιέχουν κενά.

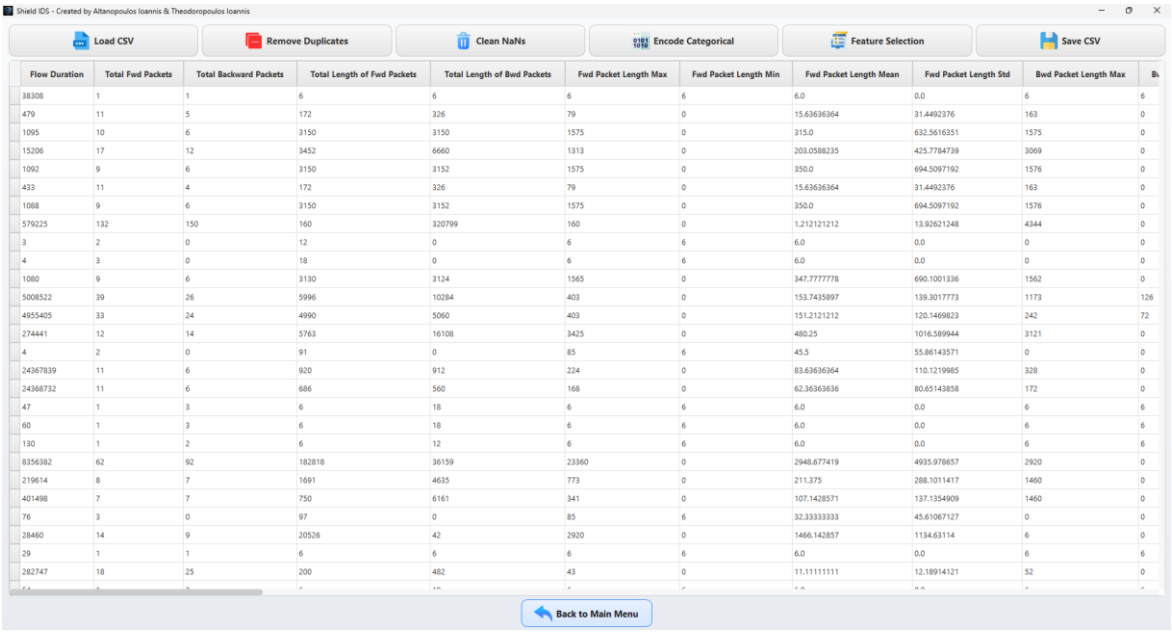
Μέθοδος	**Στόχος**	**Πλεονεκτήματα**
Remove duplicates	Αφαίρεση διπλότυπων εγγραφών από το σύνολο δεδομένων.	Βελτίωση της αξιοπιστίας και της συνέπειας των δεδομένων, αποτροπή παραπλανητικών στατιστικών αποτελεσμάτων και δημιουργία πιο αντιπροσωπευτικού συνόλου εκπαίδευσης.
Data cleaning	Διόρθωση ελλιπουσών, ασυνεπών ή ακραίων δεδομένων.	Αύξηση ποιότητας και αξιοπιστίας δεδομένων, σταθερότητα εκπαίδευσης.
Encode categorical	Μετατροπή κατηγορικών σε αριθμητικά δεδομένα.	Μείωση υπολογιστικού κόστους, συμβατότητα με μοντέλα, βελτιωμένη επεξηγηματικότητα μέσω XAI τεχνικών.
Feature selection (Top 20)	Μείωση διαστατικότητας, απομάκρυνση άσχετων χαρακτηριστικών.	Βελτίωση ακρίβειας και ταχύτητας, αποφυγή overfitting, ερμηνευσιμότητα, μείωση υπολογιστικού κόστους.

Πίνακας 3 Συνοπτικός πίνακας μεθόδων προεπεξεργασίας.

Συγκεκριμένα, αν το ποσοστό των γραμμών με τιμές NaN είναι μικρότερο από 5%, προτείνεται η διαγραφή τους, αφού η απώλεια δεδομένων θεωρείται αμελητέα και δεν αναμένεται να επηρεάσει ουσιαστικά την εκπαίδευση του μοντέλου. Η επιλογή αυτή ευθυγραμμίζεται με πρακτικές προεπεξεργασίας που εφαρμόζονται σε σύγχρονες μελέτες ανίχνευσης εισβολών, όπου γραμμές με null ή infinity values αφαιρούνται όταν αποτελούν μικρό μέρος του συνόλου δεδομένων [39]. Αντίθετα, όταν τα κενά είναι πολλά, προτείνεται η αντικατάσταση των ελλειπουσών τιμών με τον μέσο όρο της αντίστοιχης κλάσης, ώστε να διατηρηθεί η πληροφορία χωρίς να χαθεί σημαντικό μέρος του συνόλου δεδομένων. Σε περίπτωση που δεν εντοπιστεί στήλη ετικέτας (όπως Label, class, attack κτλ.), η προτεινόμενη μέθοδος είναι και πάλι η διαγραφή, καθώς δεν μπορεί να εφαρμοστεί αντικατάσταση τιμών ανά κλάση.

Για τη διαδικασία επιλογής χαρακτηριστικών (Feature Selection), το σύστημα αρχικά πραγματοποιεί επανέλεγχο, για την ύπαρξη προβληματικών γραμμών π.χ. τιμές NaN / Inf ή αριθμοί εκτός λογικού εύρους. Εφόσον εντοπιστούν, εμφανίζεται διάλογος που δίνει τη δυνατότητα στον χρήστη να επιλέξει μέθοδο καθαρισμού, όπως Imputation (αντικατάσταση ελλειπουσών τιμών με τον μέσο όρο της στήλης), Drop (διαγραφή γραμμών με προβληματικές τιμές) ή Clip (περιορισμός των τιμών σε προκαθορισμένο εύρος και αντικατάσταση NaN/Inf με μηδενικά).

Αντίθετα, αν το σύνολο δεδομένων δεν περιέχει προβληματικά δεδομένα, το στάδιο καθαρισμού παραλείπεται αυτόματα και η διαδικασία συνεχίζει απευθείας με την εκπαίδευση του μοντέλου XGBoost, από το οποίο προκύπτουν οι βαθμολογίες σημαντικότητας βασισμένες στο gain score και παρουσιάζονται τα 20 σημαντικότερα χαρακτηριστικά.



Εικόνα 3 Επιλογή συνόλου δεδομένων για προεπεξεργασία.

4.1. Φόρτωση του αρχικού συνόλου δεδομένων

Για να ξεκινήσει η διαδικασία της προεπεξεργασίας ο χρήστης θα πρέπει να επιλέξει το Preprocessing Dataset (όπως φαίνεται στην Εικόνα 4) από το κεντρικό μενού. Πατώντας την επιλογή Load CSV, ενεργοποιείται η συνάρτηση loadCSV(), που είναι υπεύθυνη για τη διαχείριση φόρτωσης (αποσπάσματα της οποίας ακολουθούν παρακάτω), επιλέγεται το αρχείο τύπου CSV μέσω του αντίστοιχου παραθύρου επιλογής αρχείου και το σύνολο δεδομένων εισάγεται στην εφαρμογή για επεξεργασία. Η εφαρμογή έχει σχεδιαστεί με τέτοιο τρόπο ώστε να πραγματοποιείται η ανάγνωση ακόμη και μεγάλων αρχείων, έτσι ώστε παράλληλα να μειώνεται η πιθανότητα σφαλμάτων κατά την αρχική εισαγωγή των δεδομένων.

Όπως φαίνεται και στο σχετικό απόσπασμα κώδικα που ακολουθεί, μόλις ολοκληρωθεί η φόρτωση γίνεται ένας αρχικός έλεγχος και μια βασική προετοιμασία της δομής του συνόλου δεδομένων, βάση των οποίων πραγματοποιείται καθαρισμός των ονομάτων των στηλών από πιθανά περιττά κενά, ώστε να υπάρχει ομοιομορφία και σωστή αναγνώριση των χαρακτηριστικών για τα επόμενα στάδια της επεξεργασίας.

```
# Φόρτωση του CSV αρχείου στη μεταβλητή dataframe (pandas)
self.df = pd.read_csv(
      fileName, low_memory=False
)
```

```
# Καθαρισμός πιθανών κενών στα ονόματα των στηλών
self.df.columns = (
      self.df.columns.str.strip()
)
```

Στη συνέχεια η εφαρμογή προχωρά σε έναν αρχικό έλεγχο για τον εντοπισμό στηλών που ενδέχεται να περιέχουν τιμές σχετικές με διευθύνσεις IP ή αριθμούς θυρών (ports). Σε αυτό το σημείο, όπως υλοποιείται και στο σχετικό απόσπασμα κώδικα, πραγματοποιείται διπλός έλεγχος ο οποίος βασίζεται τόσο στην ονομασία των στηλών όσο και στη μορφή των τιμών που περιέχουν.

```
ip_like_cols = []  # Λίστα IP / port
      #Για κάθε στήλη του dataframe
      for col in self.df.columns:
            # Αν η στήλη ξεκινά με ct_ (counters)
            if col.lower().startswith("ct_"):
                  continue # Παράλειψη
            # 50 πρώτες τιμές της στήλης ως string για ανάλυση
            sample_values = self.df[col].astype(str).head(50)
            # Έλεγχος αν κάποια τιμή μοιάζει με IPv4 (xxx.xxx.xxx.xxx)
            ip_pattern = sample_values.str.match(
                  r"^(\d{1,3}\.){3}\d{1,3}$").any()
            # Έλεγχος για πιθανή port στήλη
            port_pattern = (
                    sample_values.str.match(
                        r"^\d{1,5}$").any() # 1 έως 5 χαρακτήρες
                        # περισσότερες από 5 μοναδικές τιμές
                        and self.df[col].nunique() > 5
            )
            name_hint = any( # Αν εμπεριέχονται λέξεις κλειδιά
                  x in col.lower() for x in ["ip", "addr",
                        "address", "port"]
            )
            # Αν η στήλη μοιάζει με IP ή Port
            if ip_pattern or (port_pattern and name_hint):
                  if ip_pattern: # Αν είναι IP - Περιγραφή στήλης
                        col_type = "Στήλη IP (π.χ. 192.168.x.x)"
                        # Αν είναι port - Περιγραφή στήλης
                        elif port_pattern:
```

```
                        col_type = "Στήλη port (
                              αριθμητικές τιμές 0-65535)"
                  else: # Απροσδιόριστη στήλη
                        col_type = "Απροσδιόριστη στήλη
                              με τιμές IP ή port"
                  # Προσθήκη στήλης και τύπου στήλης στη λίστα
                  ip_like_cols.append((col, col_type))
```

Εάν εντοπιστούν τέτοιες στήλες, ο χρήστης ενημερώνεται μέσω παραθύρου ειδοποίησης και του παρέχεται η δυνατότητα να επιλέξει αν επιθυμεί την αφαίρεσή τους (Εικόνα 5). Μετά την ολοκλήρωση της διαδικασίας, η εφαρμογή εμφανίζει μήνυμα επιβεβαίωσης (Εικόνα 6), στο οποίο αναφέρονται οι στήλες που αφαιρέθηκαν. Αυτή η λειτουργία προστέθηκε ώστε να παρέχεται επιπλέον έλεγχος κατά τη διαδικασία προεπεξεργασίας και να μειώνεται η πιθανότητα σφαλμάτων μετά την αφαίρεση στηλών που δεν κρίνονται σημαντικές και υλοποιείται ως εξής:

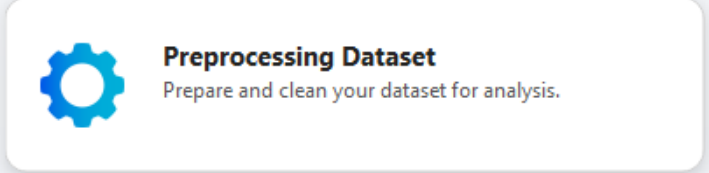
```
if ip_like_cols:  # Αν βρέθηκαν στήλες IP / port
      dialog = QDialog( # Διάλογος επιβεβαίωσης
            self
      ) # Τίτλος παραθύρου
      dialog.setWindowTitle("Ανίχνευση Στηλών IP/Port")
      layout = QVBoxLayout(dialog) # κάθετη διάταξη

      info = QLabel(  # Ετικέτα με οδηγίες
            "Βρέθηκαν οι εξής στήλες που μοιάζουν με IP ή Port.\n"
            "Δες την περιγραφή και επίλεξε ποιες θέλεις να διαγραφούν:"
      )
      # … Γραμμές Κώδικα

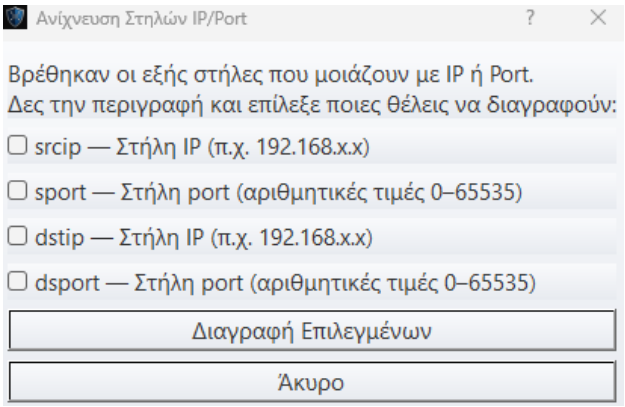
if ( # Αν ο χρήστης πατήσει OK
      dialog.exec_() == QDialog.Accepted
):
      # Συλλέγονται οι στήλες που επιλέχθηκαν για διαγραφή
      cols_to_remove = [
            col for cb, col in checkboxes if cb.isChecked()
      ]
      # Αν υπάρχουν στήλες για διαγραφή
      if cols_to_remove:
            self.df.drop( # Διαγραφή στηλών απο το dataframe
```

```
                columns=cols_to_remove, inplace=True
            )
```

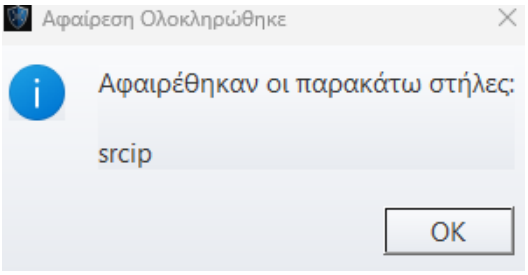
Αφού ολοκληρωθούν τα παραπάνω βήματα, το σύνολο δεδομένων προβάλλεται στον πίνακα της εφαρμογής, δίνοντας στον χρήστη μια άμεση εικόνα του περιεχομένου των δεδομένων που πρόκειται να επεξεργαστεί. Επίσης η εφαρμογή καταγράφει τη διαδικασία φόρτωσης σε αρχείο καταγραφής, ώστε να υπάρχει ιστορικό των ενεργειών και διευκόλυνση στον έλεγχο πιθανών σφαλμάτων.



Εικόνα 4 Επιλογή της ενότητας "Preprocessing Dataset" από το κεντρικό μενού για την έναρξη της διαδικασίας προεπεξεργασίαςς.



Εικόνα 5 Παράθυρο εντοπισμού στηλών που αντιστοιχούν σε διευθύνσεις IP ή αριθμούς θυρών (ports), με δυνατότητα επιλογής των στηλών που θα διαγραφούν πριν από την περαιτέρω προεπεξεργασίας.



Εικόνα 6 Μήνυμα επιβεβαίωσης της επιτυχούς αφαίρεσης στηλών.

4.2. Διαχείριση διπλότυπων εγγραφών

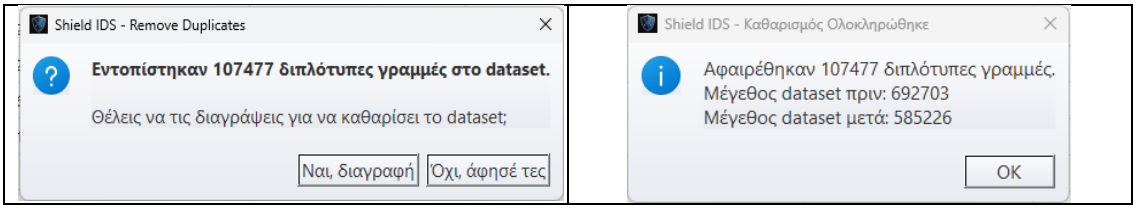
Σε αυτό το στάδιο η εφαρμογή ελέγχει αν το φορτωμένο σύνολο δεδομένων περιέχει επαναλαμβανόμενες γραμμές και δίνει στον χρήστη τη δυνατότητα να προχωρήσει στην αφαίρεσή τους. Ο έλεγχος πραγματοποιείται μετά τη φόρτωση του συνόλου δεδομένων και πριν από τα επόμενα βήματα προεπεξεργασίας, ώστε το σύνολο δεδομένων να καθαριστεί από πλεονάζουσες εγγραφές που δεν προσφέρουν νέα πληροφορία. Εφόσον εντοπιστούν διπλότυπες γραμμές, εμφανίζεται σχετικό παράθυρο επιβεβαίωσης, μέσω του οποίου ο χρήστης ενημερώνεται για το πλήθος τους και επιλέγει αν επιθυμεί τη διαγραφή τους. Μετά την ολοκλήρωση της διαδικασίας, η εφαρμογή εμφανίζει ενημερωτικό μήνυμα με το πλήθος των εγγραφών που αφαιρέθηκαν, καθώς και με το μέγεθος του συνόλου δεδομένων πριν και μετά τον καθαρισμό (όπως φαίνεται και στην Εικόνα 7, Εικόνα 8, Εικόνα 9 και Εικόνα 10). Σε περίπτωση που δεν εντοπιστούν διπλότυπες εγγραφές, η εφαρμογή ενημερώνει τον χρήστη ότι δεν απαιτείται κάποια ενέργεια και η διαδικασία συνεχίζεται κανονικά. Η διαδικασία ελέγχου και αφαίρεσης υλοποιείται από το παρακάτω απόσπασμα της σχετικής συνάρτησης:

```
def removeDuplicates(self):
      # Αν υπάρχει φορτωμένο dataset
      if not hasattr(self, "df"):
            # Ειδοποίηση για μη φορτωμένο dataset
            self.notify("No dataset loaded.", error=True)
                  return # Τερματισμός
            # Σύνολο γραμμών πριν την αφαίρεση
            before = len(self.df)
            # Σύνολο διπλότυπων
            duplicate_count = self.df.duplicated().sum()
            # Διαγραφή διπλότυπων απο το DataFrame
            self.df = self.df.drop_duplicates()
            after = len(self.df) # Σύνολο γραμμών μετά την αφαίρεση
            removed = before - after # Σύνολο γραμμών που αφαιρέθηκαν
```

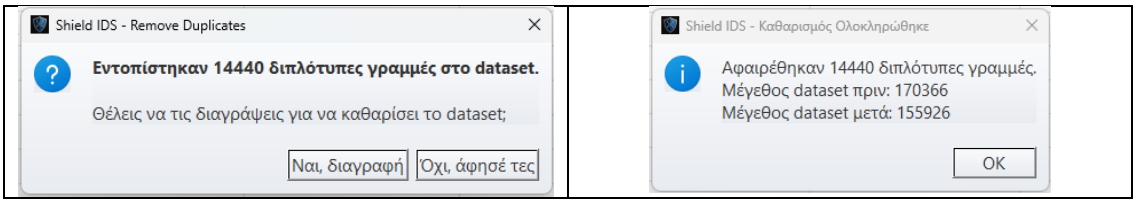
Με βάση την παραπάνω διαδικασία η προεπεξεργασίας εφαρμόστηκε ξεχωριστά σε κάθε υποσύνολο δεδομένων, δηλαδή στα DoS, Port Scan και Web Attack καθώς και στο σύνολο δεδομένων UNSW-NB15. Ο Πίνακας 4 παρουσιάζει τον αριθμό των διπλότυπων εγγραφών που εντοπίστηκαν σε καθένα από τα παραπάνω σύνολα δεδομένων.

Σύνολο / Υπό-σύνολο δεδομένων	Αριθμός διπλότυπων εγγραφών
Wednesday-workingHours- Dos_cleaned	107477
Thursday-WorkingHours-Morning-WebAttacks_cleaned	14440
Friday-WorkingHours-Afternoon-PortScan_cleaned	166889
UNSW-NB15_1	59623

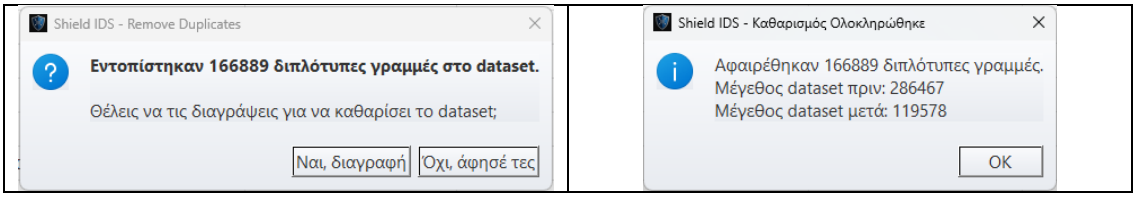
Πίνακας 4 Διπλότυπες εγγραφές ανά σύνολο και υποσύνολο δεδομένων



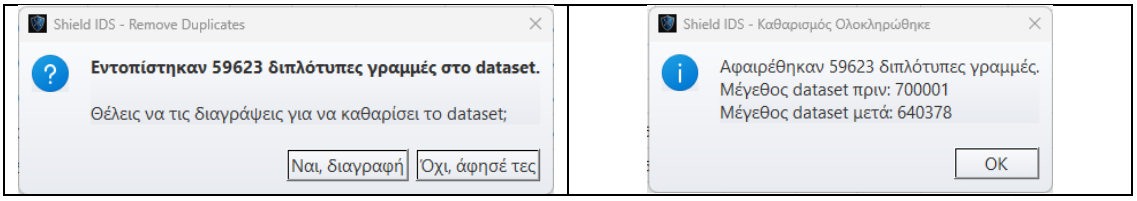
Εικόνα 7 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset Wednesday-workingHours- Dos_cleaned



Εικόνα 8 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned



Εικόνα 9 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned



Εικόνα 10 Επιβεβαίωση εντοπισμού και αφαίρεσης διπλότυπων εγγραφών από το dataset UNSW-NB15_1

4.3. Διαχείριση ελλιπουσών τιμών (NaN)

Μετά την αφαίρεση των διπλότυπων εγγραφών, ο χρήστης επιλέγει το κουμπί Clean NaNs, με το οποίο η εφαρμογή (μέσω της συνάρτησης cleanNaNs() αποσπάσματα της οποίας ακολουθούν) προχωρά στον έλεγχο των ελλιπουσών τιμών του συνόλου δεδομένων. Σε αυτό το στάδιο εξετάζονται οι τιμές NaN αλλά και οι άπειρες τιμές, οι οποίες αρχικά μετατρέπονται σε NaN, ώστε να αντιμετωπίζονται με ενιαίο τρόπο κατά τη διαδικασία καθαρισμού και στη συνέχεια υπολογίζεται ο συνολικός αριθμός των κενών τιμών και ο αριθμός των γραμμών που περιέχουν τουλάχιστον μία ελλιπή τιμή. Εάν δεν εντοπιστούν κενά δεδομένα, η διαδικασία τερματίζεται και ο χρήστης ενημερώνεται ότι δεν απαιτείται περαιτέρω παρέμβαση.

```
df = self.df.copy()  # Αντίγραφο του dataframe για καθαρισμό
# Μετατροπή άπειρων αριθμών σε NaN
df.replace([np.inf, -np.inf], np.nan, inplace=True)
before_rows = len(df)  # Σύνολο γραμμών πριν τον καθαρισμό
total_imputed = 0  # Μετρητής αντικαταστάσεων τιμών NaN
total_removed = 0  # Μετρητής αφαιρούμενων γραμμές

# Υπολογισμός συνόλου NaN τιμών
total_nans = df.isna().sum().sum()
# Υπολογίσμός γραμμών με τουλάχιστον μία NaN τιμή
rows_with_nans = df.isna().any(axis=1).sum()

if total_nans == 0: # Αν δεν υπάρχουν NaN τιμές
      self.notify("Δεν υπάρχουν NaN τιμές στο dataset ") # Ενημέρωση
return # Τερματισμός
```

Η εφαρμογή έχει σχεδιαστεί με τέτοιο τρόπο ώστε να προσαρμόζει την προτεινόμενη ενέργεια ανάλογα με τα χαρακτηριστικά του συνόλου δεδομένων. Συγκεκριμένα, όπως αναφέρθηκε όταν το ποσοστό των γραμμών που περιέχουν NaN είναι μικρότερο από 5% του συνολικού πλήθους γραμμών, προτείνεται η διαγραφή τους καθώς η απώλεια πληροφορίας θεωρείται περιορισμένη. Εάν αυτό το ποσοστό είναι μεγαλύτερο το σύστημα εξετάζει αν υπάρχει διαθέσιμη στήλη ετικέτας όπως για παράδειγμα label, class, target ή attack. Εάν εντοπιστεί τέτοια στήλη προτείνεται η αντικατάσταση των ελλιπουσών τιμών με τον μέσο όρο της αντίστοιχης κλάσης έτσι ώστε να διατηρείται όσο το δυνατόν

περισσότερη πληροφορία από το αρχικό σύνολο δεδομένων. Σε περίπτωση που δεν βρεθεί στήλη ετικέτας, η εφαρμογή προτείνει και πάλι τη διαγραφή των γραμμών με NaN.

```
# Εύρεση στήλης ετικέτας (label/class/target)
class_col = next(
      (
            col # Αναζήτηση σε κάθε στήλη του DataFrame
            for col in df.columns
                  if col.lower() # Αν εμπεριέχονται οι ετικέτες
                        in [
                              "label","class","target","Label","attack",
                              "LABEL","ATTACK","TARGET","Class","Target"
                        ]
)
# Βέλτιστη μέθοδος καθαρισμού ανάλογα με το ποσοστό των NaN γραμμών
if rows_with_nans / before_rows < 0.05: # Αν ποσοστό μικρότερο από 0.05%
      suggestion = "Προτείνεται διαγραφή (οι NaN γραμμές είναι λίγες)."
else: # Αν ποσοστό μεγαλύτερο από 0.05%
      if class_col: # Αν περιέχεται μια από τις ετικέτες
            suggestion = "Προτείνεται αντικατάσταση με
                  μέσο όρο ανά κλάση (πολλές NaN)."
      else: # Αν δεν περιέχεται μια από τις ετικέτες
            suggestion = "Δεν βρέθηκε στήλη label.
                  Προτείνεται διαγραφή."

# QDialog με επιλογές για καθαρισμό (radio buttons)
dialog = QDialog(self) # Δημιουργία παραθύρου διαλόγου
dialog.setWindowTitle("Καθαρισμός NaN") # Τίτλος παραθύρου
layout = QVBoxLayout(dialog) # Προσθήκη

# Κείμενο παραθύρου - Πληροφορίες και μέθοδοι καθαρισμού
info = QtWidgets.QLabel(
      f"Το dataset έχει {rows_with_nans} γραμμές με NaN
            ({total_nans} συνολικά κενά).\n\n"
      f"{suggestion}\n\n"
            "Επέλεξε μέθοδο καθαρισμού:"
)
layout.addWidget(info)
# Μέθοδοι καθαρισμού σε radio buttons
rb1 = QtWidgets.QRadioButton(  # 1η Μέθοδος
```

```
        "Αντικατάσταση με μέσο όρο ανά κλάση (Εφόσον υπάρχει στήλη
            label/class και δεν μπορεί να εφαρμοστεί η Διαγραφή)"
)
rb2 = QtWidgets.QRadioButton( # 2η Μέθοδος
        "Διαγραφή όλων των γραμμών με NaN"
)
```

Η τελική επιλογή της μεθόδου πραγματοποιείται από τον χρήστη μέσω του σχετικού παραθύρου (βλέπε Εικόνα 11). Ο χρήστης ενημερώνεται για τον αριθμό των προβληματικών γραμμών και καλείται να επιλέξει ανάμεσα στις διαθέσιμες μεθόδους καθαρισμού. Αν επιλεγεί η αντικατάσταση τιμών, η διαδικασία εφαρμόζεται μόνο στις αριθμητικές στήλες που περιέχουν NaN. Για κάθε μία από αυτές, υπολογίζεται ο μέσος όρος χωριστά ανά κλάση και οι ελλιπείς τιμές αντικαθίστανται με βάση την κλάση στην οποία ανήκει κάθε εγγραφή. Αν επιλεγεί η διαγραφή, αφαιρούνται όλες οι γραμμές που περιέχουν τουλάχιστον μία τιμή NaN.

```
if (
        rb1.isChecked()
):  # Αν είναι επιλεγμένο το ραδιοκουμπί αντικατάστασης τιμών
        # Ελέγχει αν υπάρχει στήλη ετικέτας για αντικατάσταση
        if not class_col:
                self.notify(
                "Δεν βρέθηκε στήλη label για αντικατάσταση ανά κλάση.",
                error=True,
                )
                return

        # Επιλογή των στηλών των ζητούμενων τύπων δεδομένων
        numeric_cols = df.select_dtypes(
                # Να συμπεριλαμβάνονται μόνο αριθμητικοί τύποι
                include=[np.number]
        ).columns.tolist() # Μετατροπή δείκτη σε λίστα
        cols_with_nans = [  # Λίστα στηλών με τουλάχιστον ένα NaN
                # Διατήρηση στηλών που περιέχουν κενές τιμές
                col for col in numeric_cols if df[col].isna().sum() > 0
        ]
        for col in cols_with_nans:  # Για κάθε στήλη με NaN τιμές
                for cls in ( # Για κάθε κατηγορία ετικέτας
                        # Μοναδικές - μη κενές τιμές
```

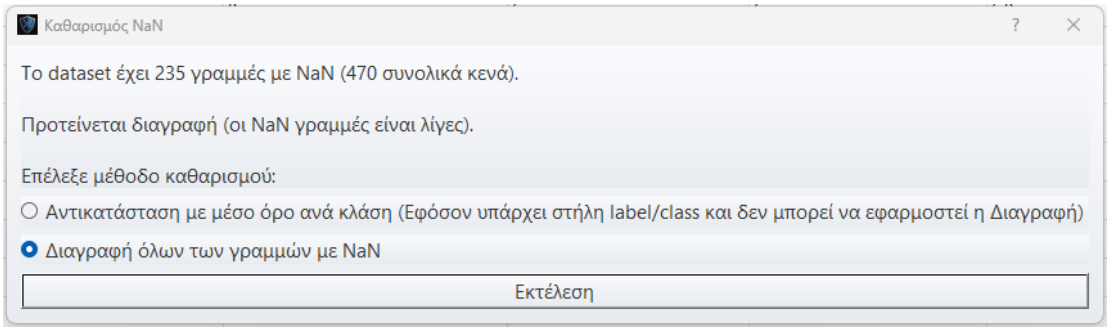
```
                df[class_col].dropna().unique()
            ):
                # Μάσκα γραμμών για τη συγκεκριμένη κλάση
                class_mask = df[class_col] == cls
                mean_val = df.loc[ # ΜΟ στήλης ανά κλάση
                    class_mask, col
                ].mean()  # ΜΟ διαθέσιμων τιμών
                # Αν ο μέσος όρος είναι έγκυρος αριθμός
                if not np.isnan(
                    mean_val # ΜΟ
                ):
                    # Κενά στήλης
                    mask_nan = class_mask & df[col].isna()
                    # Σύνολο τιμών προς συμπλήρωση
                    imputed_count = mask_nan.sum()
                    # Αντικατάσταση NaN με ΜΟ ανά κλάση
                    df.loc[mask_nan, col] = mean_val
                    total_imputed += imputed_count # Αύξηση μετρητή
                else: # Αν ο μέσος όρος δεν είναι έγκυρος αριθμός
                    # Καταγραφή log ότι δεν έγινε αντικατάσταση
                    self.log(
                        f" Δεν έγινε imputation για: κλάση={cls},
                        στήλη={col} (όλα NaN)"
                    )

    elif ( # Αν δεν είναι επιλεγμένο το κουμπί αντικατάστασης τιμών
        rb2.isChecked()
    ):
        # Σύνολο γραμμών που περιέχουν τουλάχιστον ενα NaN
        total_removed = df.isna().any(axis=1).sum()
        # Διαγραφή γραμμών που έχουν έστω μια ελλειπούσα τιμή
        df.dropna(inplace=True)
```

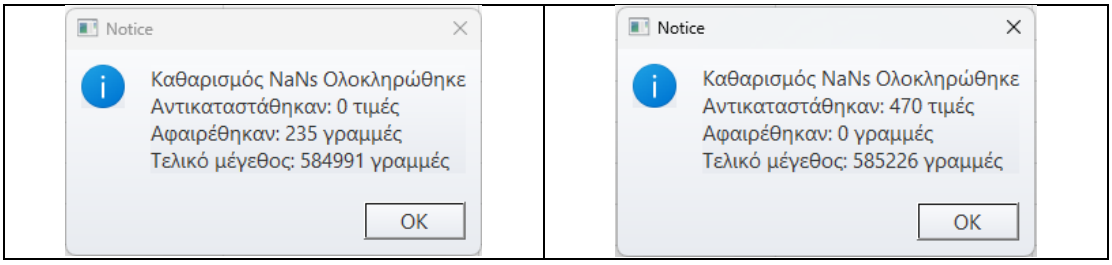
Μετά την ολοκλήρωση της διαδικασίας, το καθαρισμένο σύνολο δεδομένων εμφανίζεται εκ νέου στον πίνακα της εφαρμογής και ο χρήστης ενημερώνεται από την εφαρμογή για το αποτέλεσμα του καθαρισμού. Πιο αναλυτικά εμφανίζεται ο αριθμός των τιμών που αντικαταστάθηκαν, ο αριθμός των γραμμών που αφαιρέθηκαν και το τελικό μέγεθος του συνόλου δεδομένων (Εικόνα 12). Την ίδια στιγμή καταγράφονται οι σχετικές πληροφορίες και στο αρχείο log της εφαρμογής, ώστε να υπάρχει πλήρες ιστορικό.

```
after_rows = len(df)  # Σύνολο γραμμών μετά τον καθαρισμό
self.df = df  # Ενημέρωση dataframe κλάσης
self.showData(self.df)  # Εμφανίζει τα καθαρισμένα δεδομένα στο UI
# Μήνυμα με τα αποτελέσματα καθαρισμού
msg = f"Καθαρισμός NaNs Ολοκληρώθηκε\n"
msg += f"Αντικαταστάθηκαν: {total_imputed} τιμές\n"
msg += f"Αφαιρέθηκαν: {total_removed} γραμμές\n"
msg += f"Τελικό μέγεθος: {after_rows} γραμμές"
self.notify(msg)  # Εμφανίζει μήνυμα στον χρήστη
self.log(msg)  # Καταγράφει τα αποτελέσματα στο log
```

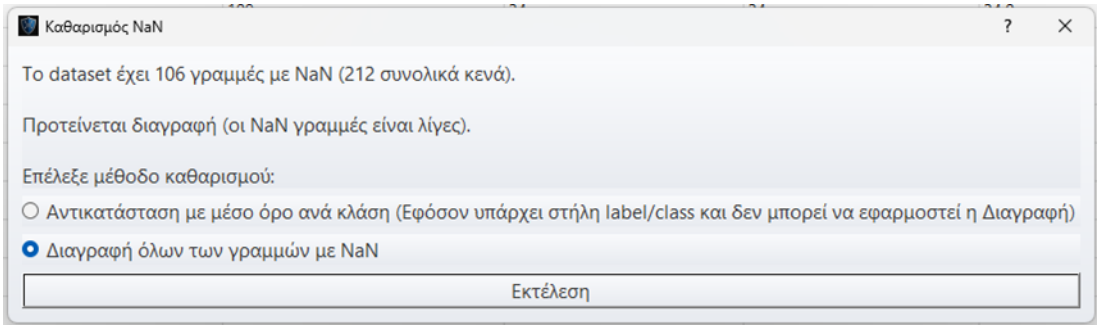
Προκειμένου να αποσαφηνιστούν οι καλύτερες τεχνικές, πραγματοποιήθηκαν όλοι οι δυνατοί συνδυασμοί, έτσι και στην συγκεκριμένη περίπτωση δημιουργήθηκε προεπεξεργασμένο σύνολο δεδομένων στο οποίο έχουν διαγραφεί οι εγγραφές που περιέχουν NaN τιμές αλλά και προ επεξεργασμένο σύνολο δεδομένων στο οποίο έχει πραγματοποιηθεί αντικατάσταση με Μ.Ο. ανά κλάση (Εικόνα 13, Εικόνα 14, Εικόνα 15, Εικόνα 16).



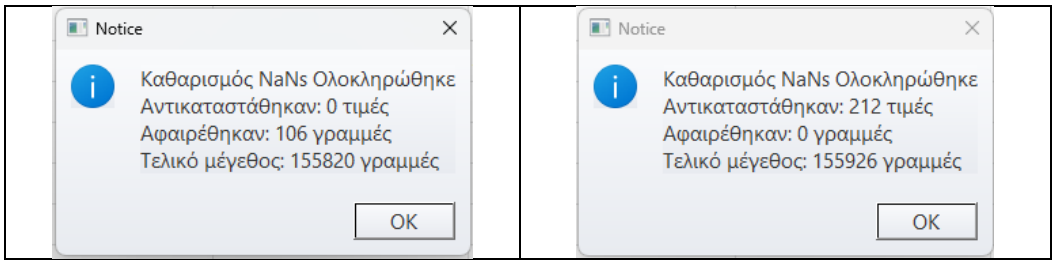
Εικόνα 11 Παράθυρο διαχείρισης NaN τιμών και επιλογής μεθόδου καθαρισμού από το dataset Wednesday-workingHours- Dos_cleaned



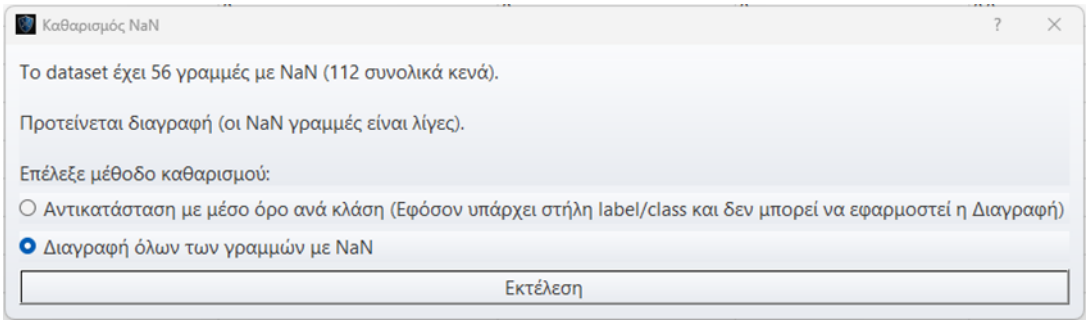
Εικόνα 12 Αποτελέσματα καθαρισμού NaN τιμών με διαγραφή γραμμών και με αντικατάσταση μέσω μέσου όρου ανά κλάση από το dataset Wednesday-workingHours- Dos_cleaned



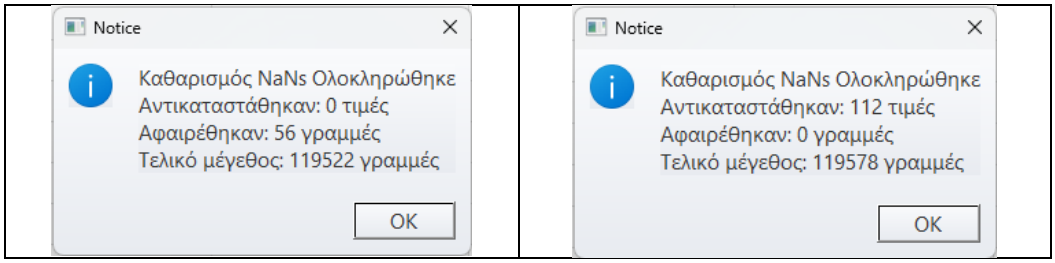
Εικόνα 13 Παράθυρο διαχείρισης NaN τιμών και επιλογής μεθόδου καθαρισμού από το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned



Εικόνα 14 Αποτελέσματα καθαρισμού NaN τιμών με διαγραφή γραμμών και με αντικατάσταση μέσω μέσου όρου ανά κλάση από το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned



Εικόνα 15 Παράθυρο διαχείρισης NaN τιμών και επιλογής μεθόδου καθαρισμού από το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned



Εικόνα 16 Αποτελέσματα καθαρισμού NaN τιμών με διαγραφή γραμμών και με αντικατάσταση μέσω μέσου όρου ανά κλάση από το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned

Για το σύνολο δεδομένων UNSW-NB15_1 δεν εφαρμόστηκε η μέθοδος καθώς το συγκεκριμένο δεν έχει NaN τιμές αλλά ούτε και άπειρες.

4.4. Κωδικοποίηση κατηγορικών μεταβλητών

Σε αυτό το στάδιο ο χρήστης επιλέγοντας το κουμπί Encode Categorical μέσω της συνάρτησης encodeCategorical() αρχικά γίνεται ο εντοπισμός των στηλών που περιέχουν κατηγορικές τιμές και η μετατροπή τους σε αριθμητική μορφή ώστε να μπορούν να αξιοποιηθούν στα επόμενα στάδια επεξεργασίας και εκπαίδευσης.

```
if hasattr(self, "df"): # Αν υπάρχει διαθέσιμο DataFrame
      allowed_cols = [ # Λίστα στηλών - στόχων
            "label", "class", "target", "Label", "attack",
            "LABEL", "ATTACK", "TARGET", "Class", "Target", "attack_cat",
      ]
      cat_cols = [ # Εντοπισμός των κατηγορικών στηλών (τύπου object)
      col for col in self.df.columns if self.df[col].dtype == "object"
      ]
      if not cat_cols: # Αν δεν υπάρχει κατηγορική στήλη
            self.notify("Δεν βρέθηκαν κατηγορικές στήλες για encoding.")
            return # Ενημέρωση χρήστη κ' Τερματισμός
```

Η εφαρμογή εξετάζει τις στήλες του συνόλου δεδομένων και αναγνωρίζει ως κατηγορικές εκείνες που είναι αποθηκευμένες σε μορφή αντικειμένου. Εφόσον εντοπιστούν τέτοιες στήλες, εφαρμόζεται σε καθεμία από αυτές η μέθοδος κωδικοποίησης ετικετών μέσω της οποίας κάθε διακριτή κατηγορία αντιστοιχίζεται σε μία μοναδική ακέραια αριθμητική τιμή (όπως φαίνεται χαρακτηριστικά στην Εικόνα 17, Εικόνα 18, Εικόνα 19, και Εικόνα 20). Έτσι, όπως παρουσιάζεται και στο σχετικό απόσπασμα κώδικα που ακολουθεί, οι αρχικά λεκτικές ή συμβολικές τιμές μετατρέπονται σε μορφή κατάλληλη για αριθμητική επεξεργασία χωρίς να απαιτείται δημιουργία επιπλέον στηλών.

```
for col in cat_cols: # Για κάθε κατηγορική στήλη
      le = LabelEncoder() # Νέο LabelEncoder για τη συγκεκριμένη στήλη
      # Μετατροπή σε αριθμητικές τιμές
      self.df[col] = le.fit_transform(self.df[col].astype(str))
      # Ορισμός ονόματος αρχείου encoder
      encoder_filename = os.path.join(save_dir, f"{col}_encoder.pkl")
      joblib.dump(le, encoder_filename) # Αποθήκευση LabelEncoder
```

Κατά την υλοποίηση της εφαρμογής έγινε ξεχωριστή πρόβλεψη για τις στήλες που σχετίζονται με την ετικέτα κλάσης ή τον τύπο επίθεσης, όπως για παράδειγμα οι στήλες label, class, target, attack. Για αυτές τις στήλες η εφαρμογή δεν περιορίζεται μόνο στην

κωδικοποίηση αλλά παρουσιάζει και την αντιστοίχιση των αρχικών κατηγοριών με τις νέες αριθμητικές τιμές ώστε ο χρήστης να γνωρίζει τη σημασία των νέων κωδικών που προκύπτουν. Αυτό είναι χρήσιμο τόσο για την ερμηνεία των αποτελεσμάτων όσο και για την περαιτέρω ανάλυση του συνόλου δεδομένων.

```
if col in allowed_cols: # Αν είναι στήλη - στόχος
      mapping = { # Λεξικό αντιστοίχισης από αρχική κατηγορία σε αριθμό
            cls: idx for idx, cls in enumerate(le.classes_)
      } # Κείμενο της μορφής "κατηγορία -> αριθμός"
      # Δημιουργεί κείμενο με τη μορφή "κατηγορία -> αριθμός"
      mapping_text = "\n".join(
            [f"{cls} → {idx}" for cls, idx in mapping.items()]
      ) # Τελικό μήνυμα με αντιστοίχιση της στήλης
      msg = f"Στήλη: {col}\n\nMapping:\n{mapping_text}"
```

Για τις στήλες-στόχους η εφαρμογή δημιουργεί εικόνα του πίνακα αντιστοίχισης με τη χρήση του QPixmap, όπου απεικονίζονται οι αρχικές κατηγορίες μαζί με τις αντίστοιχες αριθμητικές τιμές τους. Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει αυτή την αναπαράσταση ως αρχείο εικόνας μορφής PNG με προτεινόμενο όνομα (mapping_{col}.png), μέσω παραθύρου αποθήκευσης που εμφανίζεται αυτόματα.

```
# Παράθυρο Save As για αποθήκευση PNG εικόνας
save_path, _ = QFileDialog.getSaveFileName(
    self, # Κύριο παράθυρο της εφαρμογής
    "Αποθήκευση εικόνας ως πίνακα", # Τίτλος παραθύρου
    f"mapping_{col}.png", # Προτεινόμενο όνομα αρχείου
    "PNG Files (*.png)", # Φίλτρο αρχείων
)
if save_path: # Αν ο χρήστης επέλεξε τοποθεσία (και δεν πάτησε cancel)
    pixmap.save(save_path, "PNG") # Αποθήκευση εικόνας
    self.notify(f"Η εικόνα αποθηκεύτηκε στο: {save_path}")
```

Επίσης κατά την εκτέλεση της διαδικασίας αποθηκεύεται ο αντίστοιχος encoder για κάθε κατηγορική στήλη σε ξεχωριστό αρχείο έτσι ώστε να χρησιμοποιηθεί στην IDS εφαρμογή μας. Με αυτόν τον τρόπο διασφαλίζεται η συνέπεια στη μετατροπή των κατηγορικών τιμών στην εφαρμογή μας. Μετά την ολοκλήρωση της κωδικοποίησης, το ενημερωμένο σύνολο δεδομένων εμφανίζεται εκ νέου στο περιβάλλον της εφαρμογής και καταγράφεται σχετική πληροφορία στο αρχείο καταγραφής.

Label Encoding – Label

Κατηγορία	Αντιστοίχιση
BENIGN	0
DoS GoldenEye	1
DoS Hulk	2
DoS Slowhttptest	3
DoS slowloris	4
Heartbleed	5

Εικόνα 17 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset Wednesday-workingHours- Dos_cleaned

Label Encoding – Label

Κατηγορία	Αντιστοίχιση
BENIGN	0
Web Attack  Brute Force	1
Web Attack  Sql Injection	2
Web Attack  XSS	3

Εικόνα 18 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned

Label Encoding – Label

Κατηγορία	Αντιστοίχιση
BENIGN	0
PortScan	1

Εικόνα 19 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned

Label Encoding – attack_cat

Κατηγορία	Αντιστοίχιση
Fuzzers	0
Analysis	1
Backdoors	2
DoS	3
Exploits	4
Generic	5
Reconnaissance	6
Shellcode	7
Worms	8
normal	9

Εικόνα 20 Πίνακας αντιστοίχισης κατηγοριών και αριθμητικών τιμών για το dataset UNSW-NB15_1

4.5. Επιλογή χαρακτηριστικών (feature selection)

Μετά το στάδιο της κωδικοποίησης των κατηγορικών μεταβλητών και εφόσον ο χρήστης επιλέξει το κουμπί Feature Selection η εφαρμογή, μέσω της συνάρτησης featureSelectionXGBoost(), προχωρά στην επιλογή των σημαντικότερων χαρακτηριστικών του συνόλου δεδομένων. Η διαδικασία εφαρμόζεται αφού πρώτα προηγηθεί έλεγχος της καταλληλότητας του συνόλου δεδομένων, ώστε να διασφαλιστεί ότι η επιλογή των χαρακτηριστικών βασίζεται σε ορθά επεξεργασμένα δεδομένα.

Αρχικά ο χρήστης καλείται να επιλέξει τη στήλη-στόχο (Εικόνα 21, Εικόνα 23, Εικόνα 25, Εικόνα 27) δηλαδή τη μεταβλητή που θα χρησιμοποιηθεί ως ετικέτα για την εκπαίδευση του μοντέλου. Στη συνέχεια, η εφαρμογή διαχωρίζει το σύνολο δεδομένων σε χαρακτηριστικά εισόδου και σε μεταβλητή στόχο και πραγματοποιεί νέο έλεγχο για την ύπαρξη προβληματικών γραμμών. Ο έλεγχος αυτός έχει ενσωματωθεί ως επιπλέον μηχανισμός επαλήθευσης παρότι η αντίστοιχη διαδικασία έχει ήδη εφαρμοστεί στο στάδιο του καθαρισμού NaN. Όπως φαίνεται και στο σχετικό απόσπασμα κώδικα που ακολουθεί, στον έλεγχο αυτό λαμβάνονται υπόψη οι τιμές NaN και οι τιμές Inf. Αν εντοπιστούν τέτοιες περιπτώσεις η εφαρμογή ενημερώνει τον χρήστη και του παρέχει τη δυνατότητα να επιλέξει ανάμεσα σε διαφορετικές μεθόδους αντιμετώπισης, όπως αντικατάσταση των προβληματικών τιμών με τον μέσο όρο της στήλης, διαγραφή των αντίστοιχων γραμμών ή περιορισμό των τιμών σε προκαθορισμένο εύρος.

```
# Παράθυρο διαλόγου επιλογής στήλη – στόχου
target_col, ok = QInputDialog.getItem(
      self, # Γονικό widget
      "Select Target Column", # Τίτλος
      "Choose the target column:", # Ετικέτα
      self.df.columns.tolist(), # Λίστα με ονόματα στηλών
      0, # Προεπιλογή (πρώτη στήλη)
      False, # editable=False (δεν επιτρέπεται χειροκίνητη εισαγωγή)
)
if not ok: # Ακύρωση από χρήστη
      # Μήνυμα ακύρωσης
      self.notify("No target column selected.", error=True)
      return # Τερματισμός

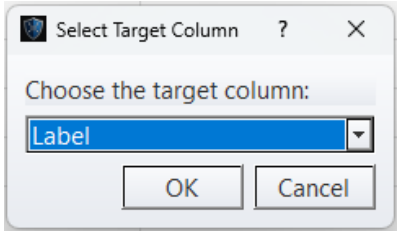
label_col = target_col # Αποθήκευση επιλεγμένης στήλης στόχου
# Διαχωρισμός features / target
# Features: Όλες οι στήλες εκτός από target
X = self.df.drop(label_col, axis=1)
y = self.df[label_col] # Target: Η στήλη ετικέτας

# Προβληματικές γραμμές (NaN / Inf ή τιμές εκτός float64 range)
mask_problem = ~np.isfinite(X).all(axis=1) |
                  (X.abs() > 1e308).any(axis=1)
problematic_rows = mask_problem.sum() # Σύνολο προβληματικών γραμμών
total_rows = len(X) # Σύνολο γραμμών
if (
      problematic_rows > 0 # Αν υπάρχει προβληματική γραμμή
):
      # Ποσοστό προβληματικών γραμμών
      ratio = problematic_rows / total_rows
      if ratio < 0.05: # Αν είναι λιγότερες από το 5%
            suggestion = "Προτείνεται Drop (οι προβληματικές γραμμές
                              είναι λίγες)."
            default = "drop" # Προεπιλογή πρόταση διαγραφής
      else: # Αν είναι περισσότερες από το 5%
            suggestion = "Προτείνεται Imputation (πολλές γραμμές έχουν
                        NaN/inf)."
            default = "impute" # Προεπιλογή πρόταση συμπλήρωσης
```

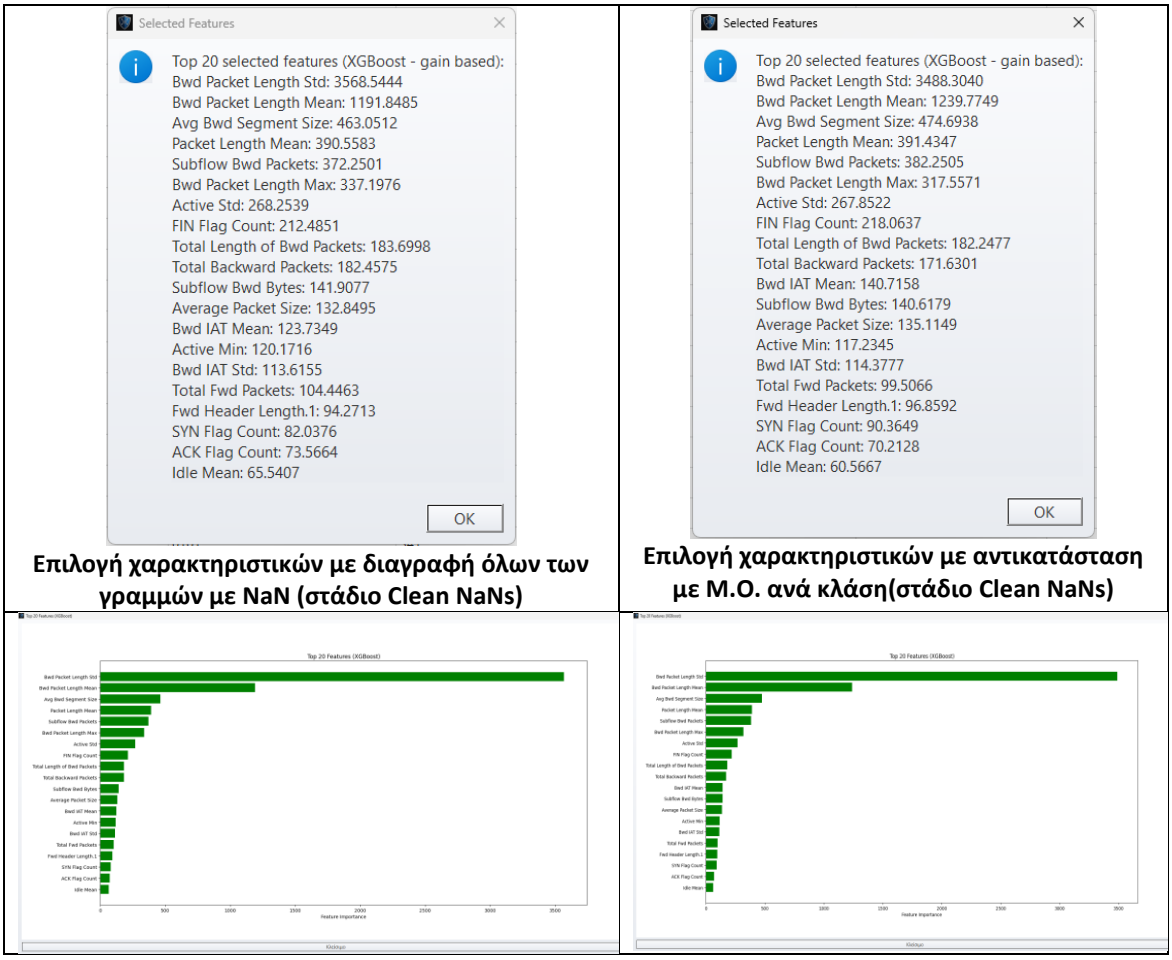
Μετά τον έλεγχο και την κατάλληλη προετοιμασία των δεδομένων, εφαρμόζεται μοντέλο XGBoost για την εκτίμηση της σημαντικότητας των χαρακτηριστικών. Η επιλογή των χαρακτηριστικών βασίζεται στις βαθμολογίες σημαντικότητας που παράγονται από το μοντέλο και ειδικότερα στην προσέγγιση gain-based importance, μέσω της οποίας αποτιμάται η συμβολή κάθε χαρακτηριστικού στη βελτίωση της απόδοσης του μοντέλου. Στη συνέχεια, τα χαρακτηριστικά ταξινομούνται κατά φθίνουσα σειρά σημαντικότητας και διατηρούνται τα 20 σημαντικότερα.

```
# Εκπαίδευση μοντέλου με features (float32, target y)
model.fit(X.astype(np.float32), y)
# Εσωτερικό booster αντικείμενο από εκπαιδευμένο μοντέλο XGBoost
booster = model.get_booster()
# Υπολογισμός σημαντικότητας ανά χαρακτηριστικό με βάση το average gain
importances_dict = booster.get_score(importance_type='gain')
# Μετατροπή σε pandas Series
importances = pd.Series(importances_dict)
# Μετατροπή feature ids (f0, f1...) στα πραγματικά ονόματα στηλών
importances.index = [i if i in X.columns else X.columns[int(i[1:])] for
                     i in importances.index]
# Ταξινόμηση από το σημαντικότερο προς το λιγότερο και επιλογή top 20
features_sorted = importances.sort_values(ascending=False).head(20)
```

Η εφαρμογή παρουσιάζει στον χρήστη τόσο λεκτικά όσο και γραφικά (όπως φαίνεται στην Εικόνα 22, Εικόνα 24, Εικόνα 26 και Εικόνα 28) το αποτέλεσμα της διαδικασίας, εμφανίζοντας τα επιλεγμένα χαρακτηριστικά και τη σχετική τους βαρύτητα. Μετά την ολοκλήρωση της διαδικασίας δημιουργείται νέο επεξεργασμένο σύνολο δεδομένων το οποίο περιλαμβάνει μόνο τα επιλεγμένα χαρακτηριστικά μαζί με τη στήλη-στόχο.



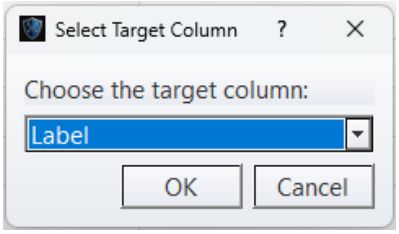
Εικόνα 21 Επιλογή της στήλης στόχου για το dataset Wednesday-workingHours- Dos_cleaned.



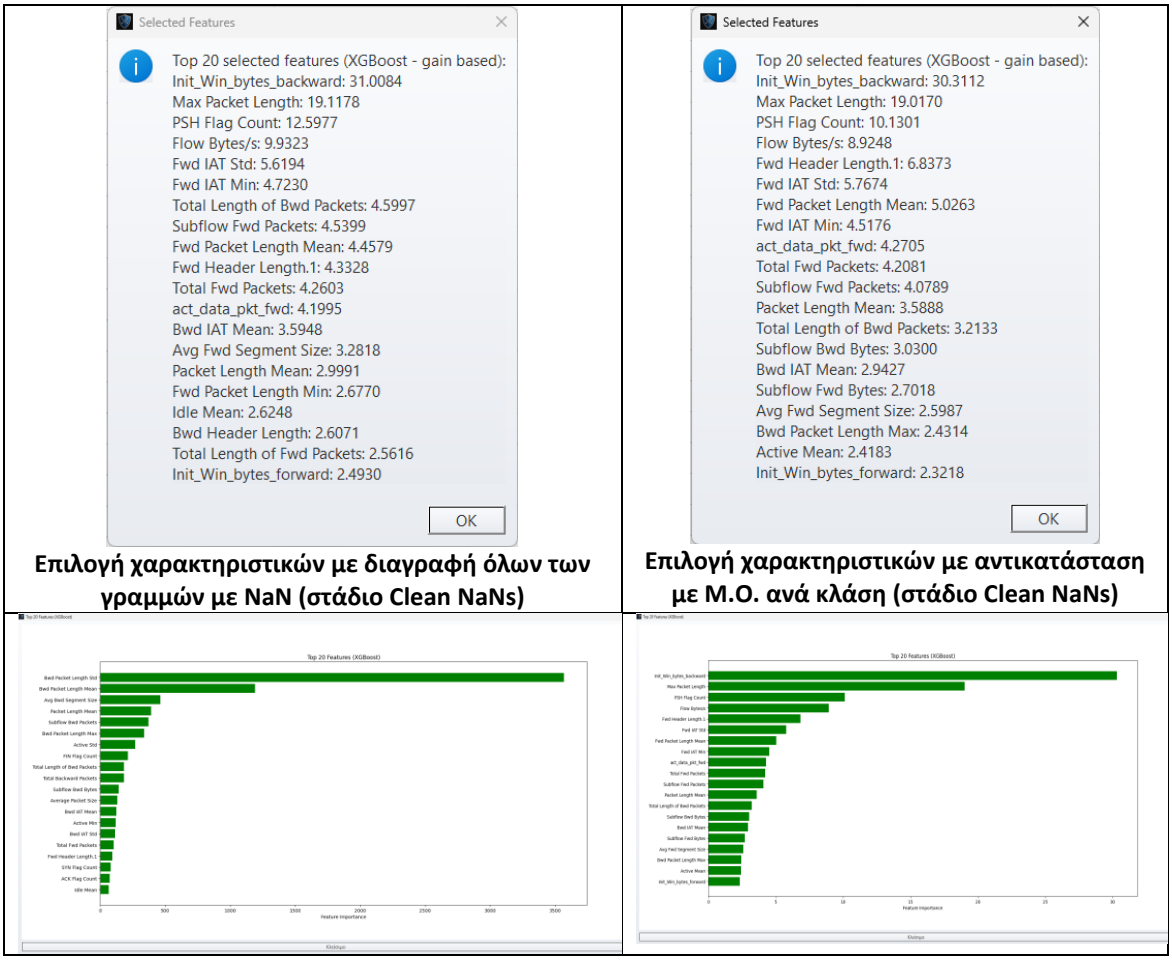
Επιλογή χαρακτηριστικών με διαγραφή όλων των γραμμών με NaN (στάδιο Clean NaNs)

Επιλογή χαρακτηριστικών με αντικατάσταση με Μ.Ο. ανά κλάση(στάδιο Clean NaNs)

Εικόνα 22 Επιλεγμένα χαρακτηριστικά για το dataset Wednesday-workingHours- Dos_cleaned μετά τη διαδικασία feature selection με XGBoost για τις δύο μεθόδους διαχείρισης NaN τιμών



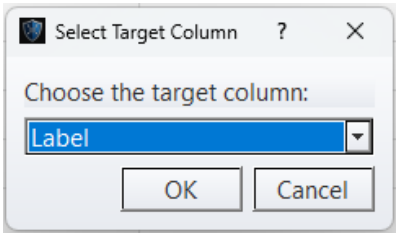
Εικόνα 23 Επιλογή της στήλης στόχου για το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.



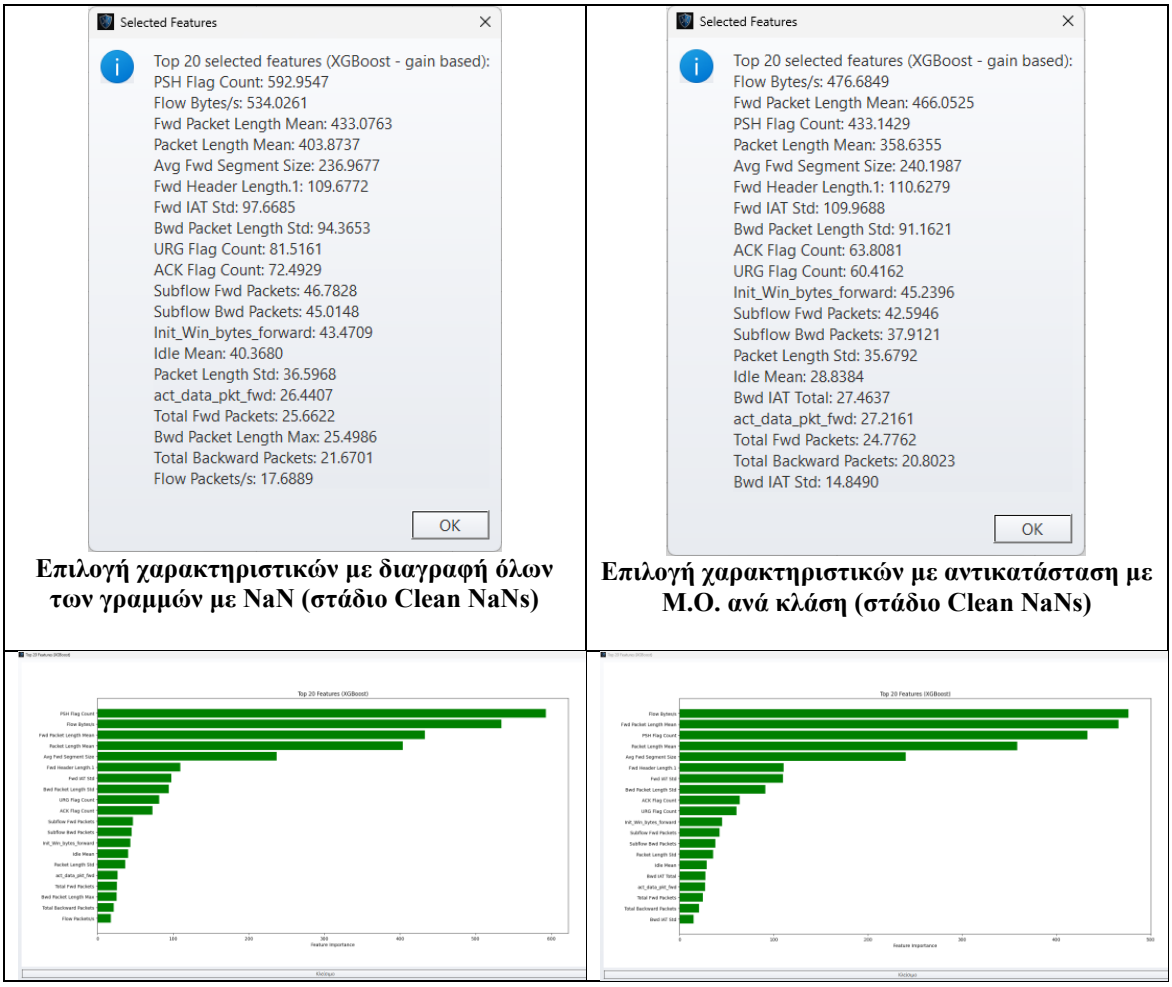
Επιλογή χαρακτηριστικών με διαγραφή όλων των γραμμών με NaN (στάδιο Clean NaNs)

Επιλογή χαρακτηριστικών με αντικατάσταση με Μ.Ο. ανά κλάση (στάδιο Clean NaNs)

Εικόνα 24 Επιλεγμένα χαρακτηριστικά για το dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned μετά τη διαδικασία feature selection με XGBoost για τις δύο μεθόδους διαχείρισης NaN τιμών



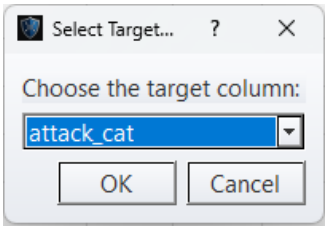
Εικόνα 25 Επιλογή της στήλης στόχου για το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.



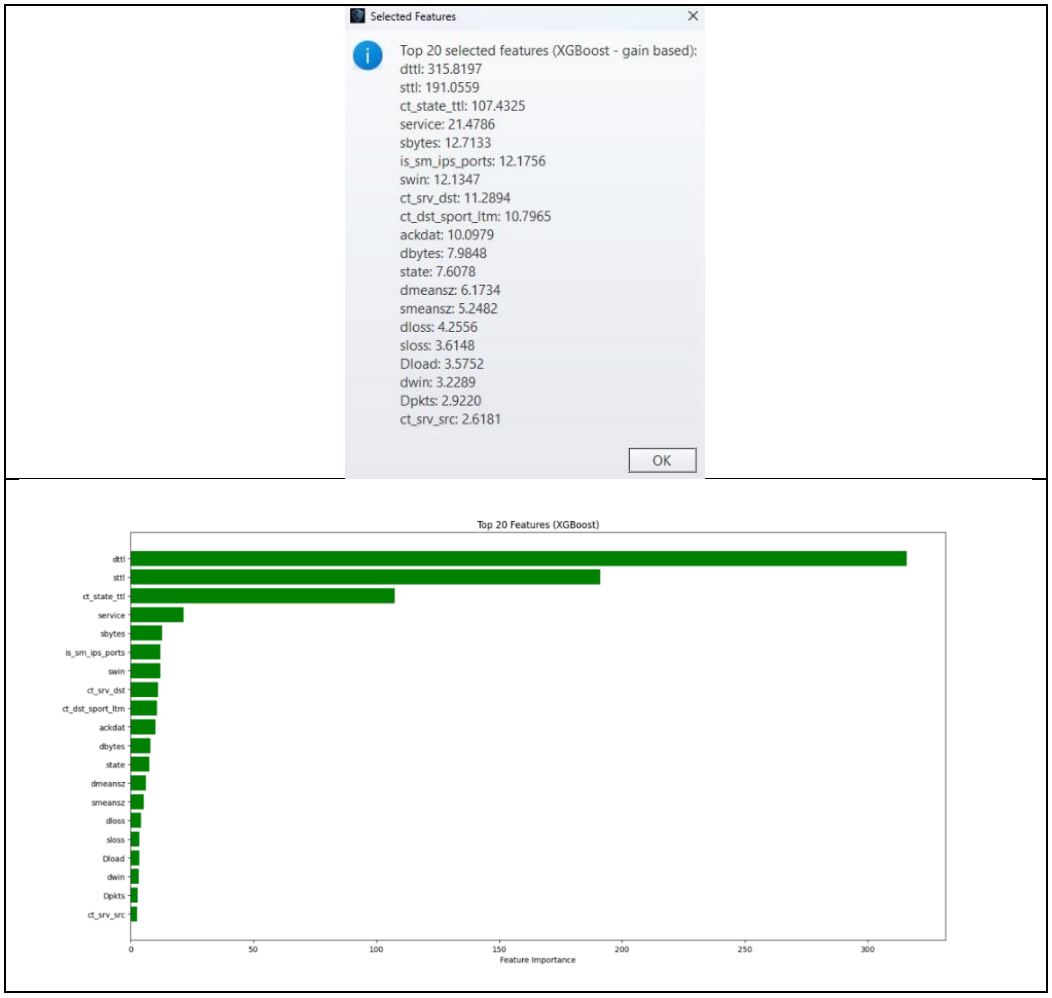
Επιλογή χαρακτηριστικών με διαγραφή όλων των γραμμών με NaN (στάδιο Clean NaNs)

Επιλογή χαρακτηριστικών με αντικατάσταση με Μ.Ο. ανά κλάση (στάδιο Clean NaNs)

Εικόνα 26 Επιλεγμένα χαρακτηριστικά για το dataset Friday-WorkingHours-Afternoon-PortScan_cleaned μετά τη διαδικασία feature selection με XGBoost για τις δύο μεθόδους διαχείρισης NaN τιμών



Εικόνα 27 Επιλογή της στήλης στόχου για το dataset UNSW-NB15_1



Εικόνα 28 Επιλεγμένα χαρακτηριστικά για το dataset UNSW-NB15_1

4.6. Δημιουργία και αποθήκευση τελικού επεξεργασμένου συνόλου δεδομένων

Μετά την ολοκλήρωση όλων των παραπάνω σταδίων και εφόσον ο χρήστης επιλέξει την ενέργεια Save CSV, η εφαρμογή δημιουργεί το τελικό επεξεργασμένο σύνολο δεδομένων. Το αρχείο αυτό αποτελεί το αποτέλεσμα ολόκληρης της διαδικασίας προεπεξεργασίας, καθώς συγκεντρώνει όλες τις μεταβολές που πραγματοποιήθηκαν στα προηγούμενα βήματα. Το τελικό σύνολο δεδομένων περιλαμβάνει μόνο τα χαρακτηριστικά που κρίθηκαν ως σημαντικότερα, μαζί με τη μεταβλητή στόχο γεγονός που το καθιστά κατάλληλο για την χρήση του στο στάδιο της εκπαίδευσης.

Η αποθήκευση του τελικού επεξεργασμένου συνόλου δεδομένων σε αρχείο τύπου CSV είναι ιδιαίτερα σημαντική, καθώς μπορεί να χρησιμοποιηθεί ξανά χωρίς να απαιτείται εκ νέου εκτέλεση όλων των σταδίων προεπεξεργασίας. Κατά την αποθήκευση η εφαρμογή εμφανίζει παράθυρο επιλογής τοποθεσίας και προτείνει αυτόματα την ονομασία αρχείου

preprocessed.csv. Μετά την επιτυχή αποθήκευση, εμφανίζεται ενημερωτικό μήνυμα που επιβεβαιώνει την ολοκλήρωση της διαδικασίας και την τοποθεσία του αρχείου. Η υλοποίηση πραγματοποιείται ως ακολούθως:

```
# Αποθήκευση τρέχοντος DataFrame σε αρχείο CSV
def saveCSV(self):
      if hasattr(self, "df"): # Αν υπάρχει φορτωμένοDataFrame
            # Παράθυρο διαλόγου (Save As)
            fileName, _ = QFileDialog.getSaveFileName(
                  # Τίτλος, προτεινόμενο όνομα, φίλτρο τύπου αρχείου
                  self, "Save CSV", "preprocessed.csv", "CSV Files
                        (*.csv)"
            )
            # Αν ο χρήστης επέλεξε όνομα αρχείου και δεν πάτησε Cancel
            if fileName:
                  # Αποθήκευση DataFrame σε CSV χωρίς να τη στήλη index
                  self.df.to_csv(fileName, index=False)
                  # Ενημέρωση χρήστη
                  self.notify("CSV saved successfully.")
                  # Καταγραφή διαδρομής αποθηκευμένου αρχείου στο log
                  self.log(f"Saved processed CSV to: {fileName}")
```

5. Εκπαίδευση, αξιολόγηση κι επεξηγησιμότητα μοντέλου

Η εκπαίδευση μοντέλων μηχανικής μάθησης για ανίχνευση εισβολών περιλαμβάνει στάδια όπως η επιλογή κατάλληλου αλγορίθμου, η αντιμετώπιση της ανισορροπίας κλάσεων, η βελτιστοποίηση υπέρ παραμέτρων και η αξιολόγηση σε ανεξάρτητο test set [22], [29], [36]. Επίσης είναι σημαντικό ότι ένα μοντέλο υψηλής απόδοσης δεν είναι από μόνο του επαρκές για πρακτική χρήση. Τα black-box μοντέλα, παρά την εξαιρετική τους αποδοτικότητα, δεν παρέχουν στους αναλυτές ασφάλειας την εξήγηση που χρειάζονται για να επαληθεύσουν τα αποτελέσματα και να λάβουν τεκμηριωμένες αποφάσεις [16], γεγονός που καθιστά την ενσωμάτωση μεθόδων επεξηγησιμότητας αναγκαία προϋπόθεση για την πρακτική αξιοποίησή τους [22], [36].

Όπως αναφέρθηκε, για την προεπεξεργασίας των δεδομένων χρησιμοποιήθηκαν τα τρία επιλεγμένα υποσύνολα του CIC-IDS2017 (Wednesday-workingHourscleaned.csv, Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv και Friday-WorkingHours-Afternoon-PortScan_cleaned.csv) καθώς και το αρχείο UNSW-NB15_1.csv, πάνω στα οποία εφαρμόστηκαν συστηματικά δύο διαφορετικά σενάρια καθαρισμού και μετασχηματισμού. Στο πρώτο σενάριο αφαιρέθηκαν διπλότυπες εγγραφές, οι γραμμές με ελλιπείς τιμές (NaN) διαγράφηκαν πλήρως και στη συνέχεια οι κατηγορικές μεταβλητές κωδικοποιήθηκαν σε αριθμητική μορφή, πριν πραγματοποιηθεί η επιλογή των πιο σημαντικών χαρακτηριστικών. Στο δεύτερο σενάριο ακολουθήθηκε η ίδια ροή, με τη διαφορά ότι οι ελλιπείς τιμές δεν διαγράφηκαν αλλά αντικαταστάθηκαν με τον μέσο όρο της αντίστοιχης κλάσης (class-wise mean imputation), ώστε να διατηρηθούν περισσότερα δείγματα χωρίς να χαθεί η πληροφορία της ετικέτας. Εφαρμόζοντας αυτά τα σενάρια και στις τέσσερις αρχικές πηγές δεδομένων, προέκυψαν συνολικά επτά νέα, καθαρισμένα και μετασχηματισμένα αρχεία CSV, τα οποία χρησιμοποιούνται στο επόμενο στάδιο της διαδικασίας, αυτό της εκπαίδευσης και αξιολόγησης των μοντέλων ανίχνευσης εισβολών.

Στην εφαρμογή μας, η εκπαίδευση (αρχική οθόνη της οποίας δείχνει η Εικόνα 29) πραγματοποιείται ως μια ολοκληρωμένη διαδικασία, που ξεκινά από τα δεδομένα που έχουν ήδη προ επεξεργαστεί και καταλήγει σε ένα εκπαιδευμένο και ερμηνεύσιμο μοντέλο, έτοιμο για χρήση στην κύρια IDS εφαρμογή. Αρχικά, φορτώνεται ένα κατάλληλο CSV αρχείο μέσω του κουμπιού “Load CSV” και επιλέγεται από το σχετικό διάλογο ποια στήλη θα χρησιμοποιηθεί ως ετικέτα (label), ώστε η εφαρμογή να διαχωρίσει αυτόματα τις

μεταβλητές εισόδου (features) από την κλάση στόχο. Αν η κλάση είναι κατηγορική, κωδικοποιείται σε αριθμητική μορφή, ώστε να είναι συμβατή με τον ταξινομητή XGBoost.

Η επιλογή του XGBoost ως κεντρικού ταξινομητή της εφαρμογής δεν είναι τυχαία. Στη σχετική βιβλιογραφία, ο αλγόριθμος αυτός αναδεικνύεται επανειλημμένα ως ο πλέον αποδοτικός για σενάρια ανίχνευσης εισβολών. Ο XGBoost αναδείχθηκε ως ένας από τους κορυφαίους ταξινομητές σε πειράματα με τα σύνολα δεδομένων CICIDS2017 και UNSW-NB15, υπερτερώντας στην πλειονότητα των κατηγοριών επίθεσης [5], [36].

Στη συνέχεια, με το κουμπί “Start Training” ξεκινά η διαδικασία εκπαίδευσης σε ξεχωριστό νήμα, ώστε να μην παγώνει το γραφικό περιβάλλον και να παραμένει λειτουργικό κατά τη διάρκεια των υπολογισμών. Το σύνολο δεδομένων χωρίζεται αυτόματα σε δύο σύνολα train και test σε αναλογία 70%–30% με διατήρηση της αναλογίας των κλάσεων επιλογή που είναι ευρέως διαδεδομένη στη βιβλιογραφία [22]. Πριν εκπαιδευτεί το μοντέλο, εμφανίζεται ο διάλογος “Resampling Method”, όπου εκεί επιλέγεται αν θα γίνει εξισορρόπηση των κλάσεων και με ποια μέθοδο. Ανάλογα με την επιλογή, το σύνολο εκπαίδευσης προσαρμόζεται ώστε να μειωθεί το πρόβλημα ανισορροπίας κλάσεων που αποτελεί ένα από τα πιο συχνά εμπόδια στην εκπαίδευση μοντέλων IDS, όπου η κανονική κίνηση υπερτερεί αριθμητικά των επιθέσεων κατά πολύ, με αποτέλεσμα το μοντέλο να τείνει να αγνοεί τις μειοψηφικές κλάσεις [16], [29].

Αφού ολοκληρωθεί η προσαρμογή του training set, ξεκινά η εκπαίδευση του ταξινομητή XGBoost, με αυτόματο καθορισμό του τύπου ταξινόμησης ανάλογα με τον αριθμό των κλάσεων της ετικέτας. Παράλληλα, εκτελείται διαδικασία αναζήτησης υπερπαραμέτρων μέσω RandomizedSearchCV, ώστε να επιλεγεί ο καλύτερος συνδυασμός βασικών ρυθμίσεων του μοντέλου. Στη συνέχεια, το τελικό μοντέλο αξιολογείται στο test set με μετρικές όπως accuracy, precision, recall και F1-score, καθώς και με τη χρήση confusion matrix , μετρικές που αποτελούν το καθιερωμένο πλαίσιο αξιολόγησης στη βιβλιογραφία [15], [16], [22], [29], [36].

Ένα επιπλέον στάδιο της διαδικασίας είναι η επεξηγησιμότητα του μοντέλου. Η ανάγκη για ερμηνεύσιμα συστήματα IDS έχει αναδειχθεί ως πρωταρχική απαίτηση στη σύγχρονη βιβλιογραφία, καθώς τα υψηλής απόδοσης μοντέλα τείνουν να λειτουργούν ως «μαύρα κουτιά», χωρίς να παρέχουν στους αναλυτές την κατανόηση που χρειάζονται για τη λήψη αποφάσεων [16], [40]. Για τον σκοπό αυτό, η εφαρμογή αξιοποιεί τη μέθοδο SHAP, ώστε

να αναλυθεί η συνεισφορά των χαρακτηριστικών στις αποφάσεις του μοντέλου και να παραχθούν γραφήματα που βοηθούν στην ερμηνεία των προβλέψεων. Στην βιβλιογραφία έχει χρησιμοποιηθεί συνδυασμός SHAP summary plots και bar plots για την ανάδειξη των κορυφαίων χαρακτηριστικών που συνεισφέρουν περισσότερο στις αποφάσεις του μοντέλου σε πολλαπλές κατηγορίες επίθεσης [15], [36].

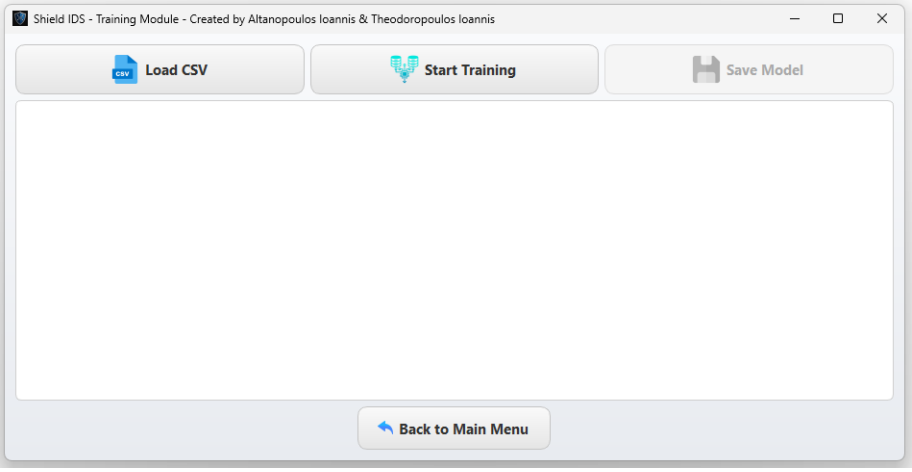
Τέλος, μόλις ολοκληρωθεί η εκπαίδευση και ενημερωθούν οι οπτικές αναφορές, μπορούμε, μέσω του κουμπιού “Save Model”, να αποθηκεύσουμε το εκπαιδευμένο μοντέλο σε αρχείο pickle (.pkl) για χρήση στη live IDS εφαρμογή. Παράλληλα, η διεπαφή εμφανίζει ένα παράθυρο προόδου με spinner και log μηνύματα, ώστε να βλέπουμε σε πραγματικό χρόνο τα βήματα της διαδικασίας, προσφέροντας μια ολοκληρωμένη, καθοδηγούμενη εμπειρία εκπαίδευσης του ανιχνευτή εισβολών.

Και σε αυτή τη περίπτωση, όπως και στην προεπεξεργασία, προκειμένου να αποσαφηνιστούν οι καλύτερες τεχνικές, πραγματοποιήθηκαν όλοι οι δυνατοί συνδυασμοί (Πίνακας 5). Έτσι έχουμε τρείς επιλογές για το κάθε ένα σύνολο δεδομένων και συνολικά εικοσιένα συνδυασμούς.

Α/Α	**Σύνολο δεδομένων**	**1ο Βήμα**	**2ο Βήμα**	**3ο Βήμα**	**4ο Βήμα**	**5ο Βήμα**
1	Wednesday-workingHours_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
2	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
3	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
4	Wednesday-workingHours_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
5	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
6	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
7	Wednesday-workingHours_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid

8	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid
9	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	Διαγραφή διπλότυπων	Αντικατάσταση NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid
10	Wednesday-workingHours_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
11	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
12	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
13	Wednesday-workingHours_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
14	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
15	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
16	Wednesday-workingHours_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid
17	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid
18	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	Διαγραφή διπλότυπων	Διαγραφή NaN	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid
19	UNSW-NB15_1.csv	Διαγραφή διπλότυπων	-	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE
20	UNSW-NB15_1.csv	Διαγραφή διπλότυπων	-	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	SMOTE - ENN
21	UNSW-NB15_1.csv	Διαγραφή διπλότυπων	-	Κωδικοποίηση Κατηγορικών	Επιλογή Χαρ/κών	Hybrid

Πίνακας 5 Σύνολο δυνατών συνδυασμών (προεπεξεργασία - εκπαίδευση)



Εικόνα 29 Αρχική οθόνη εφαρμογής εκπαίδευσης

5.1. Διαχωρισμός δεδομένων σε training και test σύνολο

Η διαδικασία εκπαίδευσης ξεκινά με τη φόρτωση ενός προ επεξεργασμένου συνόλου δεδομένων μέσω του κουμπιού "Load CSV", όπως υλοποιείται από την ομώνυμη συνάρτηση loadCSV(). Έπειτα, ο χρήστης πατώντας το κουμπί "Start Training", καλεί την μέθοδο μέσω της οποίας μπορεί να επιλέξει τη στήλη-στόχο (label) (η Εικόνα 30 παρουσιάζει την επιλογή στήλης – στόχου) μέσω του γραφικού περιβάλλοντος, δηλαδή τη μεταβλητή που το μοντέλο καλείται να προβλέψει. Στην συνέχεια εμφανίζεται παράθυρο επιλογής μεθόδου εξισορρόπησης κλάσεων, οι οποίες αναλύονται διεξοδικά στην ενότητα 6.2. Με την επιλογή της μεθόδου ξεκινά αυτόματα η διαδικασία διαχωρισμού των δεδομένων και εκπαίδευσης του μοντέλου.

```
# Φόρτωση του αρχείου CSV μέσω του κουμπιού Load CSV
def loadCSV(self):
      # Άνοιγμα παραθύρου επιλογής αρχείου από τον δίσκο
      fileName, _ = QFileDialog.getOpenFileName(
            self, "Open CSV", "", "CSV Files (*.csv)")
      if not fileName: # Αν δεν επιλεγεί αρχείο
            return # Τερματισμός διαδικασίας
      try: # Μπλοκ φόρτωσης αρχείου και δεδομένων
            # Φόρτωση των δεδομένων σε DataFrame
            self.df = pd.read_csv(fileName, low_memory=False)
            # Καθαρισμός κενών διαστημάτων από τα ονόματα στηλών
            self.df.columns = self.df.columns.str.strip()
            # Εμφάνιση δεδομένων στον πίνακα του GUI
            self.showData(self.df)
```

```
            # Ενημερωτικό μήνυμα
            QMessageBox.information(self,
                  "CSV Loaded", f"Φορτώθηκε επιτυχώς:\n{fileName}")
      except Exception as e: # Σφάλμα διαδικασίας
            # Ενημερωτικό μήνυμα
            QMessageBox.critical(self, "Error",
                  f"Σφάλμα στη φόρτωση:\n\n{str(e)}")

# Κλήση με το πάτημα του κουμπιού Start Training
def startTraining(self): # Απόσπασμα μεθόδου
      if not hasattr(self, "df"): # Αν δεν έχουν φορτωθεί δεδομένα
            # Ενημερωτικό μήνυμα
            QMessageBox.warning(self, "Warning",
                  "Πρέπει να φορτώσεις CSV πρώτα!")
            return # Τερματισμός
      # Παράθυρο επιλογής της στήλης-στόχου (label)
      label_col, ok = QtWidgets.QInputDialog.getItem(
            self, "Select Label Column",
            "Επέλεξε τη στήλη-στόχο (Label):",
            self.df.columns.tolist(), 0, False
      )
      if not ok: # Αν δεν ολοκληρωθεί η διαδικασία επιλογής
            return # Τερματισμός
      # Παράθυρο επιλογής μεθόδου εξισορρόπησης (resampling)
      resampling_method, ok = QtWidgets.QInputDialog.getItem(
      self, "Resampling Method", "Επέλεξε τεχνική εξισορρόπησης:",
            ["None", "SMOTE", "SMOTEENN", "Hybrid
            (30% Under + SMOTE + ENN)"], 0, False
      )
      if not ok: # Αν δεν ολοκληρωθεί η διαδικασία
            return # Τερματισμός
```

Η εφαρμογή έχει σχεδιαστεί με τέτοιο τρόπο ώστε ο χρήστης να ενημερώνεται σε κάθε στάδιο για την πρόοδο της διαδικασίας μέσω ενός αναδυόμενου παραθύρου με γραφικό περιβάλλον (Εικόνα 31). Έτσι, κατά την εκτέλεση της εκπαίδευσης εμφανίζεται το παράθυρο προόδου που περιλαμβάνει έναν περιστρεφόμενο δείκτη φόρτωσης (spinner), μια κινούμενη μπάρα προόδου και ένα σύστημα καταγραφής μηνυμάτων (log) σε μορφή καρτών. Στο ίδιο παράθυρο εμφανίζονται επίσης τα βήματα της διαδικασίας σε πραγματικό χρόνο όπως ο διαχωρισμός των δεδομένων, η εφαρμογή της μεθόδου resampling, η

αναζήτηση υπερπαραμέτρων και η αξιολόγηση του μοντέλου. Η εκπαίδευση εκτελείται σε ξεχωριστό νήμα εκτέλεσης, ώστε το γραφικό περιβάλλον να παραμένει πλήρως λειτουργικό και να μην παγώνει κατά τη διάρκεια των υπολογισμών. Με την ολοκλήρωση της εκπαίδευσης, ο δείκτης φόρτωσης αντικαθίσταται από εικονίδιο επιτυχίας και η μπάρα προόδου συμπληρώνεται πλήρως, ενημερώνοντας οπτικά τον χρήστη για την επιτυχή ολοκλήρωση της διαδικασίας.

```
# Απόσπασμα συνάρτησης startTraining()
self.progress_dialog = QDialog(self) # Δημιουργία παραθύρου προόδου
# Εμφάνιση / Τίτλος παραθύρου προόδου
self.progress_dialog.setWindowTitle("Training Progress - Shield IDS")
# Αρχικοποίηση spinner στο κέντρο του παραθύρου & Προσθήκη
self.spinner =
      LoadingSpinner(size=70, line_length=12, line_width=5, speed=80)
layout.addWidget(self.spinner, alignment=QtCore.Qt.AlignCenter)
# ProgressBar με ακαθόριστο εύρος (indefinite)
self.progress_bar = QtWidgets.QProgressBar()
self.progress_bar.setRange(0, 0) # Το 0,0 δημιουργεί την κίνηση αναμονής
layout.addWidget(self.progress_bar) # Προσθήκη στο layout
# Αρχικοποίηση για εκτέλεση σε ξεχωριστό νήμα
self.worker = TrainingWorker(self.df, label_col)


# Σύνδεση του σήματος ολοκλήρωσης με τη συνάρτηση τερματισμού
def on_training_done():
      QMessageBox.information(self, "Training",
                              "Training ολοκληρώθηκε επιτυχώς!")
      self.spinner.timer.stop() # Σταμάτα το animation του spinner
      self.spinner.hide() # Απόκρυψη του spinner
      # Σταμάτα την μπάρα και συμπλήρωσέ την
      self.progress_bar.setRange(0, 100) # Εύρος μπάρας στο 100%
      self.progress_bar.setValue(100) # Γέμισμα της μπάρας προόδου
```

```
        # Ετικέτα για το εικονίδιο επιτυχίας
        self.done_icon = QtWidgets.QLabel()
        self.done_icon.setAlignment(QtCore.Qt.AlignCenter)



        self.done_icon.setPixmap(QtGui.QPixmap("icons/check.png").scaled(
              80, 80, QtCore.Qt.KeepAspectRatio,
                    QtCore.Qt.SmoothTransformation
        ))
        # Εμφάνιση εικονιδίου ολοκλήρωσης (πράσινο τικ) - Διάταξη
        self.progress_dialog.layout().insertWidget(2, self.done_icon,
                                           alignment=QtCore.Qt.AlignCenter)


self.worker.start() # Εκκίνηση νέου νήματος
```

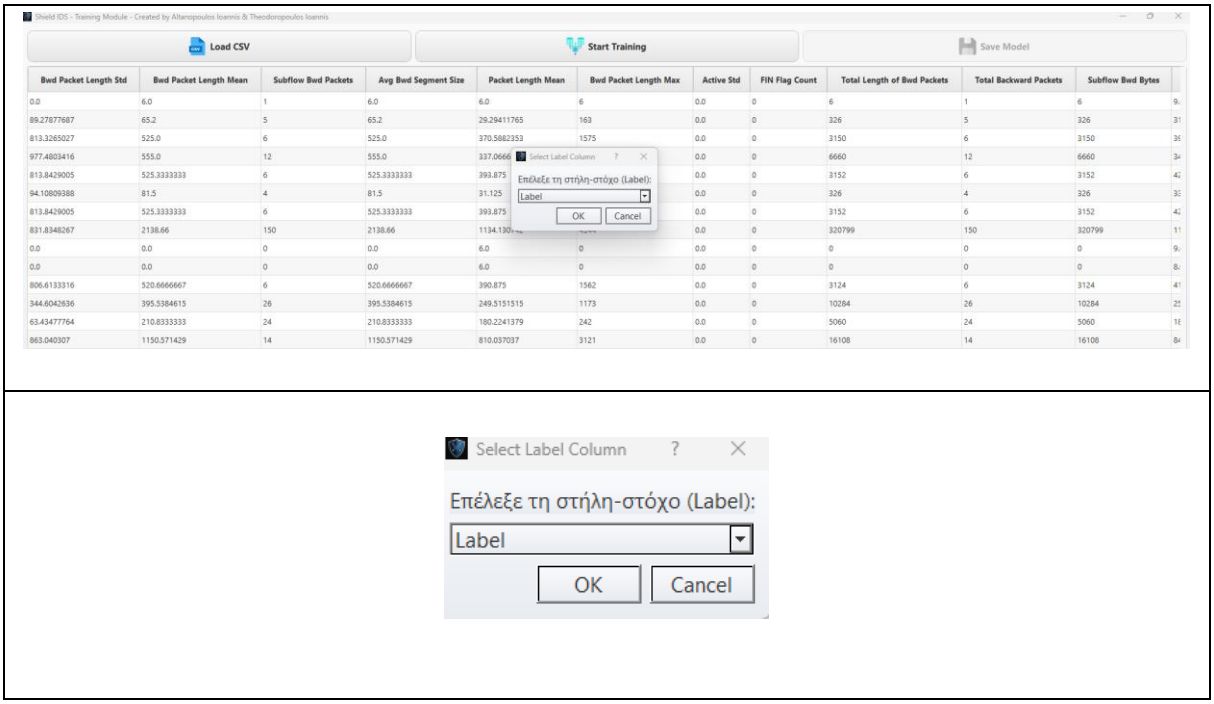
Τα χαρακτηριστικά εισόδου αποθηκεύονται στη μεταβλητή X, ενώ η στήλη-στόχος, δηλαδή η μεταβλητή που το μοντέλο καλείται να προβλέψει, αποθηκεύεται στη μεταβλητή y. Όπως προαναφέρθηκε, η στήλη αυτή επιλέγεται από τον χρήστη μέσω του γραφικού περιβάλλοντος της εφαρμογής. Με αυτόν τον τρόπο, η εφαρμογή δεν είναι δεσμευμένη σε ένα συγκεκριμένο σύνολο δεδομένων, αλλά μπορεί να χρησιμοποιηθεί με διαφορετικά σύνολα αρκεί ο χρήστης να ορίσει τη σωστή στήλη ετικέτας.

```
# Χαρακτηριστικά εισόδου (X) χωρίς τη στήλη-στόχο
X = self.df.drop(columns=[self.label_col])
y = self.df[self.label_col] # Στήλη στόχος (y) - Ετικέτες
# Αν οι ετικέτες είναι κείμενο
if y.dtype == 'object':
      # Μετατροπή σε αριθμητικές τιμές
      y = LabelEncoder().fit_transform(y)
```

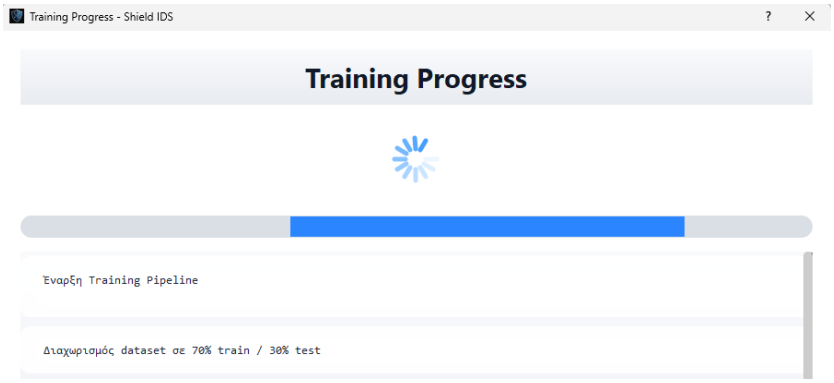
Ο διαχωρισμός των δεδομένων πραγματοποιείται με τη μέθοδο train_test_split της βιβλιοθήκης scikit-learn. Συγκεκριμένα, το 70% των δεδομένων χρησιμοποιείται για την εκπαίδευση του μοντέλου, ενώ το υπόλοιπο 30% διατηρείται ως σύνολο ελέγχου. Το

σύνολο εκπαίδευσης αξιοποιείται για την προσαρμογή του αλγορίθμου XGBoost στα δεδομένα, ενώ το σύνολο ελέγχου χρησιμοποιείται μετά την ολοκλήρωση της εκπαίδευσης, ώστε να μετρηθεί η ικανότητα γενίκευσης του μοντέλου. Η διαδικασία αυτή συμβάλλει στη μείωση του κινδύνου υπερπροσαρμογής (overfitting), δηλαδή της κατάστασης όπου το μοντέλο απομνημονεύει τα δεδομένα εκπαίδευσης αντί να μαθαίνει γενικεύσιμα μοτίβα, με αποτέλεσμα να αποτυγχάνει σε νέα δεδομένα [5]. Επιπλέον, κατά τη διαδικασία του διαχωρισμού χρησιμοποιείται η παράμετρος stratify με τη βοήθεια της οποίας εξασφαλίζεται ότι η κατανομή των κλάσεων θα παραμείνει αναλογικά ίδια, τόσο στο training set, όσο και στο test set. Αυτό είναι ιδιαίτερα σημαντικό σε προβλήματα ανίχνευσης εισβολών, όπου η κανονική κίνηση υπερτερεί αριθμητικά των επιθέσεων [16], [29].

```
# Καταγραφή διαχωρισμού στο log
self.log("Διαχωρισμός dataset σε 70% train / 30% test")
# Δημιουργία συνόλων εκπαίδευσης και ελέγχου
X_train, X_test, y_train, y_test = train_test_split(
      X, y, test_size=0.3, random_state=42, stratify=y
)
```



Εικόνα 30 Επιλογή στήλης στόχου για την εκκίνηση της εκπαίδευσης



Εικόνα 31 Παράθυρο για την ενημέρωση της προόδου της διαδικασίας

5.2. Το πρόβλημα του "Data Imbalance", διαχείριση ανισορροπίας κλάσεων

Όταν ο λόγος ανισορροπίας (Imbalance Ratio – IR) μεταξύ της μειοψηφικής και της πλειοψηφικής κλάσης είναι χαμηλός, η μειοψηφική κλάση τείνει να αντιμετωπίζεται ως θόρυβος και να αγνοείται από τον αλγόριθμο εκπαίδευσης. Αυτό έχει ως αποτέλεσμα την ύπαρξη μεροληψίας του μοντέλου υπέρ της πλειοψηφικής κλάσης, οδηγώντας στην αύξηση των Ψευδώς Θετικών (False Positive) και στην μείωση των Αληθώς Θετικών (True Positive) που αποτελούν ακριβώς εκείνες τις προβλέψεις με την υψηλότερη πρακτική σημασία στο πεδίο της ανίχνευσης εισβολών [28].

Χαρακτηριστικό παράδειγμα αποτελεί το σύνολο δεδομένων CIC-IDS2017 που αξιοποιεί η εφαρμογή μας, στο οποίο οι επιθέσεις τύπου SQL Injection αριθμούν μόλις 21 εγγραφές έναντι 168.186 δειγμάτων κανονικής κίνησης. Βάση της βιβλιογραφίας, αυτό αντιμετωπίστηκε εφαρμόζοντας random under-sampling για την πλειοψηφική κλάση και SMOTE oversampling για τις μειοψηφικές κλάσεις επιθέσεων, επιτυγχάνοντας ισορροπημένο training set [16].

Οι λύσεις στο πρόβλημα της ανισορροπίας κατηγοριοποιούνται σε επίπεδο δεδομένων και σε αλγοριθμικό επίπεδο. Στο επίπεδο δεδομένων εφαρμόζονται τεχνικές δειγματοληψίας (sampling) όπως υπό-δειγματοληψία (undersampling) της πλειοψηφικής κλάσης, υπερ-δειγματοληψία (oversampling) της μειοψηφικής κλάσης, ή συνδυασμός και των δύο (hybrid sampling). Στο αλγοριθμικό επίπεδο χρησιμοποιούνται βελτιστοποιημένοι αλγόριθμοι για ανισόρροπα δεδομένα, όπως cost-sensitive και ensemble μοντέλα. Οι τεχνικές επιπέδου δεδομένων διορθώνουν την ανισορροπία επιτρέποντας τη χρήση τυπικών μοντέλων μηχανικής μάθησης, ενώ παράλληλα μπορούν να συνδυαστούν με συνδυαστικά μοντέλα (ensemble) για περαιτέρω βελτίωση της απόδοσης [28].

Από τις 55 διαφορετικές τεχνικές δειγματοληψίας που εντοπίστηκαν στη βιβλιογραφία, η υπερ-δειγματοληψία (oversampling) παραμένει η πιο αποτελεσματική κατηγορία, ενώ οι υβριδικές τεχνικές εμφανίζουν την υψηλότερη σχετική απόδοση. Μεταξύ των υβριδικών αυτών μεθόδων, ο συνδυασμός SMOTE και ENN (Edited Nearest Neighbours) εντοπίζεται στη βιβλιογραφία ως μια από τις πιο συχνά εφαρμοζόμενες προσεγγίσεις, με αξιόλογα αποτελέσματα σε εφαρμογές μηχανικής μάθησης με ανισόρροπα δεδομένα [28]. Επιπλέον τεκμηριώνεται ότι η καθαρή υπό-δειγματοληψία (pure undersampling) συνεπάγεται απώλεια χρήσιμης πληροφορίας, ενώ η αποκλειστική χρήση υπερ-δειγματοληψίας μπορεί να προκαλέσει τεχνητή μεροληψία ή overfitting ιδιαίτερα σε δεδομένα με υψηλό IR [28]. Αποδεικνύεται στατιστικά, μέσω των Friedman και Nemenyi tests, ότι η επιλογή της τεχνικής resampling επηρεάζει σημαντικά τα αποτελέσματα των μετρικών και τον ρυθμό ψευδών συναγερμών (FAR) σε δυαδικά μοντέλα ταξινόμησης. Ωστόσο, στα πολύ κλασικά (multiclass) μοντέλα δεν εντοπίστηκαν στατιστικά σημαντικές διαφορές μεταξύ των τεχνικών, υποδεικνύοντας ότι η απόδοση εξαρτάται από το αν η ταξινόμηση είναι δυαδική ή πολύ κλασική [5].

Με βάση τα παραπάνω η εφαρμογή μας παρέχει τρεις διακριτές τεχνικές resampling την SMOTE, SMOTEENN και μια Υβριδική μέθοδο οι οποίες αναλύονται διεξοδικά στις παρακάτω υπό ενότητες, τόσο ως προς τη θεωρητική τους βάση όσο και ως προς τον τρόπο υλοποίησής τους στον κώδικα του συστήματος.

5.2.1. Τεχνική SMOTE

Όπως προαναφέρθηκε η ανισορροπία των κλάσεων αποτελεί ένα από τα βασικότερα προβλήματα στα σύνολα δεδομένων για ανίχνευση εισβολών, καθώς οι κανονικές ροές δεδομένων είναι σαφώς περισσότερες από τις κακόβουλες. Η τεχνική SMOTE αντιμετωπίζει αυτό το ζήτημα δημιουργώντας συνθετικά δείγματα για τις μειονοτικές κατηγορίες, με αποτέλεσμα την εξισορρόπηση του συνόλου δεδομένων κατά την εκπαίδευση [16]. Αυτή η προσέγγιση βελτιώνει σημαντικά την ικανότητα γενίκευσης των μοντέλων, ενισχύοντας τη δυνατότητά τους να αναγνωρίζουν επιθέσεις. Επιπλέον, η χρήση SMOTE έχει αποδειχθεί ότι βελτιώνει κρίσιμες μετρικές επίδοσης, όπως η ακρίβεια (precision) καθιστώντας τα μοντέλα πιο ευαίσθητα και αξιόπιστα [5]. Σε ορισμένες προσεγγίσεις, όπως στο σύνολο δεδομένων CICIDS-2017, η εφαρμογή του SMOTE σε συνδυασμό με τεχνικές υπό-δειγματοληψίας για την υπερ-εμφάνιση της κυρίαρχης «Normal» κλάσης έχει οδηγήσει σε σημαντική βελτίωση της απόδοσης και της

σταθερότητας των αποτελεσμάτων, ενώ παράλληλα ενισχύει την ερμηνευσιμότητα μέσω προχωρημένων μεθόδων επεξηγηματικής ανάλυσης, όπως η SHAP [16]. Με αυτό τον τρόπο, η διαχείριση της ανισορροπίας συμβάλλει όχι μόνο στη βελτίωση της ανίχνευσης επιθέσεων αλλά και στην αυξημένη διαφάνεια των συστημάτων.

Προκειμένου να υλοποιηθεί η τεχνική, αρχικά εισάγουμε στο κώδικα τη βιβλιοθήκη με τη βοήθεια της οποίας θα χρησιμοποιήσουμε τον αλγόριθμο.

```
# Εισαγωγή της κλάσης SMOTE από τη βιβλιοθήκη imbalanced-learn
from imblearn.over_sampling import SMOTE
```

Στη συνέχεια, όταν ο χρήστης επιλέξει SMOTE από το γραφικό περιβάλλον (βλέπε Εικόνα 32), εκτελείται το παρακάτω τμήμα κώδικα εντός της μεθόδου run:

```
self.resampling_method == "SMOTE": # Αρχικοποίηση μεθόδου
# Καταγραφή της έναρξης της διαδικασίας στο log της εφαρμογής
self.log("Εφαρμογή SMOTE...")
# Αρχικοποίηση αλγορίθμου με σταθερό random_state για επαναληψιμότητα
sm = SMOTE(random_state=42)
# Εφαρμογή αλγορίθμου στα δεδομένα (παραγωγή συνθετικών δειγμάτων)
X_train, y_train = sm.fit_resample(X_train, y_train)
# Ενημέρωση των μετρητών κλάσεων μετά την εξισορρόπηση
self.class_counts_after_train = pd.Series(y_train).value_counts()
# Καταγραφή διαστάσεων του πίνακα δεδομένων μετά την αύξηση των δειγμάτων
self.log(f"   Νέο μέγεθος μετά το SMOTE: {X_train.shape}\n")
```

Μόλις η διαδικασία ολοκληρωθεί παράγονται αυτόματα, μέσω της βιβλιοθήκης matplotlib (αναλύεται στο κεφάλαιο 9.1.1), τα γραφήματα κατανομής των κλάσεων πριν και μετά την ανά δειγματοληψία:

```
# Τα αρχικά δεδομένα (Απόσπασμα κώδικα)
before_counts = class_counts_before_train.sort_index()
# Τα δεδομένα μετά το SMOTE
after_counts = class_counts_after_train.sort_index()
# Δημιουργία του σχήματος με matplotlib
fig = plt.Figure(figsize=(14, 6), dpi=150) # Ορισμός διαστάσεων γραφήματος
```

```
ax = fig.add_subplot(111)

# Σχεδιασμός της μπάρας "Πριν"

bars_before = ax.bar(before_counts.index.astype(str),

                     before_counts.values, alpha=0.30,

                     color=blue_shield, label="Πριν"

                     )

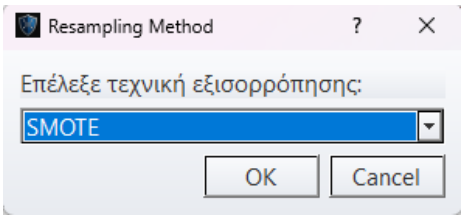
# Σχεδιασμός της αύξησης (π.χ. από SMOTE) πάνω από την αρχική μπάρα

ax.bar(

      after_counts.index.astype(str), diff.clip(lower=0).values,

      bottom=before_counts.values, color=orange_alert, label="Αύξηση"

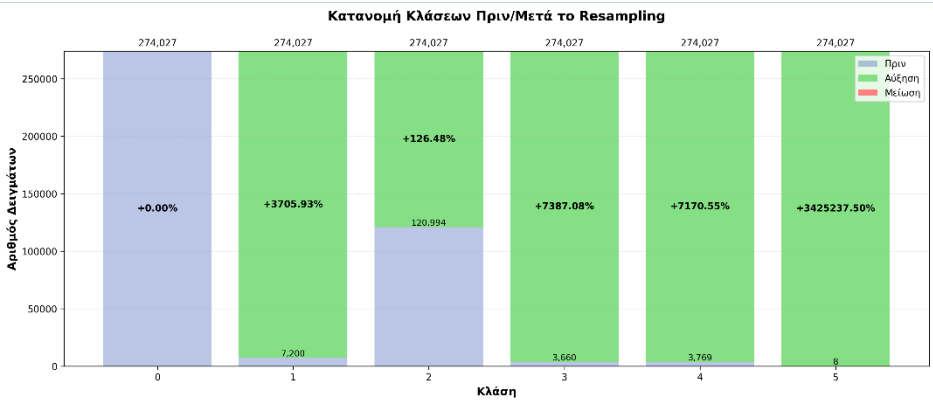
      )
```



Εικόνα 32 Επιλογή της τεχνικής SMOTE από το γραφικό περιβάλλον

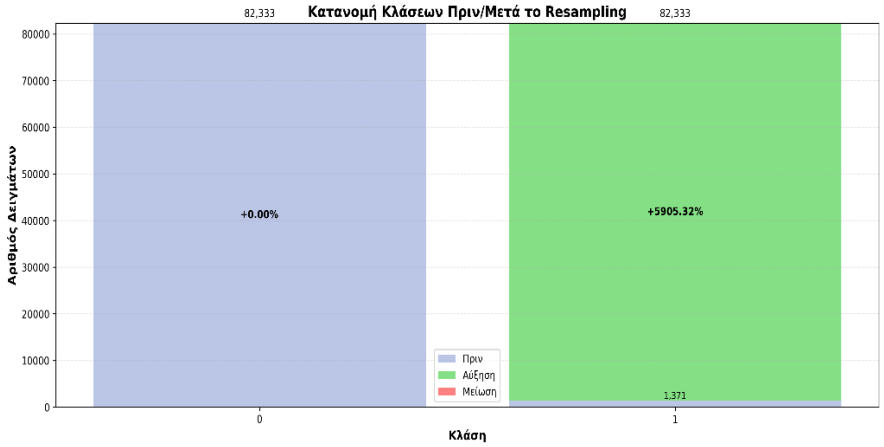
Παρακάτω παραθέτουμε τα αποτελέσματα των συνδυασμών (όπως δείχνει ο Πίνακας 5) τα οποία αφορούν την τεχνική SMOTE (Εικόνα 33, Εικόνα 34, Εικόνα 35, Εικόνα 36, Εικόνα 37, Εικόνα 38, Εικόνα 39).

Η μπλε μπάρα αντιστοιχεί στον αρχικό αριθμό δειγμάτων ανά κλάση, ενώ η πράσινη περιοχή δείχνει πόσο αυξήθηκαν τα δείγματα μετά την εφαρμογή της επιλεγμένης μεθόδου εξισορρόπησης SMOTE μαζί με το ποσοστό μεταβολής πάνω σε κάθε μπάρα.



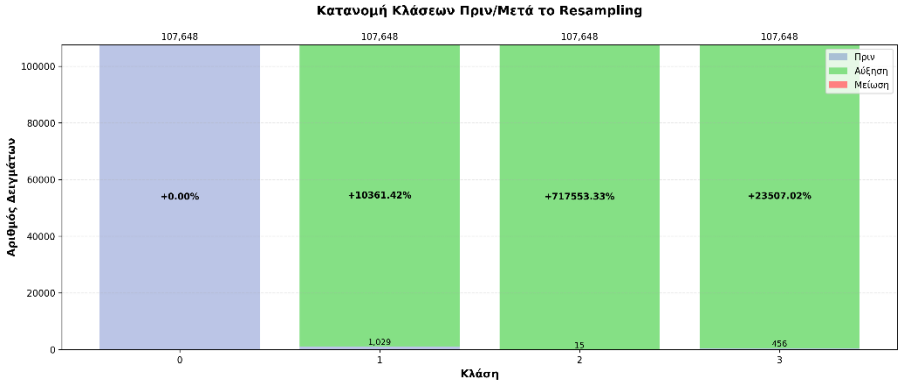
Εικόνα 33 1ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις.

Η μπλε μπάρα δείχνει τον αρχικό αριθμό δειγμάτων για την κλάση 0 (π.χ. κανονική κίνηση) και για την κλάση 1 (επιθέσεις), ενώ η πράσινη περιοχή απεικονίζει πόσο αυξήθηκαν τα δείγματα της μειοψηφικής κλάσης μετά την εφαρμογή της επιλεγμένης μεθόδου εξισορρόπησης. Στο γράφημα φαίνεται ότι η κλάση 1 ενισχύθηκε σημαντικά (+5905%), με αποτέλεσμα ένα πολύ πιο ισορροπημένο training set, που βοηθά το μοντέλο να «βλέπει» αρκετά παραδείγματα επιθέσεων και να μην εκπαιδεύεται σχεδόν μόνο πάνω στην κανονική κίνηση.



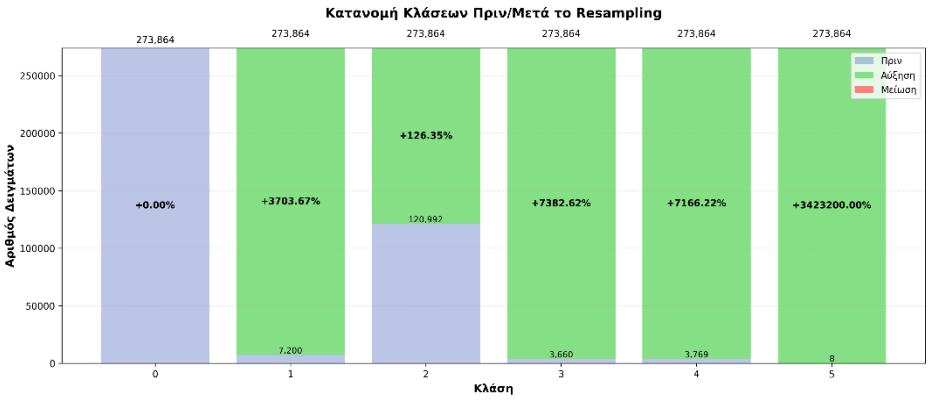
Εικόνα 34 2ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE. Κατανομή της κλάσεων πριν και μετά το resampling.

Οι μπλε μπάρες αντιστοιχούν στον αρχικό αριθμό δειγμάτων ανά κλάση, ενώ οι πράσινες εκφράζουν πόσο αυξήθηκαν οι μειονοτικές κλάσεις μετά την εφαρμογή της μεθόδου εξισορρόπησης. Πάνω από κάθε μπάρα αναγράφεται το ποσοστό μεταβολής, το οποίο είναι ιδιαίτερα υψηλό για τις πιο σπάνιες επιθέσεις, δείχνοντας ότι το training set έγινε πολύ πιο ισορροπημένο ώστε το μοντέλο να εκπαιδευτεί επαρκώς και σε σπάνιους τύπους συμβάντων.



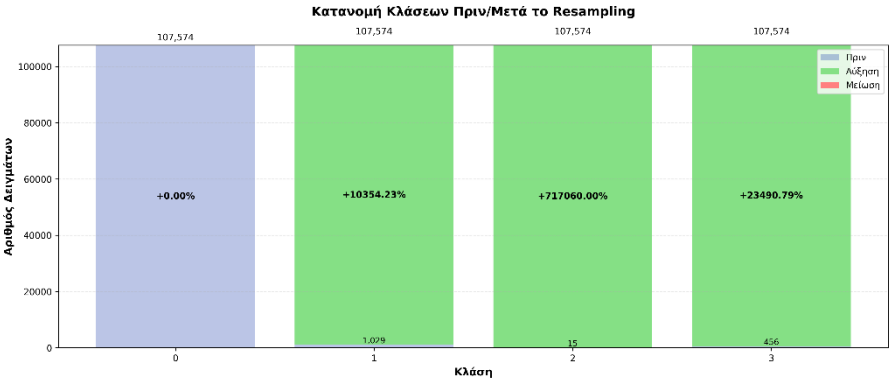
Εικόνα 35 3ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 4 κλάσεις.

Η κλάση 0 διατηρεί τον ίδιο αριθμό δειγμάτων, ενώ ενισχύονται όλες οι επιθετικές κλάσεις (1–5), από +3 700% έως και πάνω από +34 000 000% για την πιο σπάνια κλάση 5. Το αποτέλεσμα είναι ένα πλήρως εξισορροπημένο training set, με 273 864 δείγματα σε κάθε κλάση, επιτρέποντας έτσι στο μοντέλο να μάθει με την ίδια λεπτομέρεια τόσο την κανονική κίνηση όσο και όλες τις υποκατηγορίες DoS.



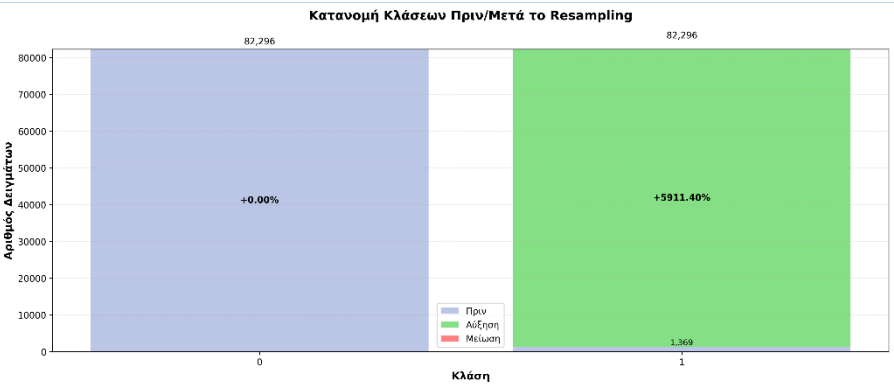
Εικόνα 36 10ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Smote. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις

Η κλάση 0 διατηρεί τον ίδιο αριθμό δειγμάτων (107 574), ενώ οι τρεις επιθετικές κλάσεις 1, 2 και 3 αυξάνονται θεαματικά (έως και +7 170 600% για την πιο σπάνια κλάση 2), οδηγώντας σε πλήρη εξισορρόπηση με 107 574 δείγματα ανά κλάση.

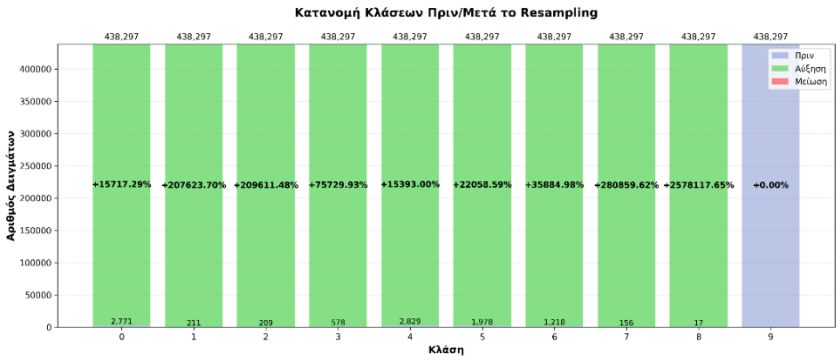


Εικόνα 37 11ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE. Κατανομή της κλάσεων πριν και μετά το resampling.

Η κλάση 0 διατηρεί 82 296 δείγματα, ενώ η PortScan κλάση 1 αυξάνεται κατά περίπου 5 911%, από 1 369 σε 82 296 δείγματα, οδηγώντας σε πλήρη εξισορρόπηση των δύο κατηγοριών στο training set.



Εικόνα 38 12ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling.

Όλες οι επιθετικές κλάσεις 0–8 αυξάνονται σε ~438 297 δείγματα η καθεμία (π.χ. +207 623.70% για την κλάση 1, +2 09611.48% για την κλάση 2, +2 80859.62% για την κλάση 7 και +2 578 117.65% για την εξαιρετικά σπάνια κλάση 8), ενώ η ήδη πολύ μεγάλη κανονική κλάση 9 παραμένει σταθερή, δημιουργώντας ισορροπημένο training set.	

Εικόνα 39 19ος Συνδυασμός: Dataset UNSW-NB15_1.csv με μέθοδο εξισορρόπησης SMOTE. Κατανομή των κλάσεων πριν και μετά το resampling.

5.2.2. Τεχνική SMOTEENN

Το ENN είναι μια τεχνική υπό δειγματοληψίας που ανήκει στις μεθόδους «under-sampling» και εφαρμόζεται ως προεπεξεργασία σε ανισόρροπα σύνολα δεδομένων με στόχο τη βελτίωση της απόδοσης των ML αλγορίθμων, αφαιρώντας πρότυπα της πλειοψηφικής κλάσης που εντοπίζονται κοντά στα όρια μεταξύ κατηγοριών ή θεωρούνται θόρυβος. Ο μηχανισμός της μεθόδου βασίζεται στον εντοπισμό και την αφαίρεση των δειγμάτων που ταξινομούνται λάθος από τους k πλησιέστερους γείτονές τους, με αποτέλεσμα να διατηρούνται μόνο τα πιο «καθαρά» και τυπικά σημεία της κάθε κατηγορίας. Με αυτόν τον τρόπο, μειώνεται ο κίνδυνος υπερ προσαρμογής του μοντέλου υπέρ της πλειονότητας, ενισχύεται ο διαχωρισμός μεταξύ των κατηγοριών και μειώνεται η προκατάληψη κατά της μειονότητας, διατηρώντας τα σημαντικά πρότυπα για τη βελτίωση της γενίκευσης του αλγορίθμου. Η τεχνική ENN χρησιμοποιείται είτε ως αυτόνομη μέθοδος είτε σε συνδυασμό με τεχνικές oversampling όπως το SMOTE ή το ADASYN (υβριδικά σχήματα), συμβάλλοντας στην επίτευξη καλύτερης ισορροπίας και καθαρών ορίων στην ταξινόμηση, ειδικά σε εφαρμογές υγείας, οικονομίας και άλλους τομείς εφαρμοσμένης μηχανικής [28].

Προκειμένου να υλοποιηθεί η τεχνική, αρχικά εισάγουμε στο κώδικα τη βιβλιοθήκη με τη βοήθεια της οποίας θα χρησιμοποιήσουμε τον αλγόριθμο.

```
# Εισαγωγή της κλάσης SMOTEEN από τη βιβλιοθήκη imbalanced-learn
from imblearn.over_sampling import SMOTEENN
```

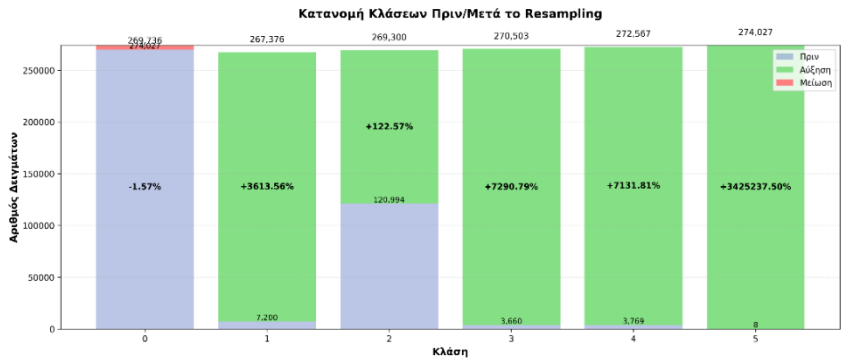
Στη συνέχεια, όταν ο χρήστης επιλέξει SMOTEENN από το γραφικό περιβάλλον, εκτελείται το παρακάτω τμήμα κώδικα εντός της μεθόδου run:

```
self.resampling_method == "SMOTEENN": # Αρχικοποίηση μεθόδου
```

```
# Καταγραφή της έναρξης της διαδικασίας στο log της εφαρμογής
self.log("Εφαρμογή SMOTE + ENN...")
# Αρχικοποίηση αλγορίθμου με σταθερό random_state για επαναληψιμότητα
smoteenn = SMOTEENN(random_state=42)
# Εφαρμογή αλγορίθμου στα δεδομένα (παραγωγή συνθετικών δειγμάτων)
X_train, y_train = smoteenn.fit_resample(X_train, y_train)
# Ενημέρωση των μετρητών κλάσεων μετά την εξισορρόπηση
self.class_counts_after_train = pd.Series(y_train).value_counts()
# Καταγραφή διαστάσεων του πίνακα δεδομένων μετά την αύξηση των δειγμάτων
self.log(f"   Νέο μέγεθος μετά το SMOTEENN: {X_train.shape}\n")
```

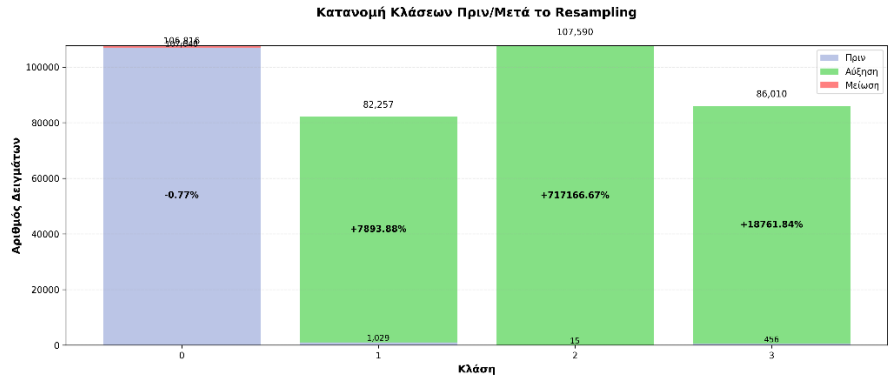
Μόλις η διαδικασία ολοκληρωθεί παράγονται αυτόματα, τα γραφήματα κατανομής των κλάσεων πριν και μετά την ανά δειγματοληψία. Παρακάτω παραθέτουμε τα αποτελέσματα των συνδυασμών (Πίνακας 5) που αφορούν την τεχνική SMOTEENN (Εικόνα 40, Εικόνα 41, Εικόνα 42, Εικόνα 43, Εικόνα 44, Εικόνα 45, Εικόνα 46).

Μετά το resampling οι σπάνιες κλάσεις (1–5) αυξήθηκαν σε πλήθος δειγμάτων, με υψηλά ποσοστά ενίσχυσης, ενώ η κλάση 0 μειώθηκε ελαφρώς. Το αποτέλεσμα είναι ένα σχεδόν πλήρως εξισορροπημένο training set, που μειώνει την προκατάληψη υπέρ της κανονικής κίνησης και επιτρέπει στο μοντέλο να μάθει αναλυτικά κάθε είδος επίθεσης.



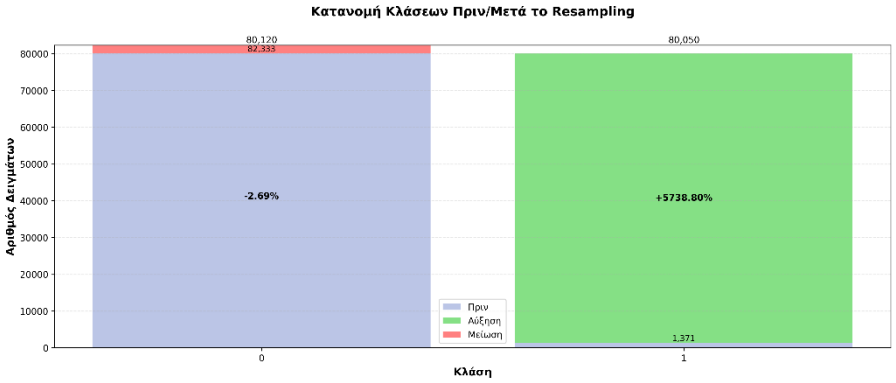
Εικόνα 40 4ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις.

Μετά το resampling, οι κλάσεις 1, 2 και 3 αυξάνονται κατά χιλιάδες τοις εκατό σε πλήθος δειγμάτων, ενώ η κλάση 0 μειώνεται ελαφρώς. Έτσι προκύπτει πολύ πιο ισορροπημένο σύνολο εκπαίδευσης, που αποτρέπει την κυριαρχία της κανονικής κίνησης και βελτιώνει την ικανότητα του μοντέλου να μαθαίνει σωστά κάθε τύπο web attack.



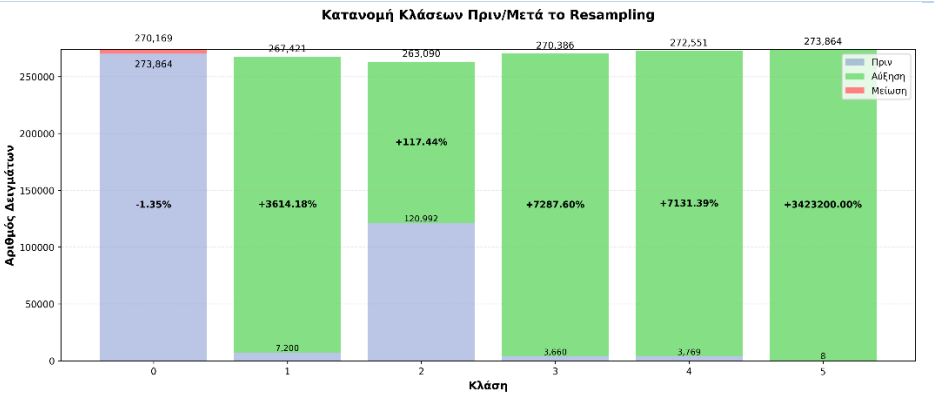
Εικόνα 41 5ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή της κλάσεων πριν και μετά το resampling.

Η κλάση 0 (normal) μειώθηκε ελαφρώς, ενώ η κλάση 1 (PortScan) αυξήθηκε πάνω από 5700% σε αριθμό δειγμάτων. Έτσι το training set γίνεται σχεδόν ισορροπημένο, κάτι που μειώνει την προκατάληψη υπέρ της κανονικής κίνησης και επιτρέπει στο μοντέλο να μάθει λεπτομερώς τα χαρακτηριστικά των PortScan ροών.



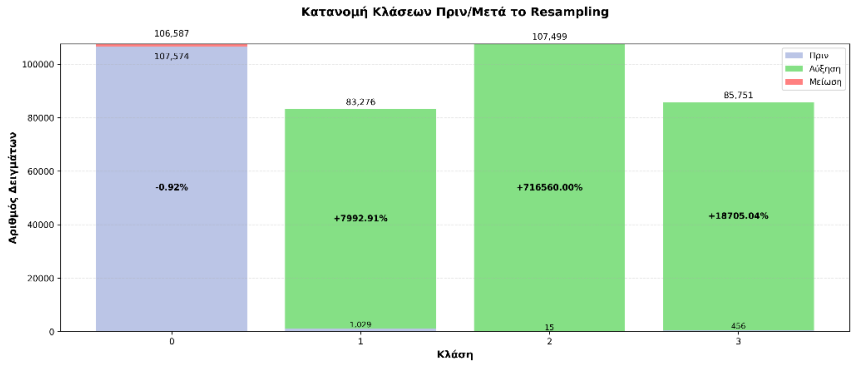
Εικόνα 42 6ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling.

Η κλάση 0 μειώθηκε ελαφρά (-1.35%), ενώ όλες οι επιθετικές κλάσεις ενισχύθηκαν από +117% έως και πάνω από +34 000 000% για την πιο σπάνια κλάση 5, με τελικό αποτέλεσμα περίπου 270–273 χιλιάδες δείγματα σε κάθε κλάση.



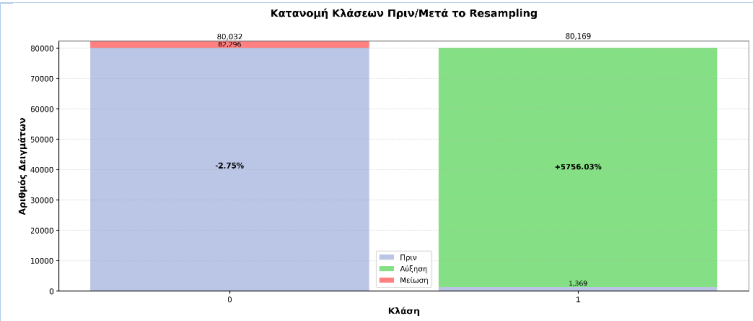
Εικόνα 43 13ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις.

Η κλάση 0 μειώνεται ελάχιστα (-0.92%), ενώ οι επιθετικές κλάσεις 1, 2 και 3 ενισχύονται δραματικά: περίπου +8 000% για την κλάση 1, πάνω από +716 000% για τη σπάνια κλάση 2 και περίπου +18 700% για την κλάση 3, οδηγώντας σε σχεδόν ίσο αριθμό δειγμάτων γύρω στις 80–107 χιλιάδες ανά κλάση.



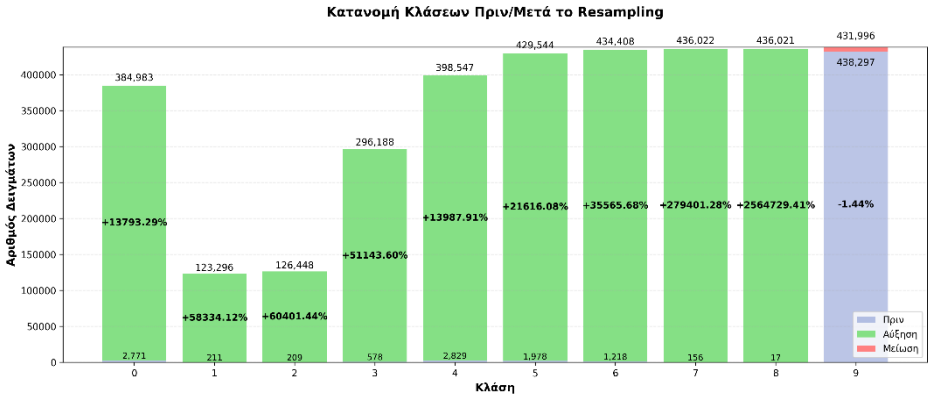
Εικόνα 44 14ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή της κλάσεων πριν και μετά το resampling.

Η κλάση 0 μειώνεται ελαφρά από 82 296 σε 80 032 δείγματα (-2.75%), ενώ η κλάση 1 αυξάνεται από 1 369 σε 80 169 δείγματα (+5 756.03%), οδηγώντας σε σχεδόν πλήρως ισορροπημένο training set μεταξύ normal και PortScan.



Εικόνα 45 15ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling.

Όλες οι επιθετικές κλάσεις 0–8 αυξάνονται: η κλάση 1 κατά +58 334.12%, η 2 κατά +60 401.44%, η 7 κατά +279 401.28% και η 8 κατά περίπου +2 564 729.41%, ενώ η κανονική κλάση 9 μειώνεται ελαφρά κατά 1.44%, οδηγώντας σε πολύ πιο ισορροπημένο training set χωρίς να «πνίγεται» από normal δείγματα.



Εικόνα 46 20ος Συνδυασμός: Dataset UNSW-NB15_1.csv με μέθοδο εξισορρόπησης SMOTE-ENN. Κατανομή των κλάσεων πριν και μετά το resampling

5.2.3. Τεχνική Hybrid (Under sampling + SMOTE + ENN)

Η υβριδική προσέγγιση συνδυάζει διαδοχικά under-sampling και over-sampling ώστε να αντιμετωπίσει πιο επιθετικά την έντονη ανισορροπία και τον θόρυβο στα δεδομένα ανίχνευσης εισβολών [16], [28]. Αρχικά, μειώνεται σκόπιμα το μέγεθος της πλειοψηφικής κλάσης (π.χ. «Normal») [16] το οποίο ορίστηκε περίπου στο 30% του αρχικού πλήθους, με ένα ελεγχόμενο υπό-δειγματοληπτικό βήμα, ώστε να περιοριστεί η κυριαρχία της χωρίς να χαθεί πλήρως η ποικιλία των φυσιολογικών ροών. Στη συνέχεια εφαρμόζεται SMOTE για να ενισχυθούν οι μειονοτικές κατηγορίες με συνθετικά δείγματα [16], [28] ενώ, στο τελικό στάδιο, η τεχνική ENN χρησιμοποιείται για να «καθαρίσει» το αποτέλεσμα, αφαιρώντας παραδείγματα που θεωρούνται θορυβώδη ή ασαφή στα όρια μεταξύ κλάσεων [28]. Με αυτό το τριπλό βήμα, το μοντέλο εκπαιδεύεται σε ένα σύνολο δεδομένων που είναι ταυτόχρονα πιο ισορροπημένο και απαλλαγμένο από ακραίες ή παραπλανητικές παρατηρήσεις, κάτι που στην πράξη οδηγεί σε πιο σταθερή συμπεριφορά, καλύτερη διάκριση μεταξύ κανονικής και κακόβουλης κίνησης και βελτιωμένες μετρικές ευαισθησίας και αξιοπιστίας σε περιβάλλοντα IDS [16], [28].

Προκειμένου να υλοποιηθεί η τεχνική, αρχικά εισάγουμε στο κώδικα τις βιβλιοθήκες με τη βοήθεια των οποίων θα χρησιμοποιήσουμε τον αλγόριθμο.

```
# Βιβλιοθήκες για την τεχνική Hybrid
from imblearn.combine import SMOTEENN # Για το συνδυασμό SMOTE και ENN
from imblearn.over_sampling import SMOTE # Για την υπέρ δειγματοληψία
# Για την υπό δειγματοληψία και τον καθαρισμό
from imblearn.under_sampling import RandomUnderSampler,
EditedNearestNeighbours
```

Στη συνέχεια, όταν ο χρήστης επιλέξει Hybrid από το γραφικό περιβάλλον, εκτελείται το παρακάτω τμήμα κώδικα εντός της μεθόδου run:

```
# Αρχικοποίηση μεθόδου
self.resampling_method == "Hybrid (30% Under + SMOTE + ENN)":
# Καταγραφή της έναρξης της διαδικασίας στο log της εφαρμογής
self.log("Εφαρμογή HYBRID: 30% UnderSampling → SMOTE → ENN")
```

Σε αυτό το βήμα, η εφαρμογή μειώνει την πλειοψηφική κλάση στο 30% του αρχικού της μεγέθους:

```
# Σύνολο δειγμάτων για την πλειοψηφική κλάση (maj_label)
```

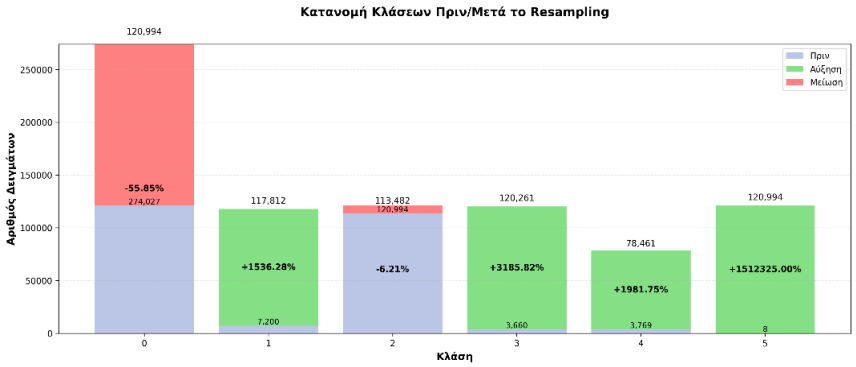
```
class_counts_before = pd.Series(y_train).value_counts()
# Ετικέτα (στήλη) στην οποία περιέχεται η πλειοψηφία
maj_label = class_counts_before.idxmax()
# Σύνολο πλειοψηφικής κλάσης
maj_count = class_counts_before.max()
# Στόχος το 30% του συνόλου
target_maj = max(int(maj_count * 0.3), 100)
# Καταγραφή του μέγεθος της πλειοψηφίας στο log μετά το undersampling
self.log(f"Πλειοψηφία πριν: {maj_count}, στόχος: {target_maj}")
# Αντικείμενο για τμηματικό undersampling της πλειοψηφικής κλάσης
rus = RandomUnderSampler(
      # Μείωση πλειοψηφικής κλάσης βάση ετικέτας και στόχου
      sampling_strategy={maj_label: target_maj},
      random_state=42 # Αρχικοποίηση γεννήτριας
)
# Undersampling (μείωση πλειοψηφίας, επιστρέφει τα νέα X/y)
X_under, y_under = rus.fit_resample(X_train, y_train)
```

Μετά τη μείωση της πλειοψηφίας, εφαρμόζεται ο συνδυασμός SMOTE και ENN για την τελική εξισορρόπηση:

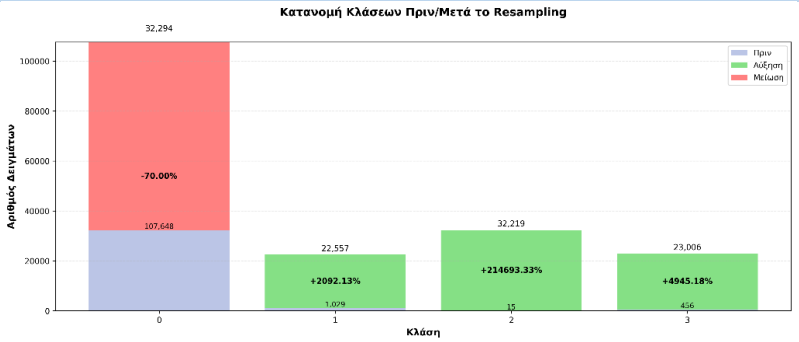
```
# SMOTE για oversampling των ελάχιστων κλάσεων (δημιουργία δειγμάτων)
smote = SMOTE(random_state=42)
# ENN για καθαρισμό των παραπλήσιων/θορυβωδών δειγμάτων
enn = EditedNearestNeighbours(n_neighbors=3)
# Συνδυασμός SMOTE + ENN (hybrid): oversample + clean
smoteenn = SMOTEENN(smote=smote, enn=enn, random_state=42)
# Εφαρμογή hybrid SMOTEENN στο undersampled σύνολο
X_train, y_train = smoteenn.fit_resample(X_under, y_under)
# Σύνολο κλάσεων μετά την επεξεργασία (για έλεγχο ισορροπίας)
self.class_counts_after_train = pd.Series(y_train).value_counts()
```

```
# Καταγραφή του νέου σχήματος στο log
self.log(f"Τελικό μέγεθος Hybrid: {X_train.shape}\n")
```

Μόλις η διαδικασία ολοκληρωθεί παράγονται τα γραφήματα κατανομής των κλάσεων πριν και μετά την ανά δειγματοληψία. Παρακάτω παραθέτουμε τα αποτελέσματα των συνδυασμών (Πίνακας 5) που αφορούν την τεχνική Hybrid (Εικόνα 47, Εικόνα 48, Εικόνα 49, Εικόνα 50, Εικόνα 51, Εικόνα 52, Εικόνα 53).

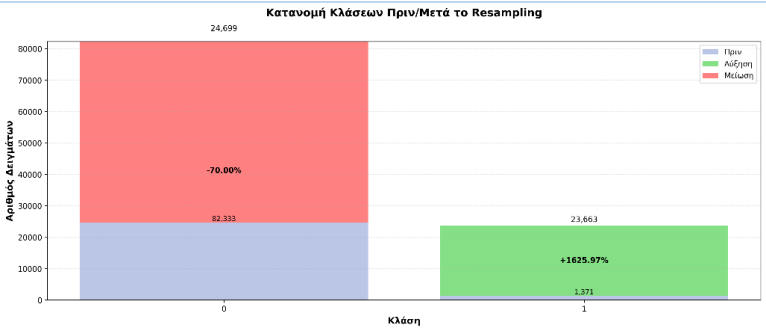
Η κλάση 0 μειώθηκε πάνω από 55%, ενώ οι κλάσεις 1, 3, 4 και 5 αυξήθηκαν (μέχρι και πάνω από 1,5 εκατομμύριο τοις εκατό για την πιο σπάνια κλάση 5). Η κλάση 2 μειώθηκε ελαφρά ώστε να έρθει πιο κοντά στις υπόλοιπες, με αποτέλεσμα ένα πολύ πιο ισορροπημένο training set για όλες τις DoS υποκατηγορίες.	

Εικόνα 47 7ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις

Η πλειοψηφική κλάση 0 μειώθηκε κατά 70%, ενώ οι κλάσεις 1, 2 και 3 ενισχύθηκαν μαζικά (από +2 092% έως πάνω από +214 000%), με αποτέλεσμα ένα σχεδόν ισορροπημένο dataset. Αυτό περιορίζει την προκατάληψη υπέρ της κλάσης 0 και βελτιώνει την ικανότητα του μοντέλου να μαθαίνει και τις τρεις κατηγορίες επιθέσεων Web.	

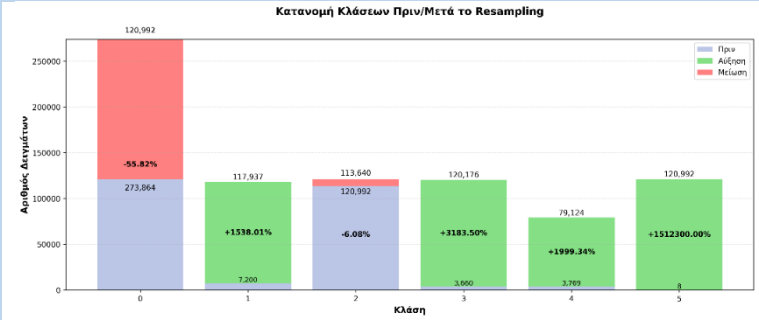
Εικόνα 48 8ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης hybrid. Κατανομή της κλάσεων πριν και μετά το resampling.

Η πλειοψηφική κλάση 0 μειώθηκε κατά 70%, ενώ η κλάση 1 (PortScan) αυξήθηκε πάνω από 1 600%, με αποτέλεσμα ένα πολύ πιο ισορροπημένο training set. Αυτή η δραστική ενίσχυση των επιθέσεων εξηγεί και την εξαιρετικά υψηλή απόδοση του μοντέλου στις μετρικές και στο confusion matrix.



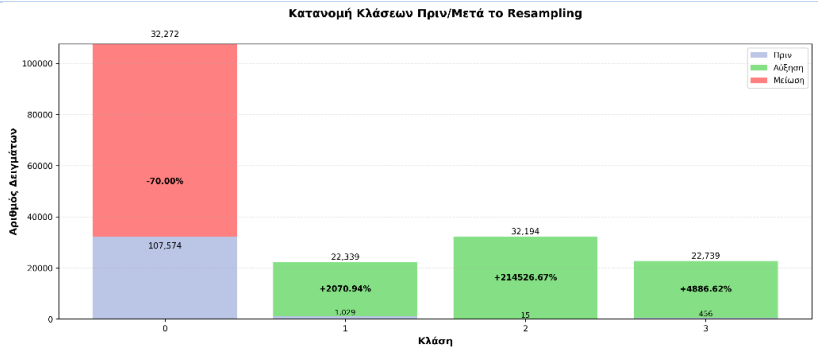
Εικόνα 49 9ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με αντικατάσταση των NaN με Μ.Ο. ανά κλάση, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling.

Η κλάση 0 μειώθηκε κατά 55.82%, ενώ όλες οι επιθετικές κλάσεις ενισχύθηκαν: η κλάση 1 περίπου +1 538%, η κλάση 3 πάνω από +3 183%, η κλάση 4 περίπου +1 999% και η εξαιρετικά σπάνια κλάση 5 πάνω από +1 512 300%, οδηγώντας σε σχεδόν ισάριθμες κλάσεις γύρω στις 110–120 χιλιάδες δείγματα.



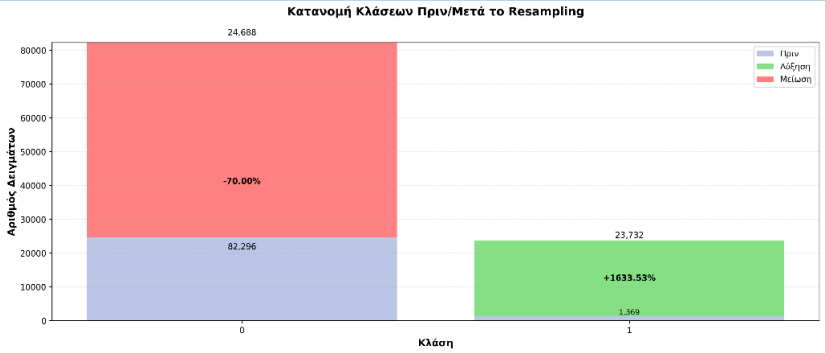
Εικόνα 50 16ος Συνδυασμός: Dataset Wednesday-workingHours_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling, για καθεμία από τις 6 κλάσεις.

Η κλάση 0 μειώθηκε κατά 70% (από 107 574 σε 32 272 δείγματα), ενώ οι επιθετικές κλάσεις 1, 2 και 3 αυξήθηκαν δραματικά: περίπου +2 071% για την κλάση 1, πάνω από +214 000% για την εξαιρετικά σπάνια κλάση 2 και περίπου +4 887% για την κλάση 3, οδηγώντας σε πιο ισορροπημένο training set.



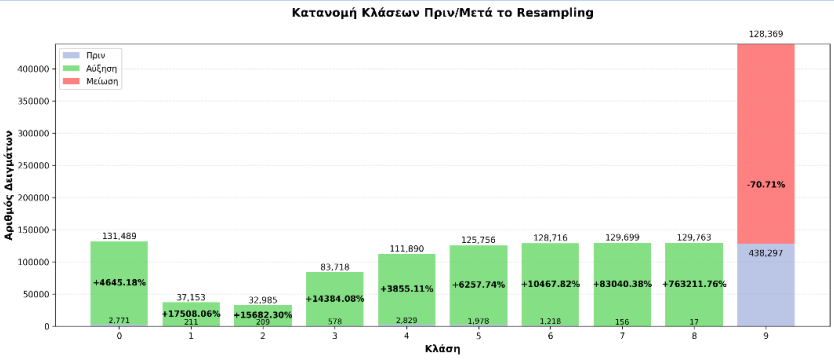
Εικόνα 51 17ος Συνδυασμός: Dataset Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Hybrid. Κατανομή της κλάσεων πριν και μετά το resampling.

Η κανονική κλάση 0 μειώνεται κατά 70% (από 82 296 σε 24 688 δείγματα), ενώ η κλάση PortScan 1 αυξάνεται περίπου κατά +1 633.53% (από 1 369 σε 23 732 δείγματα), οδηγώντας σε πολύ πιο ισορροπημένο training set.



Εικόνα 52 18ος Συνδυασμός: Dataset Friday-WorkingHours-Afternoon-PortScan_cleaned.csv με διαγραφή των NaN, μέθοδος εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling.

Παρουσιάζεται πολύ μεγάλη αύξηση σε όλες τις επιθετικές κλάσεις (π.χ. κλάση 1 +17 508.06%, κλάση 2 +15 682.30%, κλάση 8 +763 211.76%), ενώ η κανονική κλάση 9 μειώνεται κατά 70.71%, οδηγώντας σε πολύ πιο ισορροπημένη εκπαίδευση που ενισχύει τις σπάνιες επιθέσεις.



Εικόνα 53 21ος Συνδυασμός: Dataset UNSW-NB15_1.csv με μέθοδο εξισορρόπησης Hybrid. Κατανομή των κλάσεων πριν και μετά το resampling.

5.3. Βελτιστοποίηση και εκπαίδευση

Αφού ολοκληρωθεί ο διαχωρισμός των δεδομένων και η εξισορρόπηση των κλάσεων, προκύπτει το training set έτοιμο για την εκπαίδευση του ταξινομητή. Αυτό το στάδιο αποτελεί ένα από τα πιο σημαντικά σημεία ολόκληρης της διαδικασίας, καθώς οι επιλογές που γίνονται καθορίζουν άμεσα την ικανότητα του μοντέλου να γενικεύει σε αγνώστους τύπους κίνησης και να ανιχνεύει επιθέσεις με αξιοπιστία. Όπως αναφέρεται στη σχετική βιβλιογραφία, η εφαρμογή συστηματικής βελτιστοποίησης υπέρ παραμέτρων μέσω grid search επηρεάζει σημαντικά την τελική απόδοση του μοντέλου [29], ενώ η χρήση του GridSearchCV με cross-validation αποδίδει τα επιθυμητά αποτελέσματα για την εύρεση βέλτιστων ρυθμίσεων σε κάθε μοντέλο που αξιολογεί το CICIDS2017 [36].

Στην εφαρμογή μας το στάδιο αυτό περιλαμβάνει δύο διαδικασίες. Η πρώτη αφορά τον καθορισμό της κατάλληλης συνάρτησης κόστους (objective function, αντικειμενική συνάρτηση) ανάλογα με τον τύπο του προβλήματος ταξινόμησης, αν είναι δηλαδή δυαδικό

ή πολυκλασικό. Η δεύτερη διαδικασία είναι η αυτοματοποιημένη αναζήτηση των βέλτιστων υπερπαραμέτρων του μοντέλου. Για τη διαδικασία αυτή επιλέχθηκε η τεχνική RandomizedSearchCV με στόχο τη μείωση του υπολογιστικού κόστους, χωρίς σημαντική απώλεια στην ποιότητα των αποτελεσμάτων. Η επιλογή αυτή τεκμηριώνεται επίσης από την βιβλιογραφία όπου η RandomizedSearchCV χρησιμοποιείται για τη βελτιστοποίηση των υπέρ παραμέτρων σε μοντέλα IDS, μεταξύ των οποίων και το XGBoost. Επιπλέον, όπως αναφέρεται, το grid search εξετάζει όλους τους συνδυασμούς και είναι πιο απαιτητικό σε χρόνο και πόρους, σε αντίθεση με το random search που δοκιμάζει τυχαίους συνδυασμούς και επιστρέφει την καλύτερη ρύθμιση μετά από συγκεκριμένες επαναλήψεις [41]. Και οι δύο διαδικασίες εκτελούνται σε ξεχωριστό νήμα εκτέλεσης διασφαλίζοντας με αυτό τον τρόπο ότι το γραφικό περιβάλλον παραμένει λειτουργικό καθ' όλη τη διάρκεια των υπολογισμών. Τα βήματα αυτά εμφανίζονται σε πραγματικό χρόνο στο παράθυρο προόδου της εφαρμογής, επιτρέποντας στον χρήστη να παρακολουθεί την εξελικτική πορεία. Οι δύο αυτές διαδικασίες αναλύονται διεξοδικά στις ακόλουθες υποενότητες.

5.3.1. Καθορισμός αντικειμενικής συνάρτησης: Διάκριση μεταξύ δυαδικής και πολυκλασικής ταξινόμησης

Στο πρώτο στάδιο της εκπαίδευσης καθορίζεται η κατάλληλη αντικειμενική συνάρτηση του XGBoost, ανάλογα με τον αριθμό των κλάσεων που υπάρχουν στο training set. Η επιλογή αυτή είναι απαραίτητη, καθώς το μοντέλο πρέπει να γνωρίζει αν το πρόβλημα που καλείται να επιλύσει είναι δυαδικής ή πολυκλασικής ταξινόμησης. Στα συστήματα ανίχνευσης εισβολών η διάκριση αυτή είναι ιδιαίτερα σημαντική, καθώς ένα IDS μπορεί είτε να ταξινομεί την κίνηση σε δύο βασικές κατηγορίες, όπως κανονική και κακόβουλη, είτε να αναγνωρίζει περισσότερους τύπους επιθέσεων. Αντίστοιχα, στη βιβλιογραφία αξιολογούνται συστήματα ανίχνευσης εισβολών βασισμένα σε ML περιβάλλοντα με διαφορετικές κατηγορίες επιθέσεων, γεγονός που δείχνει ότι η επιλογή της κατάλληλης συνάρτησης στόχου πρέπει να συνδέεται με τη μορφή των ετικετών και το είδος της ταξινόμησης που πραγματοποιείται [29], [36].

Στην υλοποίηση της εφαρμογής, ο αριθμός των κλάσεων υπολογίζεται αυτόματα από τις μοναδικές τιμές της στήλης-στόχου του training set. Όταν εντοπίζονται δύο κλάσεις, το πρόβλημα θεωρείται δυαδικό και το XGBoost μοντέλο αρχικοποιείται με δυαδική αντικειμενική συνάρτηση η οποία αποτελεί καθιερωμένη επιλογή για τέτοιου είδους προβλήματα [22]. Η δυαδική συνάρτηση είναι κατάλληλη για περιπτώσεις όπου το μοντέλο

πρέπει να αποφασίσει αν μια εγγραφή αντιστοιχεί σε κανονική ή σε κακόβουλη κίνηση. Ως μετρική παρακολούθησης κατά την εκπαίδευση χρησιμοποιείται η logloss, η οποία αξιολογεί πόσο κοντά βρίσκονται οι προβλεπόμενες πιθανότητες στις πραγματικές ετικέτες, κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

Αντίθετα όταν οι κλάσεις είναι περισσότερες από δύο, το πρόβλημα αντιμετωπίζεται ως πολυκλασική ταξινόμηση και χρησιμοποιείται η πολυκλασική συνάρτηση. Η επιλογή αυτή επιτρέπει στο XGBoost μοντέλο να επιστρέφει, για κάθε διαθέσιμη κατηγορία, πιθανότητες έτσι ώστε στη συνέχεια να επιλεχθεί η κλάση με τη μεγαλύτερη. Αυτό είναι χρήσιμο σε IDS σενάρια όπου δεν αρκεί να εντοπιστεί μόνο αν μια ροή είναι κακόβουλη, αλλά χρειάζεται να προσδιοριστεί και το είδος της επίθεσης. Για παράδειγμα το UNSW-NB15, το οποίο χρησιμοποιείται συχνά στη βιβλιογραφία, είναι σύνολο δεδομένων πολυκλασικής ταξινόμησης και οι κακόβουλες εγγραφές του ανήκουν σε πολλαπλές κατηγορίες επιθέσεων όπως Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode και Worms, κάτι που καθιστά απαραίτητη την υποστήριξη πολυκλασικής ταξινόμησης [41].

Προγραμματιστικά, ο καθορισμός της συνάρτησης και η ταξινόμηση υλοποιήθηκαν ως ακολούθως:

```
# Υπολογισμός μοναδικών κλάσεων που υπάρχουν στο y_train
num_classes = len(np.unique(y_train))
if num_classes > 2: # Αν υπάρχουν περισσότερες από 2 κλάσεις
      # Πολυκλασική αντικειμενική συνάρτηση
      objective = 'multi:softprob'
      # Πολυκλασική log loss (μετρική αξιολόγησης)
      eval_metric = 'mlogloss'
      # Καταγραφή μηνύματος στο log
      self.log(f"Εντοπίστηκαν {num_classes} κλάσεις
                        -> Multiclass ταξινόμηση.")
else: # Αν υπάρχουν λιγότερες από 2 κλάσεις
      objective = 'binary:logistic' # Δυαδική αντικειμενική συνάρτηση
      eval_metric = 'logloss' # Δυαδική log loss
      # Καταγραφή μηνύματος στο log
```

```
        self.log("Εντοπίστηκαν 2 κλάσεις -> Binary ταξινόμηση.")


# Δημιουργία μοντέλου XGBoost classifier
model = XGBClassifier(
        # Objective ανάλογα με τον αριθμό κλάσεων
        objective=objective,
        eval_metric=eval_metric, # Μετρική αξιολόγησης
        # Αριθμός κλάσεων αν είναι πολυκλασική αλλιώς τίποτα
        num_class=num_classes if num_classes > 2 else None,
        random_state=42, # Σταθερό random_state για επαναληψιμότητα
        # Απενεργοποίηση του παλιού label encoder του XGBoost
        use_label_encoder=False
)
```

Έτσι η ίδια διαδικασία εκπαίδευσης στην εφαρμογή μπορεί να προσαρμόζεται αυτόματα τόσο σε δυαδικά όσο και σε πολυκλασικά σύνολα δεδομένων χωρίς να απαιτείται χειροκίνητη αλλαγή στον κώδικα από τον χρήστη. Αυτή η επιλογή αυξάνει την ευελιξία της εφαρμογής, καθώς επιτρέπει την εκπαίδευση του μοντέλου σε διαφορετικά σύνολα δεδομένων και διαφορετικά σενάρια ανίχνευσης εισβολών. Επιπλέον, όπως αναφέρεται και στη σχετική βιβλιογραφία, η χρήση του XGBoost ως βασικού ταξινομητή παρουσιάζει υψηλή απόδοση σε εφαρμογές ανίχνευσης εισβολών και κυβερνοεπιθέσεων [29], [41].

5.3.2. RandomizedSearchCV: Γιατί επιλέχθηκε αντί του GridSearchCV - Αναφορά υπέρ παραμέτρων

Μετά τον καθορισμό της αντικειμενικής συνάρτησης, το επόμενο βήμα είναι η εύρεση των βέλτιστων υπέρ παραμέτρων του μοντέλου. Η βελτιστοποίηση υπέρ παραμέτρων αποτελεί σημαντικό βήμα στην εκπαίδευση μοντέλων μηχανικής μάθησης, καθώς επηρεάζει άμεσα την τελική τους απόδοση. Για την υλοποίηση της εφαρμογής επιλέχθηκε η τεχνική RandomizedSearchCV (έναντι της εξαντλητικής αναζήτησης GridSearchCV), προκειμένου να επιτευχθεί η μείωση του υπολογιστικού κόστους και η αποδοτική εύρεση κατάλληλων ρυθμίσεων. Η επιλογή αυτή συνάδει με τη βιβλιογραφία, όπου χρησιμοποιείται η RandomizedSearchCV για την βελτιστοποίηση των υπέρ παραμέτρων (hyperparameter tuning) σε IDS μοντέλα μεταξύ των οποίων και το XGBoost (οι βέλτιστες τιμές που

αναφέρονται για το XGBoost είναι n_estimators=200, max_depth=3 και learning_rate=0.1) [41]. Παράλληλα, αναφέρεται ότι η τεχνική αναζήτησης σε πλέγμα, στην οποία βασίζεται η GridSearchCV, εξετάζει όλους τους συνδυασμούς υπέρ παραμέτρων και είναι πιο απαιτητική σε χρόνο και πόρους. Αντίθετα, η τεχνική τυχαίας αναζήτησης, στην οποία βασίζεται η RandomizedSearchCV, δοκιμάζει τυχαίους συνδυασμούς και επιστρέφει την καλύτερη ρύθμιση μετά από συγκεκριμένες επαναλήψεις [41].

Στην παρούσα υλοποίηση η RandomizedSearchCV εκτελείται με συγκεκριμένη παραμετροποίηση των ορισμάτων της. Η μεταβλητή n_iter=10 οποία ορίζει τον αριθμό των τυχαίων συνδυασμών υπέρ παραμέτρων που θα δοκιμαστούν, η cv=5 το οποίο καθορίζει τον αριθμό των folds της cross-validation και το κριτήριο βελτιστοποίησης το f1_weighted. Η τιμή της παραμέτρου n_iter επιλέχθηκε ώστε να περιοριστεί ο αριθμός των δοκιμών και να διατηρηθεί η διαδικασία εκπαίδευσης σε αποδεκτό χρόνο, κάτι που είναι σημαντικό καθώς η εφαρμογή λειτουργεί μέσω γραφικού περιβάλλοντος και ο χρήστης παρακολουθεί τη διαδικασία σε πραγματικό χρόνο. Η χρήση 5-fold cross-validation επιλέχθηκε ώστε κάθε συνδυασμός υπέρ παραμέτρων να αξιολογείται σε περισσότερους από έναν διαχωρισμούς του training set μειώνοντας έτσι την εξάρτηση από έναν μόνο τυχαίο διαχωρισμό. Αντίστοιχες προσεγγίσεις συναντώνται στη βιβλιογραφία, όπου εφαρμόζεται 5-fold cross-validation στη διαδικασία επιλογής χαρακτηριστικών [29], ενώ αλλού χρησιμοποιείται stratified 5-fold cross-validation για την αξιολόγηση της σταθερότητας και της γενίκευσης CNN+XGBoost μοντέλου [15].

```
# Τυχαία αναζήτηση υπερ παραμέτρων με cross-validation
search = RandomizedSearchCV(
      estimator=model, # Βασικό μοντέλο αναζήτησης
      param_distributions=param_dist, # Σύνολο τιμών που θα δοκιμαστούν
      n_iter=10, # Τυχαίος συνδυασμών υπέρ παραμέτρων
      scoring="f1_weighted", # κριτήριο βελτιστοποίησης
      cv=5, # Σύνολο folds της cross-validation
      verbose=0, # Χωρίς επιπλέον μηνύματα στην κονσόλα
      random_state=42, # Σταθερό random_state για επαναληψιμότητα
      n_jobs=-1, # Χρήση όλων των διαθέσιμων CPU cores
)
```

Στην εφαρμογή διερευνώνται έξι υπέρ παράμετροι όπου καθεμία ελέγχει διαφορετική πτυχή της συμπεριφοράς του XGBoost. Οι n_estimators, max_depth και learning_rate σχετίζονται άμεσα με τον αριθμό των δέντρων, την πολυπλοκότητα κάθε δέντρου και τον ρυθμό μάθησης του μοντέλου αντίστοιχα. Όπως αναφέραμε παραπάνω, αντίστοιχες υπερ παράμετροι αναφέρονται και στη βιβλιογραφία, όπου μέσω RandomizedSearchCV καταλήγουν σε βέλτιστες τιμές n_estimators=200, max_depth=3 και learning_rate=0.1 για τα σύνολα δεδομένων UNSW-NB15 και KDDCup 1999 [41]. Οι τιμές αυτές έχουν συμπεριληφθεί στο εύρος αναζήτησης της εφαρμογής ωστόσο διερευνώνται παράλληλα και γειτονικές τιμές, καθώς οι βέλτιστες ρυθμίσεις εξαρτώνται από το εκάστοτε σύνολο δεδομένων. Διαφορετικά σύνολα όπως το CIC-IDS2017 και το UNSW-NB15 παρουσιάζουν διαφορετικά χαρακτηριστικά και η αυτόματη αναζήτηση επιτρέπει στο μοντέλο να προσαρμόζεται δυναμικά. Έτσι η εφαρμογή δεν απαιτεί από τον χρήστη να γνωρίζει εκ των προτέρων τις βέλτιστες τιμές. Ο RandomizedSearchCV τις εντοπίζει αυτόματα δοκιμάζοντας τυχαίους συνδυασμούς από το ορισμένο εύρος και επιλέγει εκείνον που αποδίδει καλύτερα στο συγκεκριμένο σύνολο δεδομένων που έχει φορτώσει ο χρήστης. Επιπλέον, εξετάζονται οι παράμετροι subsample, colsample_bytree και gamma, οι οποίες χρησιμοποιούνται για τον έλεγχο της πολυπλοκότητας και της γενίκευσης του μοντέλου. Συγκεκριμένα, οι δύο πρώτες εισάγουν δειγματοληψία σε επίπεδο εγγραφών και χαρακτηριστικών αντίστοιχα, ενώ η gamma καθορίζει την ελάχιστη απαιτούμενη μείωση της απώλειας για κάθε διαχωρισμό κόμβου, αποτρέποντας περιττές διακλαδώσεις [42]. Μετά την ολοκλήρωση της αναζήτησης, εμφανίζεται ο καλύτερος συνδυασμός (όπως δείχνει η Εικόνα 54 και Εικόνα 55), επιλέγεται αυτόματα και χρησιμοποιείται για την τελική εκπαίδευση και αξιολόγηση στο test set.

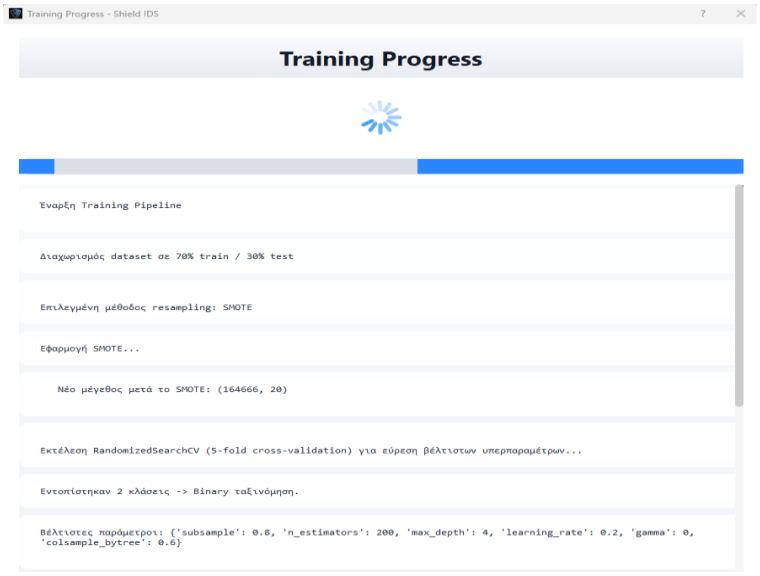
```
# Παραμετροποίηση για δοκιμές Randomized Search
param_dist = {
      "n_estimators": [100, 200, 300],  # Πλήθος δέντρων που θα εξεταστούν
      "max_depth": [3, 4, 5, 6],  # Βάθος δέντρων για έλεγχο πολυπλοκότητας
      # Ρυθμός μάθησης για κάθε δέντρο
      "learning_rate": [0.01, 0.05, 0.1, 0.2],
      # Ποσοστό δειγμάτων που θα χρησιμοποιούνται ανά δέντρο
      "subsample": [0.6, 0.8, 1.0],
```

```
        # Ποσοστό χαρακτηριστικών που θα χρησιμοποιούνται ανά δέντρο
        "colsample_bytree": [0.6, 0.8, 1.0],
        "gamma": [0, 0.1, 0.2],  # Ελάχιστη απαιτούμενη μείωση απώλειας
}

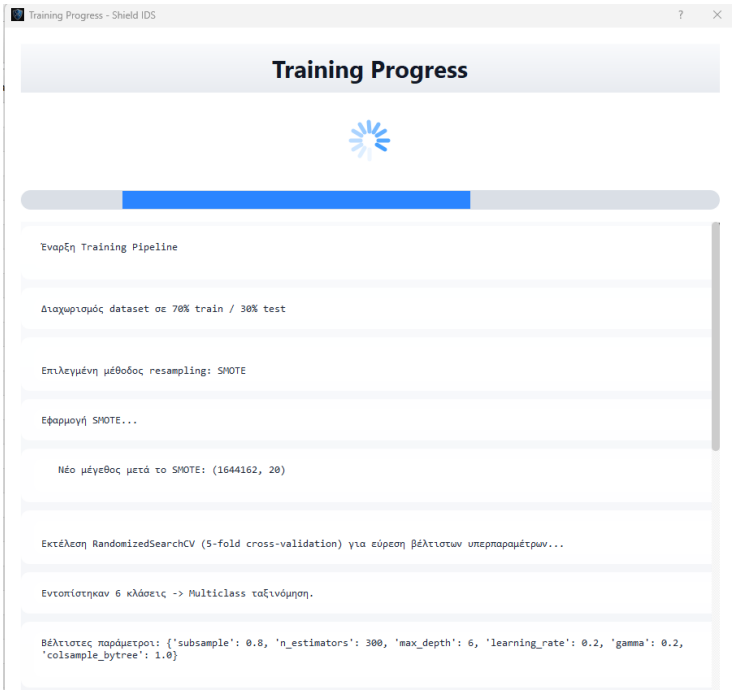
# Εκπαίδευση του randomized search πάνω στα δεδομένα εκπαίδευσης
search.fit(X_train, y_train)
# Ανάκτηση του καλύτερου μοντέλου που βρέθηκε
best_model = search.best_estimator_
# Καταγραφή των καλύτερων παραμέτρων στο log
self.log(f"Βέλτιστες παράμετροι: {search.best_params_}\n")

# Καταγραφή έναρξης εκαπίδευσης στο log
self.log("Εκπαίδευση τελικού μοντέλου...")
# Εκπαίδευση του καλύτερου μοντέλου στο training set
best_model.fit(X_train, y_train)
self.best_model = best_model  # Αποθήκευση του τελικού μοντέλου

# Καταγραφή αξιολόγησης στο log
self.log("Αξιολόγηση στο test set (Real Data)...")
# Πρόβλεψη ετικετών για το test set
y_pred = best_model.predict(X_test)
```



Εικόνα 54 Εντοπισμός δύο κλάσεων - Δυαδική ταξινόμηση και εύρεση βέλτιστων παραμέτρων.



Εικόνα 55 Εντοπισμός έξι κλάσεων - Πολυκλασική ταξινόμηση και εύρεση βέλτιστων παραμέτρων.

6. Αξιολόγηση απόδοσης

Η αξιολόγηση της απόδοσης ενός συστήματος ανίχνευσης εισβολών αποτελεί ένα πολύ σημαντικό στάδιο της ανάπτυξής του, καθώς επιτρέπει την ποσοτικοποίηση της ικανότητας του μοντέλου να διακρίνει μεταξύ φυσιολογικής και κακόβουλης κίνησης δικτύου. Όπως αναφέρεται και στη σχετική βιβλιογραφία, η αξιολόγηση IDS συστημάτων που βασίζονται σε μηχανική μάθηση πραγματοποιείται συστηματικά μέσω τυποποιημένων μετρικών, όπως η ορθότητα, η ακρίβεια, η ανάκληση και η βαθμολογία F1 (accuracy, precision, recall και F1-score), οι οποίες παρέχουν συμπληρωματική πληροφορία για τη συμπεριφορά του ταξινομητή [13], [41].

Μετά την ολοκλήρωση της εκπαίδευσης και την επιλογή των βέλτιστων υπερ παραμέτρων μέσω της τεχνικής RandomizedSearchCV το μοντέλο αξιολογείται στο ανεξάρτητο test set, δηλαδή στο 30% των δεδομένων που διατηρήθηκε αποκλειστικά για τον σκοπό αυτό και δεν χρησιμοποιήθηκε σε κανένα στάδιο εκπαίδευσης ή βελτιστοποίησης ακολουθώντας την ίδια πρακτική που εφαρμόζεται στη βιβλιογραφία [15], [29].

Η αξιολόγηση πραγματοποιείται αυτόματα με την ολοκλήρωση της εκπαίδευσης να παράγει δύο ξεχωριστά αποτελέσματα. Το πρώτο είναι ένα σύνολο αριθμητικών μετρικών απόδοσης που εμφανίζονται στο παράθυρο προόδου της εφαρμογής ενώ το δεύτερο είναι το γράφημα Confusion Matrix που οπτικοποιεί τη διανομή των σωστών και λανθασμένων προβλέψεων ανά κλάση. Στη αντίστοιχη βιβλιογραφία, φαίνεται πως η χρήση συνδυασμού αριθμητικών μετρικών και οπτικής αναπαράστασης μέσω confusion matrix αποτελεί καθιερωμένη πρακτική για την ολοκληρωμένη αξιολόγηση των μοντέλων IDS [29], [36], [41].

6.1. Μετρικές αξιολόγησης

Για την αξιολόγηση της απόδοσης του εκπαιδευμένου μοντέλου χρησιμοποιήσαμε τις τέσσερις βασικές μετρικές ορθότητα, ακρίβεια, ανάκληση και βαθμολογία F1. Όλες βασίζονται στις τέσσερις ποσότητες που προκύπτουν από τη διαδικασία ταξινόμησης οι οποίες είναι: τα αληθώς θετικά (True Positives - TP), δηλαδή τα δείγματα επίθεσης που προβλέφθηκαν σωστά ως επίθεση, τα αληθώς αρνητικά (True Negatives - TN), δηλαδή τα δείγματα κανονικής κίνησης που προβλέφθηκαν σωστά ως κανονικά, τα ψευδώς θετικά (False Positives - FP), δηλαδή τα δείγματα κανονικής κίνησης που προβλέφθηκαν

λανθασμένα ως επίθεση, και τα ψευδώς αρνητικά (False Negatives - FN), δηλαδή τα δείγματα επίθεσης που προβλέφθηκαν λανθασμένα ως κανονικά [29]. Στο πλαίσιο IDS, τα FN είναι ιδιαίτερα κρίσιμα καθώς αντιστοιχούν σε επιθέσεις που δεν εντοπίστηκαν, ενώ τα FP οδηγούν σε ψευδείς συναγερμούς που επιβαρύνουν τον αναλυτή [23], [29].

Η μετρική ορθότητας εκφράζει τη συνολική ορθότητα του μοντέλου υπολογίζοντας την αναλογία των σωστά προβλεπόμενων περιπτώσεων (τόσο αληθώς θετικών όσο και αληθώς αρνητικών) προς τον συνολικό αριθμό περιπτώσεων [26]. Ο συνολικός αριθμός σωστών προβλέψεων των ανωμαλιών ορίζεται ως:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

Η μετρική ακρίβειας εκτιμά την ακρίβεια των θετικών προβλέψεων υπολογίζοντας την αναλογία των αληθώς θετικών προς το άθροισμα των αληθώς θετικών και των ψευδώς θετικών [26]. Ορίζεται ως:

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

Η μετρική ανάκλησης αξιολογεί την ικανότητα του μοντέλου να εντοπίζει όλες τις εφαρμόσιμες περιπτώσεις υπολογίζοντας την αναλογία των αληθώς θετικών προς το άθροισμα των αληθώς θετικών και των ψευδώς αρνητικών [26]. Ορίζεται ως:

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

Η βαθμολογία F1 συνδέει την ακρίβεια και την ανάκληση σε μία μόνο μετρική υπολογίζοντας τον αρμονικό μέσο όρο ακρίβειας και ανάκλησης, παρέχοντας ένα ισορροπημένο μέτρο της απόδοσης του μοντέλου [26]. Ορίζεται ως:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

Στην εφαρμογή μας οι μετρικές ακρίβειας, ανάκλησης και F1 βαθμολογίας υπολογίζονται με τη μέθοδο weighted averaging, όπου κάθε κλάση συνεισφέρει στον τελικό υπολογισμό ανάλογα με τον αριθμό των δειγμάτων της στο test set. Τα αποτελέσματα των μετρικών αυτών εμφανίζονται αυτόματα στο παράθυρο προόδου της εφαρμογής με ακρίβεια τεσσάρων δεκαδικών ψηφίων, επιτρέποντας στον χρήστη να αξιολογήσει άμεσα την απόδοση του εκπαιδευμένου μοντέλου.

```
# Καταγραφή αξιολόγησης στο log
self.log("Αξιολόγηση στο test set (Real Data)...")
# Πρόβλεψη ετικετών για το test set
y_pred = best_model.predict(X_test)
acc = accuracy_score(y_test, y_pred) # Ορθότητα
prec = precision_score(y_test, y_pred, average="weighted") # Ακρίβεια
rec = recall_score(y_test, y_pred, average="weighted") # Ανάκληση
f1 = f1_score(y_test, y_pred, average="weighted" # Βαθμολογία F1
# Confusion Matrix για σύγκριση πραγματικών και προβλεπόμενων κλάσεων
            cm = confusion_matrix(y_test, y_pred)
# Αναλυτικό report των αποτελεσμάτων με ακρίβεια 4 δεκαδικών
report = f"""
      Accuracy: {acc:.4f} # Ορθότητα
      Precision: {prec:.4f} # Ακρίβεια
      Recall: {rec:.4f} # Ανάκληση
      F1-score: {f1:.4f} # Βαθμολογία F1
      Confusion Matrix: # Επικεφαλίδα πίνακα
      {cm} # Εισαγωγή confusion matrix στο report
"""
```

Επίσης παράγεται αυτόματα ο πίνακας σύγχυσης (confusion matrix) μέσω της βιβλιοθήκη seaborn και με έγχρωμο heatmap, όπου κάθε κελί εμφανίζει αριθμητικά τον αριθμό των προβλέψεων ανά κατηγορία, επιτρέποντας την άμεσα οπτική αναγνώριση τόσο των σωστών ταξινομήσεων (κύρια διαγώνιος) όσο και των σφαλμάτων ανά κλάση. Το γράφημα αποθηκεύεται αυτόματα σε μορφή υψηλής ανάλυσης (dpi=300) και ο χρήστης καλείται να επιλέξει τον φάκελο αποθήκευσης μέσω του γραφικού περιβάλλοντος της εφαρμογής.

```
# Καταγραφή μηνύματος ότι θα παραχθεί το γράφημα
self.log("Δημιουργία γραφήματος Confusion Matrix...")
matplotlib.use("Agg") # Ορισμός (χωρίς GUI)
plt.figure(figsize=(10, 8)) # Νέο γράφημα μεγέθους 10*8
sns.heatmap( # Σχεδίαση confusion matrix ως heatmap
```

```
        cm, # Τα δεδομένα του heatmap είναι ο πίνακας confusion matrix
        annot=True, # Εμφάνιση των τιμών εντός των κελιών
        fmt="d", # Μορφοποίηση των τιμών ως ακέραιοι
        cmap="Greens", # Πράσινη χρωματική κλίμακα
        cbar=True, # Color bar δίπλα στο γράφημα
        annot_kws={"size": 10}, # Μέγεθος γραμματοσειράς για τα annotations
)
# Τίτλος γραφήματος με έντονη γραφή
plt.title("Confusion Matrix", fontsize=14, weight="bold")
# Ετικέτα άξονα x για τις προβλεπόμενες κλάσεις
plt.xlabel("Predicted", fontsize=12)
# Ετικέτα άξονα y για τις πραγματικές κλάσεις
plt.ylabel("True", fontsize=12)
# Προσαρμογή περιθώριων (για να μην κόβονται στοιχεία)
plt.tight_layout()
plt.savefig( # Αποθήκευση confusion matrix ως εικόνα
        # Όνομα αρχείου, υψηλή ανάλυση και σωστά περιθώρια
        "confusion_matrix.png", dpi=300, bbox_inches="tight",
                pad_inches=0.8
)
plt.close() # Κλείσιμο εικόνας (και απελευθέρωση μνήμη)
```

6.2. Απεικόνιση αποτελεσμάτων

Στην ενότητα αυτή παρουσιάζονται αναλυτικά τα αποτελέσματα αξιολόγησης για τους 21 συνδυασμούς προεπεξεργασίαςς και εκπαίδευσης όπως περιγράφει ο Πίνακας 5. Για κάθε συνδυασμό παρατίθενται οι τέσσερις βασικές μετρικές απόδοσης (ορθότητας, ακρίβειας, ανάκλησης και βαθμολογίας F1) καθώς ο αντίστοιχος Πίνακας σύγχυσης, το οποίο επιτρέπει την οπτική αξιολόγηση της ταξινομητικής συμπεριφοράς του μοντέλου ανά κλάση. Αρχικά παρουσιάζεται ο συγκεντρωτικός Πίνακας 6 με τις μετρικές όλων των συνδυασμών, ώστε να είναι εφικτή η άμεση σύγκριση μεταξύ τους, και στη συνέχεια ακολουθεί η αναλυτική παρουσίαση κάθε συνδυασμού ξεχωριστά.

Συνδυασμός	Σύνολο δεδομένων	Ορθότητα	Ακρίβεια	Ανάκληση	Βαθμολογία F1
1	Wednesday-workingHours_cleaned.csv	0.9819	0.9878	0.9819	0.9837
2	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	0.9987	0.9988	0.9987	0.9987
3	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	0.9943	0.9950	0.9943	0.9946
4	Wednesday-workingHours_cleaned.csv	0.9936	0.9936	0.9936	0.9936
5	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	0.9942	0.9951	0.9942	0.9946
6	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	0.9984	0.9985	0.9984	0.9984
7	Wednesday-workingHours_cleaned.csv	0.9786	0.9799	0.9786	0.9784
8	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	0.9944	0.9952	0.9944	0.9947
9	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	0.9997	0.9997	0.9997	0.9997
10	Wednesday-workingHours_cleaned.csv	0.9818	0.9877	0.9818	0.9836
11	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	0.9948	0.9952	0.9948	0.9950

12	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	0.9981	0.9983	0.9981	0.9981
13	Wednesday-workingHours_cleaned.csv	0.9933	0.9933	0.9933	0.9933
14	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	0.9944	0.9951	0.9944	0.9947
15	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	0.9989	0.9989	0.9989	0.9989
16	Wednesday-workingHours_cleaned.csv	0.9796	0.9810	0.9796	0.9794
17	Thursday-WorkingHours-Morning-WebAttacks_cleaned.csv	0.9945	0.9953	0.9945	0.9948
18	Friday-WorkingHours-Afternoon-PortScan_cleaned.csv	0.9996	0.9996	0.9996	0.9996
19	UNSW-NB15_1.csv	0.9915	0.9936	0.9915	0.9923
20	UNSW-NB15_1.csv	0.9905	0.9933	0.9905	0.9916
21	UNSW-NB15_1.csv	0.9902	0.9931	0.9902	0.9912

Πίνακας 6 Συγκεντρωτικά αποτελέσματα μετρικών για κάθε συνδυασμό.

Στο πίνακα παρατηρούμε ότι η απόδοση του μοντέλου είναι εξαιρετικά υψηλή σε όλους τους συνδυασμούς, με τις τιμές των τεσσάρων μετρικών να κυμαίνονται μεταξύ 0.9784 και 0.9997 το οποίο επιβεβαιώνει την αποτελεσματικότητα της επιλεγμένης μεθοδολογίας εκπαίδευσης ανεξάρτητα από το σύνολο δεδομένων και τη μέθοδο εξισορρόπησης που χρησιμοποιήθηκε.

Εξετάζοντας τα αποτελέσματα ανά σύνολο δεδομένων, παρατηρείται ότι οι συνδυασμοί που αφορούν το Friday-WorkingHours-Afternoon-PortScan (συνδυασμοί 2, 6, 9, 12, 15, 18 του παραπάνω πίνακα) επιτυγχάνουν τις υψηλότερες τιμές με τη βαθμολογία F1 να κυμαίνεται από 0.9946 έως 0.9997. Αυτό σημαίνει ότι ο XGBoost αναγνωρίζει με εξαιρετική ακρίβεια

τα χαρακτηριστικά της επίθεσης Portscan. Η υψηλότερη βαθμολογία F1 επιτεύχθηκε στον συνδυασμό 9 με τιμή 0.9997, υποδεικνύοντας σχεδόν τέλεια ταξινόμηση.

Οι συνδυασμοί που προκύπτουν για το Thursday-WorkingHours-Morning-WebAttacks (συνδυασμοί 3, 5, 8, 11, 14, 17) παρουσιάζουν και αυτοί πολύ υψηλή και σταθερή απόδοση, με τη βαθμολογία F1 να κυμαίνεται από 0.9946 έως 0.9950. Αυτό δείχνει ότι το μοντέλο γενικεύει αποτελεσματικά στις τρεις κατηγορίες web επιθέσεων ανεξάρτητα από τη μέθοδο εξισορρόπησης. Η υψηλότερη βαθμολογία F1 επιτεύχθηκε στον συνδυασμό 11 με τιμή 0.9950.

Οι συνδυασμοί για το Wednesday-workingHours (συνδυασμοί 1, 4, 7, 10, 13, 16) εμφανίζουν ελαφρώς χαμηλότερες τιμές σε σχέση με τα υπόλοιπα σύνολα δεδομένων, με τη βαθμολογία F1 να κυμαίνεται από 0.9784 έως 0.9936. Αυτό είναι αναμενόμενο καθώς αυτό το σύνολο δεδομένων περιλαμβάνει έξι κλάσεις (DoS επιθέσεις και κανονική κίνηση) το οποίο καθιστά το πρόβλημα πολυκλασικής ταξινόμησης πιο απαιτητικό. Οι χαμηλότερες τιμές παρατηρούνται στους συνδυασμούς 7 και 16 (Hybrid), με βαθμολογία F1 0.9784 και 0.9794 αντίστοιχα.

Για το UNSW-NB15 (συνδυασμοί 19, 20, 21), το μοντέλο επιτυγχάνει σταθερά υψηλή απόδοση με τη βαθμολογία F1 μεταξύ 0.9912 και 0.9923 παρά την σημαντική πολυπλοκότητα του συνόλου δεδομένων που περιλαμβάνει 10 κλάσεις επιθέσεων. Αξίζει να αναφέρουμε ότι η μέθοδος SMOTE (συνδυασμός 19) επιτυγχάνει ελαφρώς υψηλότερη βαθμολογία F1 (0.9923) σε σχέση με τις μεθόδους SMOTE-ENN (0.9916) και Hybrid (0.9912), το οποίο μας δείχνει ότι για το συγκεκριμένο σύνολο δεδομένων η απλούστερη μέθοδος εξισορρόπησης αποδίδει καλύτερα.

Παρατηρούμε ότι δεν υπάρχει μία μέθοδος που να υπερτερεί σε όλα τα σύνολα δεδομένων. Η μέθοδος Hybrid επιτυγχάνει την υψηλότερη απόδοση στον συνδυασμό 9 με βαθμολογία F1 = 0.9997, ενώ στο σύνολο Wednesday εμφανίζει τις χαμηλότερες τιμές. Αντίθετα, η SMOTE-ENN παρουσιάζει σταθερά καλή απόδοση σε όλα τα σύνολα δεδομένων, αποτελώντας μια ισορροπημένη επιλογή.

Βάσει των παραπάνω αποτελεσμάτων προκύπτουν οι βέλτιστοι συνδυασμοί για κάθε σύνολο δεδομένων. Συγκεκριμένα για το Wednesday-workingHours_cleaned ο συνδυασμός 4 αναδεικνύεται ως ο βέλτιστος με βαθμολογία F1 0.9936 καθώς επιτυγχάνει την υψηλότερη απόδοση στο πιο απαιτητικό πολυκλασικό σενάριο DoS επιθέσεων. Στο

Thursday-WorkingHours-Morning-WebAttacks_cleaned ο συνδυασμός 2 πετυχαίνει την καλύτερη απόδοση με 0.9987 το οποίο υποδεικνύει ότι η απλούστερη μέθοδος εξισορρόπησης είναι αρκετή για την αποτελεσματική ανίχνευση web επιθέσεων. Για το σύνολο δεδομένων Friday-WorkingHours-Afternoon-PortScan_cleaned ο συνδυασμός 9 αποτελεί τον βέλτιστο συνδυασμό με βαθμολογία F1 0.9997 συνολικά, επιτυγχάνοντας σχεδόν τέλεια ταξινόμηση μεταξύ κανονικής κίνησης και PortScan επιθέσεων.

Σχετικά με το σύνολο δεδομένων UNSW-NB15 ο συνδυασμός 19 αναδεικνύεται ως ο βέλτιστος με βαθμολογία F1 0.9923, επιτυγχάνοντας την υψηλότερη απόδοση στο πιο σύνθετο σύνολο δεδομένων με τις 10 κλάσεις επιθέσεων. Συνολικά, ο συνδυασμός αποτελεί τον βέλτιστο συνδυασμό εκπαίδευσης από όλους τους 21 που εξετάστηκαν, ενώ ο συνδυασμός 4 αναδεικνύεται ως ο βέλτιστος για το πιο απαιτητικό πολυκλασικό σενάριο.

Στην συνέχεια παρουσιάζονται αναλυτικά οι πίνακες σύγχυσης για τους 21 συνδυασμούς. Κάθε πίνακας απεικονίζει τη διανομή των προβλέψεων του μοντέλου στο test set, με τον κατακόρυφο άξονα να αντιστοιχεί στις πραγματικές κλάσεις και τον οριζόντιο στις προβλεπόμενες. Όπως αναφέραμε και παραπάνω, οι τιμές στην κύρια διαγώνιο αντιπροσωπεύουν τις σωστές ταξινομήσεις, ενώ οι τιμές εκτός διαγώνιου αποτυπώνουν τα σφάλματα ταξινόμησης μεταξύ κλάσεων. Όσο μεγαλύτερες είναι οι διαγώνιες τιμές και όσο μικρότερες οι εκτός διαγώνιου, τόσο υψηλότερη είναι η ικανότητα διάκρισης του μοντέλου

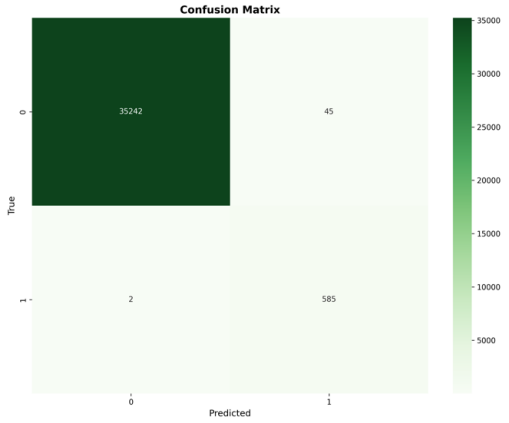
Για τον πρώτο συνδυασμό (τον οποίο παρουσιάζει η Εικόνα 56) το μοντέλο αναγνωρίζει με εξαιρετική ακρίβεια την κανονική κίνηση (BENIGN, κλάση 0) με 114.839 σωστές ταξινομήσεις, ενώ τα κυριότερα σφάλματα εντοπίζονται στη σύγχυση της κλάσης 0 με την DoS GoldenEye (κλάση 1, 2.003 δείγματα) και την DoS Hulk (κλάση 2, 550 δείγματα). Η DoS Hulk αναγνωρίζεται σωστά σε 51.311 περιπτώσεις, με μικρή σύγχυση προς την κλάση 0 (161) και την DoS GoldenEye (303). Οι κλάσεις DoS GoldenEye (3.084), DoS Slowhttptest (1.547) και DoS slowloris (1.608) ταξινομούνται με υψηλή ακρίβεια, με ελάχιστα σφάλματα. Αξίζει να αναφέρουμε την κλάση Heartbleed (κλάση 5), η οποία παρά τον εξαιρετικά μικρό αριθμό δειγμάτων (μόλις 3 σωστές προβλέψεις), ταξινομείται σωστά χωρίς καμία σύγχυση με άλλες κλάσεις.



Εικόνα 56 Πίνακας σύγχυσης για τον 1ο συνδυασμό.

Συνολικά, η πλειονότητα των σφαλμάτων εντοπίζεται μεταξύ της κανονικής κίνησης και των DoS υποκατηγοριών, κάτι που είναι αναμενόμενο λόγω της ομοιότητας ορισμένων χαρακτηριστικών τους.

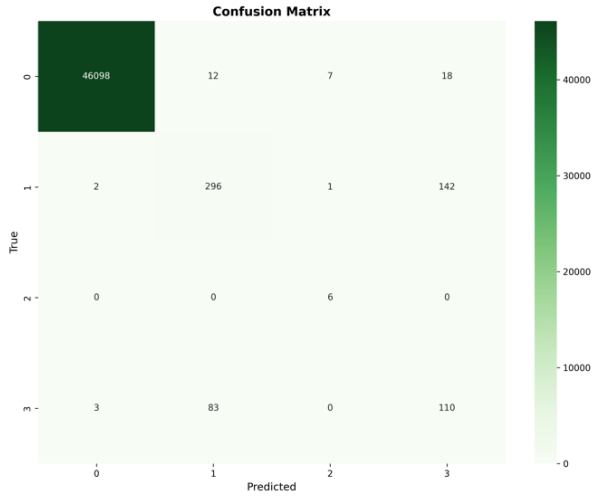
Στον δεύτερο συνδυασμό (Εικόνα 57) το μοντέλο επιτυγχάνει εξαιρετικό διαχωρισμό μεταξύ κανονικής κίνησης και PortScan επιθέσεων. Συγκεκριμένα, 35.242 BENIGN ροές ταξινομούνται σωστά, με μόλις 45 να επισημαίνονται λανθασμένα ως PortScan (ψευδώς θετικά). Από την πλευρά των επιθέσεων, 585 PortScan ροές εντοπίζονται σωστά, ενώ μόλις 2 χάνονται ως κανονική κίνηση (ψευδώς αρνητικά). Το εξαιρετικά μικρό πλήθος σφαλμάτων και στις δύο κατευθύνσεις επιβεβαιώνει ότι το μοντέλο έχει μάθει με μεγάλη σαφήνεια το προφίλ των PortScan επιθέσεων, διακρίνοντάς το αποτελεσματικά από την κανονική κίνηση.



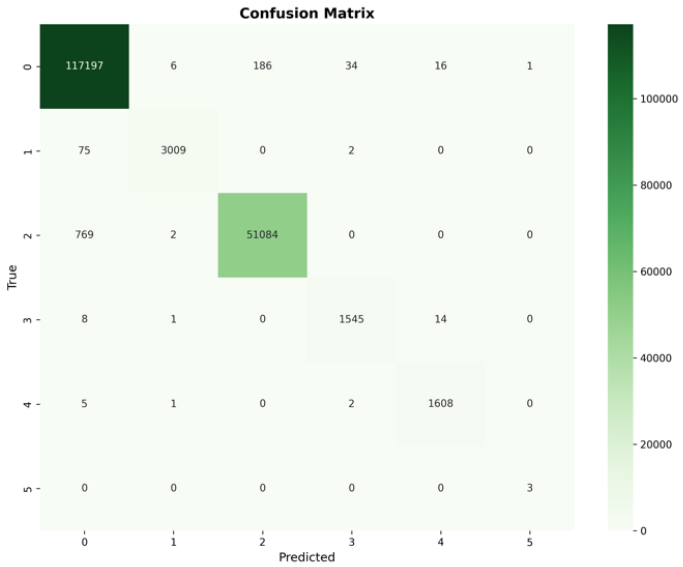
Εικόνα 57 Πίνακας σύγχυσης για τον 2ο συνδυασμό.

Στον τρίτο συνδυασμό (όπως δείχνει η Εικόνα 58) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (BENIGN) με 46.098 σωστές ταξινομήσεις και ελάχιστα σφάλματα (12 ως Brute Force, 7 ως SQL Injection και 18 ως XSS). Η κλάση Brute Force (κλάση 1) αναγνωρίζεται σωστά σε 296 περιπτώσεις, με 2 δείγματα να χάνονται ως BENIGN και 142 να ταξινομούνται λανθασμένα ως XSS, υποδεικνύοντας ομοιότητα μεταξύ των δύο αυτών κατηγοριών επίθεσης. Η κλάση SQL Injection (κλάση 2) εντοπίζεται σωστά σε 6 περιπτώσεις χωρίς κανένα ψευδώς θετικό, γεγονός αξιοσημείωτο δεδομένου του εξαιρετικά μικρού αριθμού δειγμάτων της. Η κλάση XSS (κλάση 3) αναγνωρίζεται σωστά σε 110 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση Brute Force (83 δείγματα), κάτι που είναι αναμενόμενο λόγω της ομοιότητας των χαρακτηριστικών κίνησης μεταξύ των δύο τύπων web επιθέσεων.

Στον τέταρτο συνδυασμό (Εικόνα 59) το μοντέλο αναγνωρίζει με εξαιρετική ακρίβεια την κανονική κίνηση (BENIGN) με 117.197 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με την DoS Hulk (186 δείγματα) και την DoS Slowhttptest (34 δείγματα). Η DoS GoldenEye (κλάση 1) αναγνωρίζεται σωστά σε 3.009 περιπτώσεις, με 75 δείγματα να χάνονται ως BENIGN. Η DoS Hulk (κλάση 2) εντοπίζεται σωστά σε 51.084 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση BENIGN (769 δείγματα). Οι κλάσεις DoS Slowhttptest (1.545) και DoS slowloris (1.608) ταξινομούνται με υψηλή ακρίβεια και ελάχιστα σφάλματα. Η κλάση Heartbleed (κλάση 5) παρά τον εξαιρετικά μικρό αριθμό δειγμάτων ταξινομείται σωστά και στις 3 περιπτώσεις χωρίς κανένα σφάλμα.



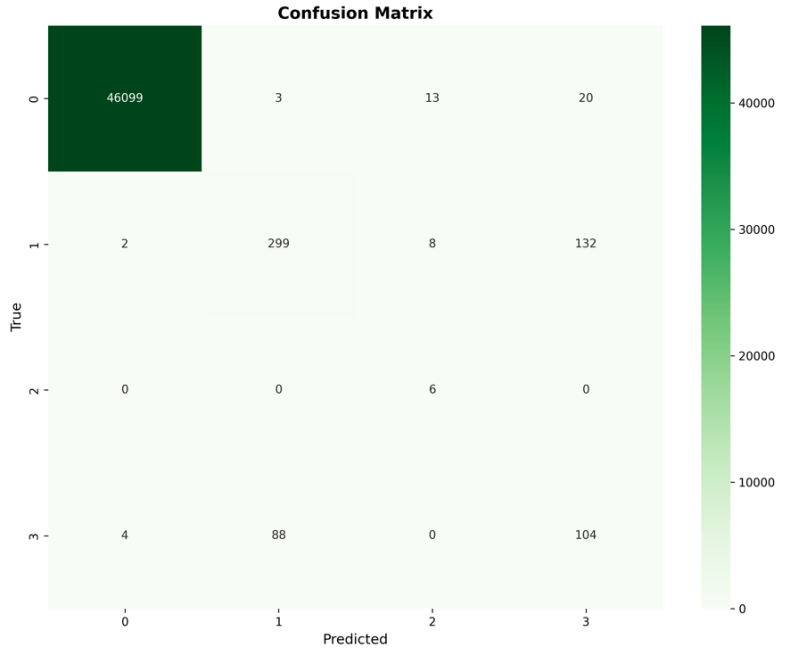
Εικόνα 58 Πίνακας σύγχυσης για τον 3ο συνδυασμό.



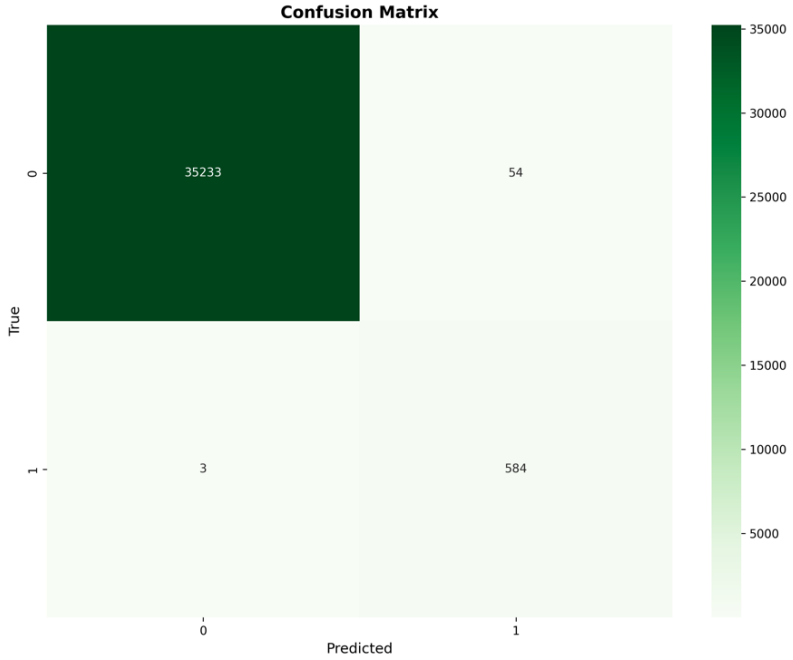
Εικόνα 59 Πίνακας σύγχυσης για τον 4ο συνδυασμό.

Στον πέμπτο συνδυασμό (παρουσιάζεται στην Εικόνα 60) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (BENIGN) με 46.099 σωστές ταξινομήσεις και ελάχιστα σφάλματα (3 ως Brute Force, 13 ως SQL Injection και 20 ως XSS). Η κλάση Brute Force (κλάση 1) αναγνωρίζεται σωστά σε 299 περιπτώσεις, με 2 δείγματα να χάνονται ως BENIGN και 132 να ταξινομούνται λανθασμένα ως XSS, επιβεβαιώνοντας την ομοιότητα των χαρακτηριστικών μεταξύ των δύο αυτών κατηγοριών, όπως παρατηρήθηκε και στον συνδυασμό 3. Η κλάση SQL Injection (κλάση 2) εντοπίζεται σωστά και στις 6 περιπτώσεις χωρίς κανένα ψευδώς θετικό, παρά τον εξαιρετικά μικρό αριθμό δειγμάτων. Η κλάση XSS (κλάση 3) αναγνωρίζεται σωστά σε 104 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται και πάλι προς την κλάση Brute Force (88 δείγματα).

Στον έκτο συνδυασμό (Εικόνα 61) το μοντέλο επιτυγχάνει εξαιρετικό διαχωρισμό μεταξύ κανονικής κίνησης και PortScan επιθέσεων. Συγκεκριμένα, 35.233 BENIGN ροές ταξινομούνται σωστά, με μόλις 54 να επισημαίνονται λανθασμένα ως PortScan (ψευδώς θετικά). Από την πλευρά των επιθέσεων, 584 PortScan ροές εντοπίζονται σωστά, ενώ μόλις 3 χάνονται ως κανονική κίνηση (ψευδώς αρνητικά). Το εξαιρετικά μικρό πλήθος σφαλμάτων και στις δύο κατευθύνσεις επιβεβαιώνει ότι το μοντέλο έχει μάθει με μεγάλη σαφήνεια το προφίλ των PortScan επιθέσεων, διακρίνοντάς το αποτελεσματικά από την κανονική κίνηση.



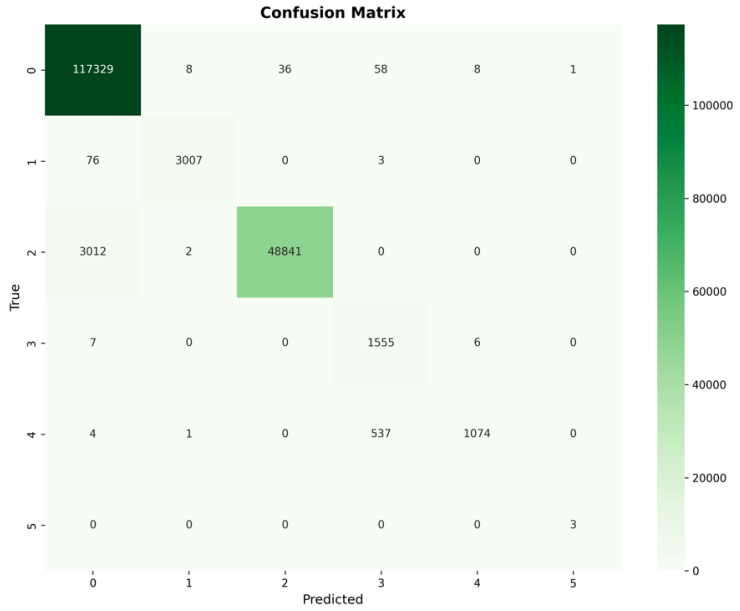
Εικόνα 60 Πίνακας σύγχυσης για τον 5ο συνδυασμό.



Εικόνα 61 Πίνακας σύγχυσης για τον 6ο συνδυασμό.

Στον έβδομο συνδυασμό (Εικόνα 62) το μοντέλο αναγνωρίζει με εξαιρετική ακρίβεια την κανονική κίνηση (BENIGN) με 117.329 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με την DoS Slowhttptest (58 δείγματα) και την DoS Hulk (36 δείγματα). Η DoS GoldenEye (κλάση 1) αναγνωρίζεται σωστά σε 3.007 περιπτώσεις, με 76 δείγματα να χάνονται ως BENIGN. Η DoS Hulk (κλάση 2) εντοπίζεται σωστά σε 48.841 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση BENIGN (3.012 δείγματα), κάτι που αποτελεί το μεγαλύτερο σφάλμα του μοντέλου στον συγκεκριμένο συνδυασμό. Η DoS Slowhttptest (κλάση 3) αναγνωρίζεται σωστά σε 1.555 περιπτώσεις με ελάχιστα σφάλματα. Αξιοσημείωτη είναι η περίπτωση της κλάσης DoS slowloris (κλάση 4), η οποία παρουσιάζει σημαντική σύγχυση με την DoS Slowhttptest (537 δείγματα), υποδεικνύοντας ομοιότητα στα χαρακτηριστικά κίνησης των δύο αυτών επιθέσεων. Τέλος, η κλάση Heartbleed (κλάση 5) ταξινομείται σωστά και στις 3 περιπτώσεις χωρίς κανένα σφάλμα.

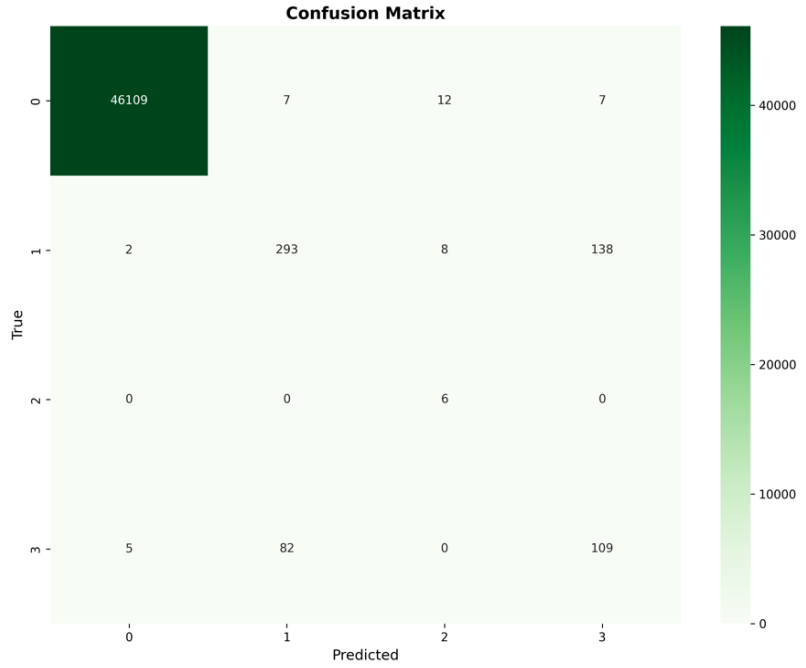
Στον όγδοο συνδυασμό (Εικόνα 63) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (BENIGN) με 46.109 σωστές ταξινομήσεις και ελάχιστα σφάλματα (7 ως Brute Force, 12 ως SQL Injection και 7 ως XSS). Η κλάση Brute Force (κλάση 1) αναγνωρίζεται σωστά σε 293 περιπτώσεις, με 2 δείγματα να χάνονται ως BENIGN και 138 να ταξινομούνται λανθασμένα ως XSS, επιβεβαιώνοντας για άλλη μια φορά την ομοιότητα των χαρακτηριστικών κίνησης μεταξύ Brute Force και XSS επιθέσεων.



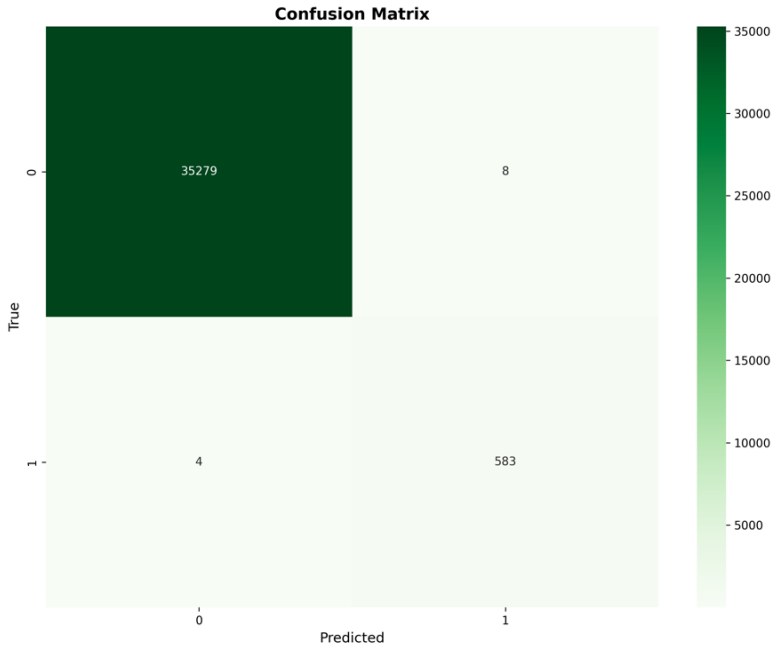
Εικόνα 62 Πίνακας σύγχυσης για τον 7ο συνδυασμό.

Η κλάση SQL Injection (κλάση 2) εντοπίζεται σωστά και στις 6 περιπτώσεις χωρίς κανένα ψευδώς θετικό, παρά τον εξαιρετικά μικρό αριθμό δειγμάτων. Η κλάση XSS (κλάση 3) αναγνωρίζεται σωστά σε 109 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση Brute Force (82 δείγματα) και 5 δείγματα να χάνονται ως BENIGN.

Στον ένατο συνδυασμό (Εικόνα 64) το μοντέλο επιτυγχάνει σχεδόν τέλειο διαχωρισμό μεταξύ κανονικής κίνησης και PortScan επιθέσεων, αποτελώντας τον βέλτιστο συνδυασμό από όλους τους 21 που εξετάστηκαν. Συγκεκριμένα, 35.279 BENIGN ροές ταξινομούνται σωστά, με μόλις 8 να επισημαίνονται λανθασμένα ως PortScan (ψευδώς θετικά). Από την πλευρά των επιθέσεων, 583 PortScan ροές εντοπίζονται σωστά, ενώ μόλις 4 χάνονται ως κανονική κίνηση (ψευδώς αρνητικά). Το συνολικό πλήθος σφαλμάτων (12 δείγματα επί συνόλου 35.874) είναι εξαιρετικά μικρό, επιβεβαιώνοντας ότι η υβριδική μέθοδος εξισορρόπησης σε συνδυασμό με την αντικατάσταση των NaN τιμών οδηγεί στην άριστη απόδοση για την ανίχνευση PortScan επιθέσεων.



Εικόνα 63 Πίνακας σύγχυσης για τον 8ο συνδυασμό.



Εικόνα 64 Πίνακας σύγχυσης για τον 9ο συνδυασμό.

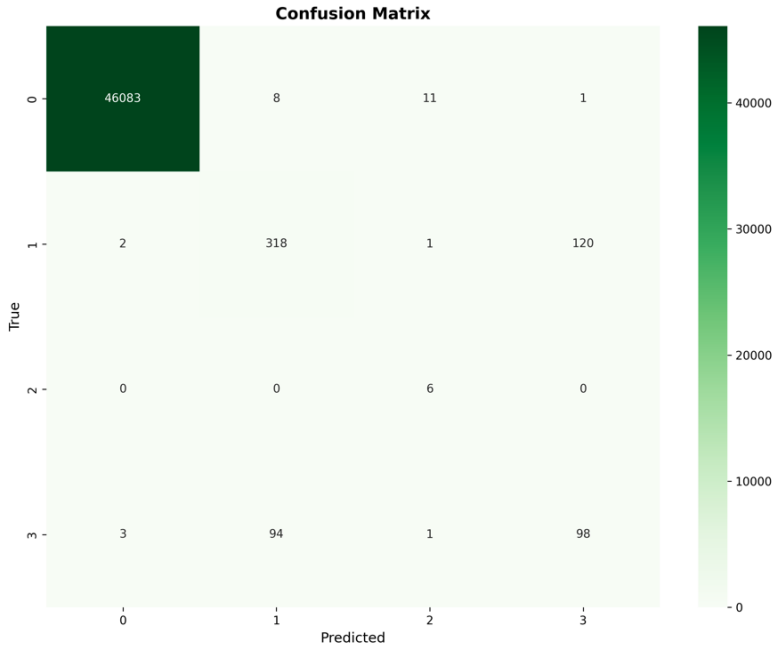
Στον δέκατο συνδυασμό (Εικόνα 65) το μοντέλο αναγνωρίζει με εξαιρετική ακρίβεια την κανονική κίνηση (BENIGN) με 114.751 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με την DoS GoldenEye (2.021 δείγματα) και την DoS Hulk (527 δείγματα). Η DoS GoldenEye (κλάση 1) αναγνωρίζεται σωστά σε 3.082 περιπτώσεις χωρίς κανένα ψευδώς αρνητικό, με μόλις 4 δείγματα να ταξινομούνται λανθασμένα ως DoS Slowhttptest. Η DoS Hulk (κλάση 2) εντοπίζεται σωστά σε 51.308 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση BENIGN (185 δείγματα) και την DoS GoldenEye (361 δείγματα). Οι κλάσεις DoS Slowhttptest (1.548) και DoS slowloris (1.604) ταξινομούνται με υψηλή ακρίβεια και ελάχιστα σφάλματα μεταξύ τους. Η κλάση Heartbleed (κλάση 5) ταξινομείται σωστά και στις 3 περιπτώσεις χωρίς κανένα σφάλμα, παρά τον εξαιρετικά μικρό αριθμό δειγμάτων της.

Στον εντέκατο συνδυασμό (Εικόνα 66) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (BENIGN) με 46.083 σωστές ταξινομήσεις και ελάχιστα σφάλματα (8 ως Brute Force, 11 ως SQL Injection και 1 ως XSS). Η κλάση Brute Force (κλάση 1) αναγνωρίζεται σωστά σε 318 περιπτώσεις, με 2 δείγματα να χάνονται ως BENIGN και 120 να ταξινομούνται λανθασμένα ως XSS, επιβεβαιώνοντας για άλλη μια φορά τη χαρακτηριστική σύγχυση μεταξύ Brute Force και XSS που παρατηρείται σταθερά σε όλους τους συνδυασμούς του Thursday dataset.



Εικόνα 65 Πίνακας σύγχυσης για τον 10ο συνδυασμό.

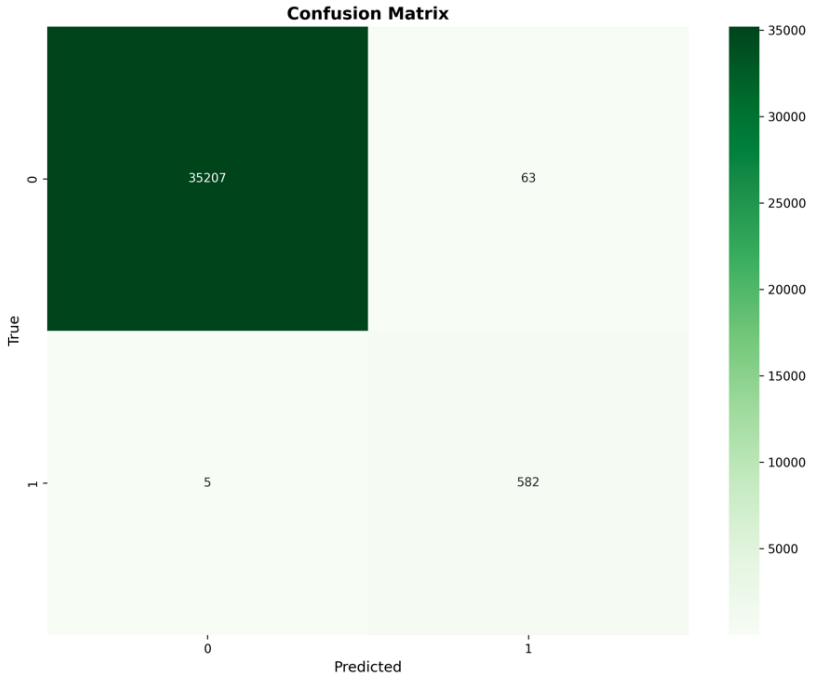
Η κλάση SQL Injection (κλάση 2) εντοπίζεται σωστά και στις 6 περιπτώσεις χωρίς κανένα ψευδώς θετικό. Η κλάση XSS (κλάση 3) αναγνωρίζεται σωστά σε 98 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση Brute Force (94 δείγματα) και 3 δείγματα να χάνονται ως BENIGN.



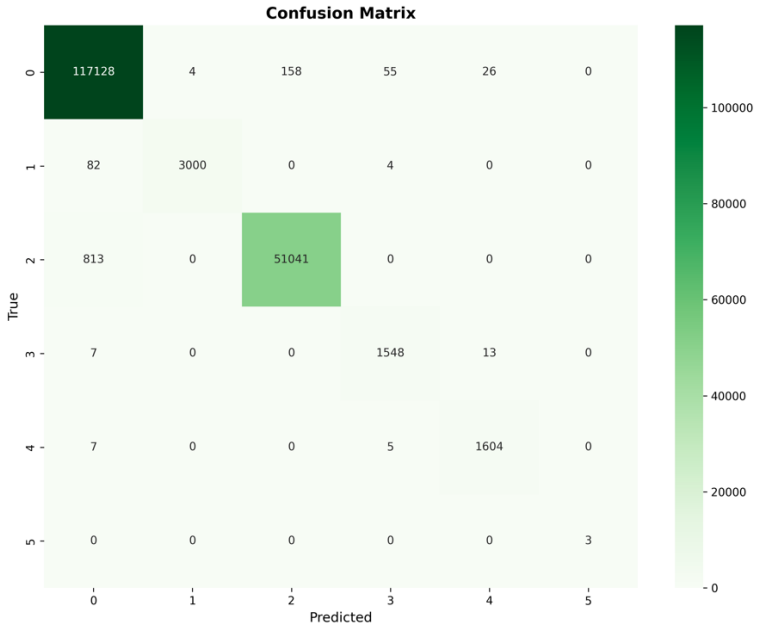
Εικόνα 66 Πίνακας σύγχυσης για τον 11ο συνδυασμό.

Στον δωδέκατο συνδυασμό (Εικόνα 67) το μοντέλο επιτυγχάνει εξαιρετικό διαχωρισμό μεταξύ κανονικής κίνησης και PortScan επιθέσεων. Συγκεκριμένα, 35.207 BENIGN ροές ταξινομούνται σωστά, με μόλις 63 να επισημαίνονται λανθασμένα ως PortScan (ψευδώς θετικά). Από την πλευρά των επιθέσεων, 582 PortScan ροές εντοπίζονται σωστά, ενώ μόλις 5 χάνονται ως κανονική κίνηση (ψευδώς αρνητικά). Το εξαιρετικά μικρό πλήθος σφαλμάτων επιβεβαιώνει ότι το μοντέλο διακρίνει με μεγάλη σαφήνεια το προφίλ των PortScan επιθέσεων από την κανονική κίνηση, ακόμα και όταν οι NaN τιμές αντιμετωπίζονται με διαγραφή αντί για αντικατάσταση.

Στον δέκατο τρίτο συνδυασμό (παρουσιάζεται στην Εικόνα 68) το μοντέλο αναγνωρίζει με εξαιρετική ακρίβεια την κανονική κίνηση (BENIGN) με 117.128 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με την DoS Hulk (158 δείγματα) και την DoS Slowhttptest (55 δείγματα). Η DoS GoldenEye (κλάση 1) αναγνωρίζεται σωστά σε 3.000 περιπτώσεις, με 82 δείγματα να χάνονται ως BENIGN και 4 ως DoS Slowhttptest.



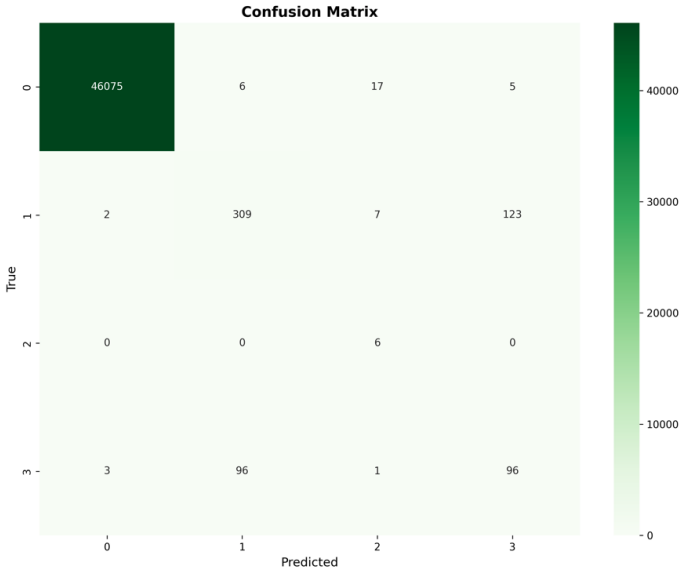
Εικόνα 67 Πίνακας σύγχυσης για τον 12ο συνδυασμό.



Εικόνα 68 Πίνακας σύγχυσης για τον 13ο συνδυασμό.

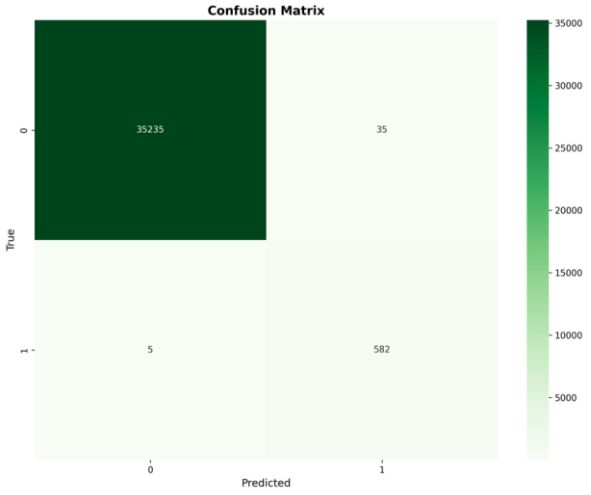
Η DoS Hulk (κλάση 2) εντοπίζεται σωστά σε 51.041 περιπτώσεις, με την κύρια σύγχυση να παρουσιάζεται αποκλειστικά προς την κλάση BENIGN (813 δείγματα), χωρίς κανένα σφάλμα προς τις υπόλοιπες. Οι κλάσεις DoS Slowhttptest (1.548) και DoS slowloris (1.604) ταξινομούνται με υψηλή ακρίβεια, με μικρή σύγχυση μεταξύ τους (13 και 5 δείγματα αντίστοιχα). Η κλάση Heartbleed (κλάση 5) ταξινομείται σωστά και στις 3 περιπτώσεις χωρίς κανένα σφάλμα.

Στον δέκατο τέταρτο συνδυασμό (όπως δείχνει η Εικόνα 69) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (BENIGN) με 46.075 σωστές ταξινομήσεις και ελάχιστα σφάλματα (6 ως Brute Force, 17 ως SQL Injection και 5 ως XSS). Η κλάση Brute Force (κλάση 1) αναγνωρίζεται σωστά σε 309 περιπτώσεις, με 2 δείγματα να χάνονται ως BENIGN, 7 ως SQL Injection και 123 να ταξινομούνται λανθασμένα ως XSS, επιβεβαιώνοντας τη χαρακτηριστική σύγχυση μεταξύ Brute Force και XSS που παρατηρείται σταθερά σε όλους τους συνδυασμούς του Thursday dataset. Η κλάση SQL Injection (κλάση 2) εντοπίζεται σωστά και στις 6 περιπτώσεις χωρίς κανένα ψευδώς θετικό. Η κλάση XSS (κλάση 3) αναγνωρίζεται σωστά σε 96 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση Brute Force (96 δείγματα) και 3 δείγματα να χάνονται ως BENIGN.



Εικόνα 69 Πίνακας σύγχυσης για τον 14ο συνδυασμό.

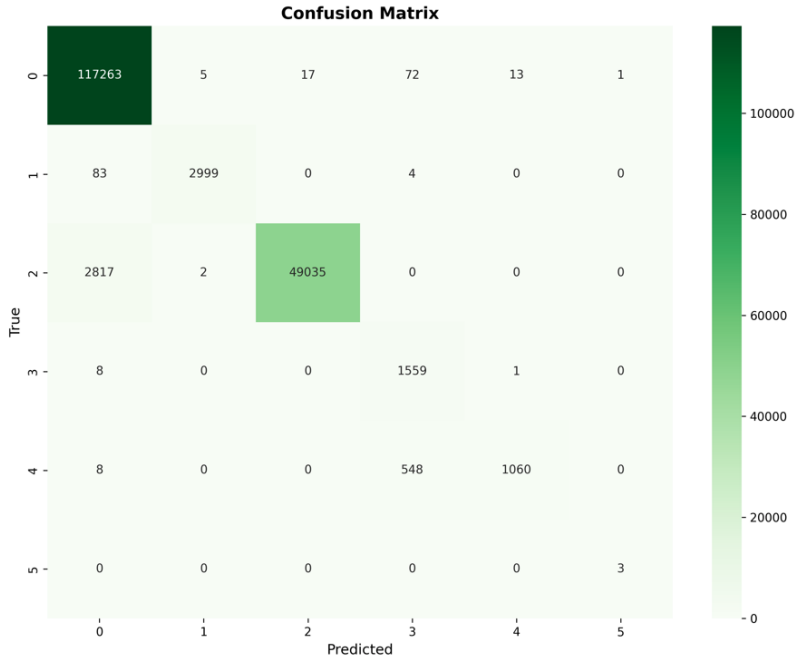
Στον δέκατο πέμπτο συνδυασμό (Εικόνα 70) το μοντέλο επιτυγχάνει εξαιρετικό διαχωρισμό μεταξύ κανονικής κίνησης και PortScan επιθέσεων. Συγκεκριμένα, 35.235 BENIGN ροές ταξινομούνται σωστά, με μόλις 35 να επισημαίνονται λανθασμένα ως PortScan (δευδώς θετικά). Από την πλευρά των επιθέσεων, 582 PortScan ροές εντοπίζονται σωστά, ενώ μόλις 5 χάνονται ως κανονική κίνηση (ψευδώς αρνητικά). Το εξαιρετικά μικρό πλήθος σφαλμάτων επιβεβαιώνει ότι το μοντέλο διακρίνει με μεγάλη σαφήνεια το προφίλ των PortScan επιθέσεων από την κανονική κίνηση, επιτυγχάνοντας υψηλή αξιοπιστία τόσο στην ανίχνευση επιθέσεων όσο και στον περιορισμό των ψευδών συναγερμών.



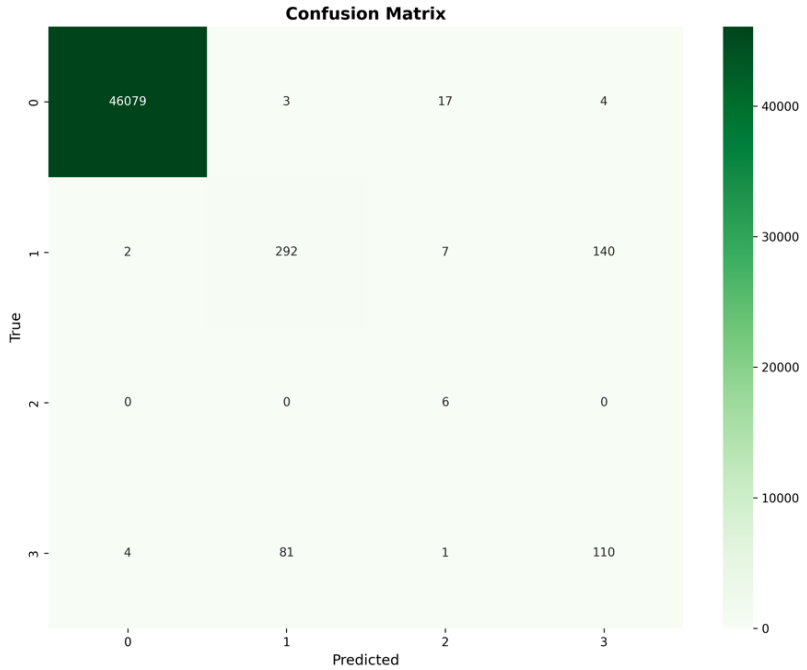
Εικόνα 70 Πίνακας σύγχυσης για τον 15ο συνδυασμό.

Στον δέκατο έκτο συνδυασμό (Εικόνα 71) το μοντέλο αναγνωρίζει με εξαιρετική ακρίβεια την κανονική κίνηση (BENIGN) με 117.263 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με την DoS Slowhttptest (72 δείγματα) και την DoS Hulk (17 δείγματα). Η DoS GoldenEye (κλάση 1) αναγνωρίζεται σωστά σε 2.999 περιπτώσεις, με 83 δείγματα να χάνονται ως BENIGN και 4 ως DoS Slowhttptest. Η DoS Hulk (κλάση 2) εντοπίζεται σωστά σε 49.035 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται αποκλειστικά προς την κλάση BENIGN (2.817 δείγματα), αποτελώντας το μεγαλύτερο σφάλμα του μοντέλου στον συγκεκριμένο συνδυασμό. Η DoS Slowhttptest (κλάση 3) αναγνωρίζεται σωστά σε 1.559 περιπτώσεις με ελάχιστα σφάλματα. Αξιοσημείωτη είναι η περίπτωση της κλάσης DoS slowloris (κλάση 4), στην οποία παρουσιάζεται σημαντική σύγχυση με την DoS Slowhttptest (548 δείγματα), υποδεικνύοντας ομοιότητα στα χαρακτηριστικά κίνησης των δύο αυτών επιθέσεων. Τέλος, η κλάση Heartbleed (κλάση 5) ταξινομείται σωστά και στις 3 περιπτώσεις χωρίς κανένα σφάλμα.

Στον δέκατο έβδομο συνδυασμό (Εικόνα 72) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (BENIGN) με 46.079 σωστές ταξινομήσεις και ελάχιστα σφάλματα (3 ως Brute Force, 17 ως SQL Injection και 4 ως XSS). Η κλάση Brute Force (κλάση 1) αναγνωρίζεται σωστά σε 292 περιπτώσεις, με 2 δείγματα να χάνονται ως BENIGN, 7 ως SQL Injection και 140 να ταξινομούνται λανθασμένα ως XSS, επιβεβαιώνοντας για άλλη μια φορά τη χαρακτηριστική σύγχυση μεταξύ Brute Force και XSS που παρατηρείται σταθερά σε όλους τους συνδυασμούς του Thursday dataset. Η κλάση SQL Injection (κλάση 2) εντοπίζεται σωστά και στις 6 περιπτώσεις χωρίς κανένα ψευδώς θετικό. Η κλάση XSS (κλάση 3) αναγνωρίζεται σωστά σε 110 περιπτώσεις, με την κύρια σύγχυση να εντοπίζεται προς την κλάση Brute Force (81 δείγματα) και 4 δείγματα να χάνονται ως BENIGN.



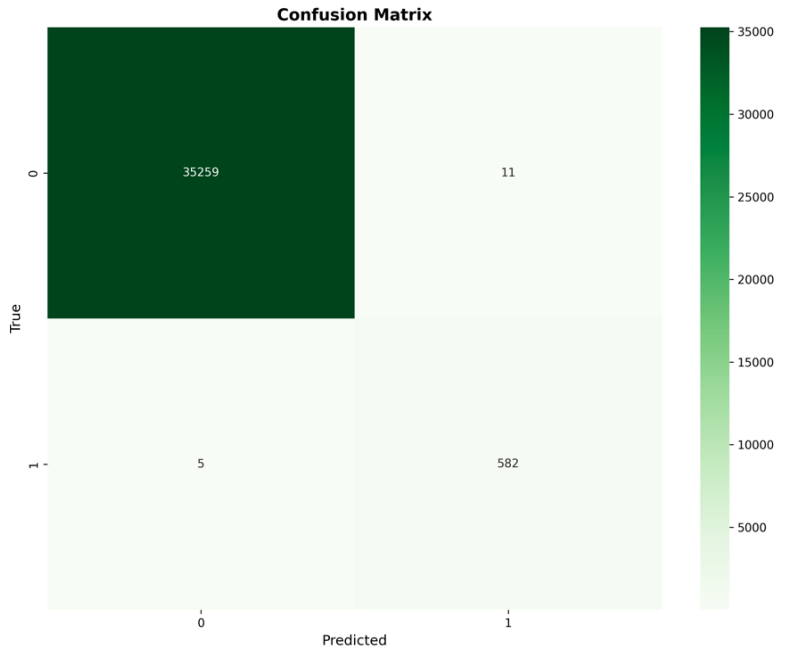
Εικόνα 71 Πίνακας σύγχυσης για τον 16ο συνδυασμό.



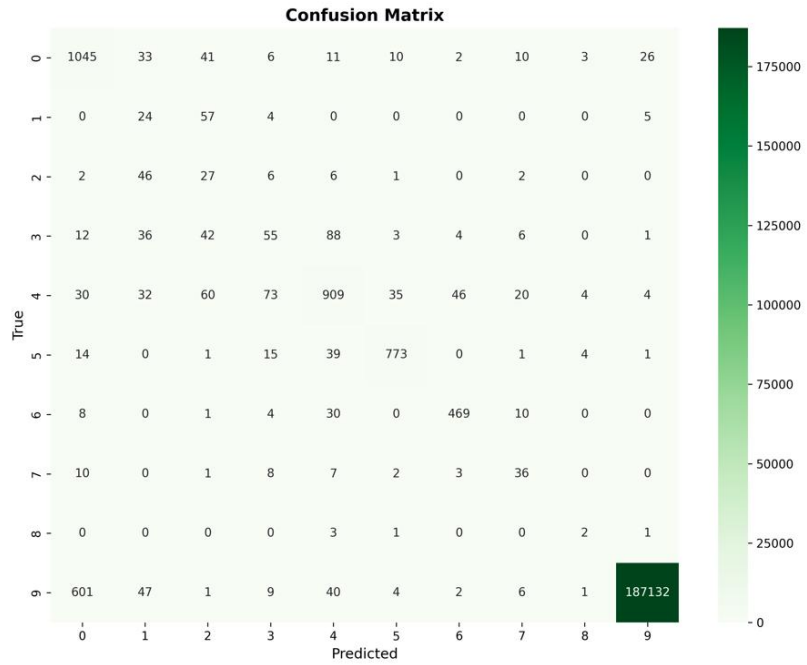
Εικόνα 72 Πίνακας σύγχυσης για τον 17ο συνδυασμό.

Στον δέκατο όγδοο συνδυασμό (Εικόνα 73) το μοντέλο επιτυγχάνει σχεδόν τέλειο διαχωρισμό μεταξύ κανονικής κίνησης και PortScan επιθέσεων. Συγκεκριμένα, 35.259 BENIGN ροές ταξινομούνται σωστά, με μόλις 11 να επισημαίνονται λανθασμένα ως PortScan (ψευδώς θετικές). Από την πλευρά των επιθέσεων, 582 PortScan ροές εντοπίζονται σωστά, ενώ μόλις 5 χάνονται ως κανονική κίνηση (ψευδώς αρνητικές). Το συνολικό πλήθος σφαλμάτων (16 δείγματα επί συνόλου 35.857) είναι εξαιρετικά μικρό, επιβεβαιώνοντας ότι η υβριδική μέθοδος εξισορρόπησης σε συνδυασμό με τη διαγραφή των NaN τιμών οδηγεί σε άριστη απόδοση για την ανίχνευση PortScan επιθέσεων.

Στον δέκατο ένατο συνδυασμό (Εικόνα 74) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (Normal, κλάση 9) με 187.132 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση Fuzzers (601 δείγματα) και Analysis (47 δείγματα). Η κλάση Exploits (κλάση 4) παρουσιάζει το μεγαλύτερο πλήθος σωστών ταξινομήσεων μεταξύ των επιθέσεων (909), αλλά και τη μεγαλύτερη σύγχυση με τις υπόλοιπες κλάσεις, κυρίως με DoS (73), Backdoors (60) και Analysis (32), αντικατοπτρίζοντας την πολυπλοκότητα της κατηγορίας αυτής. Η κλάση Generic (κλάση 5) αναγνωρίζεται σωστά σε 773 περιπτώσεις με ελάχιστα σφάλματα. Η κλάση Reconnaissance (κλάση 6) εντοπίζεται σωστά σε 469 περιπτώσεις, με μικρή σύγχυση προς την Fuzzers (8) και Exploits (30).



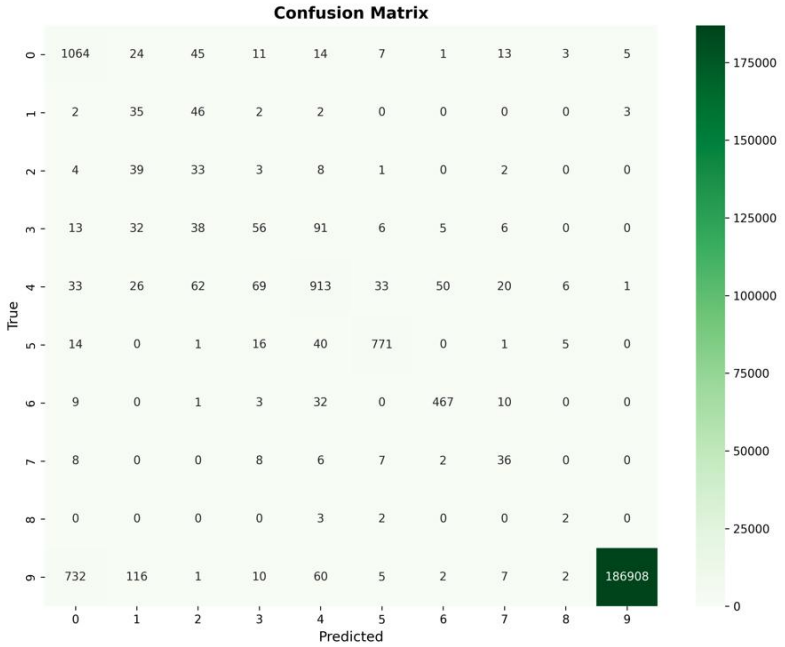
Εικόνα 73 Πίνακας σύγχυσης για τον 18ο συνδυασμό.



Εικόνα 74 Πίνακας σύγχυσης για τον 19ο συνδυασμό.

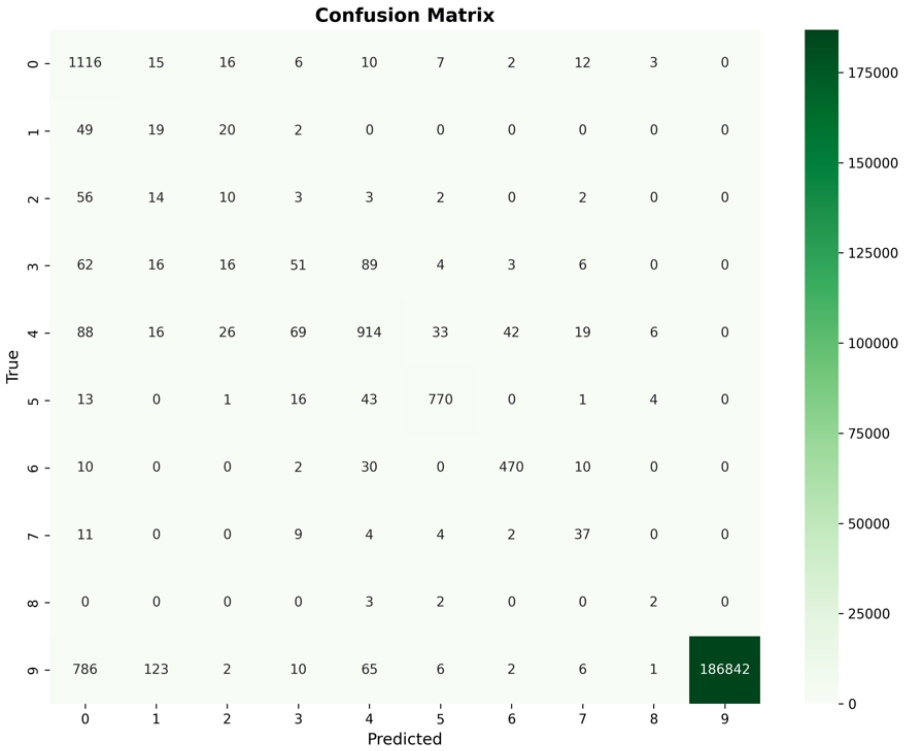
Οι κλάσεις Fuzzers (1.045), Analysis (24) και Shellcode (36) παρουσιάζουν υψηλή σύγχυση μεταξύ τους, κάτι που είναι αναμενόμενο λόγω της ομοιότητας των χαρακτηριστικών κίνησής τους. Η κλάση Worms (κλάση 8) παρουσιάζει τον μικρότερο αριθμό σωστών ταξινομήσεων (2), γεγονός που αποδίδεται στον εξαιρετικά μικρό αριθμό δειγμάτων της κατηγορίας αυτής στο σύνολο δεδομένων.

Στον εικοστό συνδυασμό (φαίνεται στην Εικόνα 75) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (Normal, κλάση 9) με 186.908 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με τους Fuzzers (732 δείγματα), την Analysis (116 δείγματα) και την Exploits (60 δείγματα). Η κλάση Exploits (κλάση 4) παρουσιάζει και εδώ το μεγαλύτερο πλήθος σωστών ταξινομήσεων μεταξύ των επιθέσεων (913), με κύρια σύγχυση προς τις DoS (69), Backdoors (62) και Fuzzers (33). Η κλάση Generic (κλάση 5) αναγνωρίζεται σωστά σε 771 περιπτώσεις με ελάχιστα σφάλματα. Η κλάση Reconnaissance (κλάση 6) εντοπίζεται σωστά σε 467 περιπτώσεις, με μικρή σύγχυση κυρίως προς την Exploits (32). Οι κλάσεις Analysis (35) και Backdoors (33) παρουσιάζουν αυξημένη σύγχυση μεταξύ τους και με άλλες κατηγορίες, αντικατοπτρίζοντας την ομοιότητα των χαρακτηριστικών κίνησής τους. Η κλάση Worms (κλάση 8) παρουσιάζει και πάλι τον μικρότερο αριθμό σωστών ταξινομήσεων (2), λόγω του εξαιρετικά μικρού αριθμού δειγμάτων της στο σύνολο δεδομένων.



Εικόνα 75 Πίνακας σύγχυσης για τον 20ο συνδυασμό.

Και στον εικοστό πρώτο συνδυασμό (παρουσιάζεται στην Εικόνα 76) το μοντέλο επιτυγχάνει εξαιρετική αναγνώριση της κανονικής κίνησης (Normal, κλάση 9) με 186.842 σωστές ταξινομήσεις, με τα κυριότερα σφάλματα να εντοπίζονται στη σύγχυση με τους Fuzzers (786 δείγματα) και την Analysis (123 δείγματα). Η κλάση Exploits (κλάση 4) παρουσιάζει και εδώ το μεγαλύτερο πλήθος σωστών ταξινομήσεων μεταξύ των επιθέσεων (914), με κύρια σύγχυση προς τις Fuzzers (88), DoS (69), Generic (33) και Backdoors (26). Η κλάση Generic (κλάση 5) αναγνωρίζεται σωστά σε 770 περιπτώσεις με ελάχιστα σφάλματα. Η κλάση Reconnaissance (κλάση 6) εντοπίζεται σωστά σε 470 περιπτώσεις, με μικρή σύγχυση κυρίως προς την Exploits (30) και την Shellcode (10). Οι κλάσεις Fuzzers (1.116), Analysis (19) και Backdoors (10) παρουσιάζουν αυξημένη σύγχυση μεταξύ τους, επιβεβαιώνοντας την ομοιότητα των χαρακτηριστικών κίνησής τους που παρατηρείται σταθερά και στους τρεις συνδυασμούς του UNSW-NB15. Η κλάση Worms (κλάση 8) παρουσιάζει και πάλι τον μικρότερο αριθμό σωστών ταξινομήσεων (2), λόγω του εξαιρετικά μικρού αριθμού δειγμάτων της στο σύνολο δεδομένων.



Εικόνα 76 Πίνακας σύγχυσης για τον 21ο συνδυασμό.

6.3. Επεξηγησιμότητα μοντέλου (XAI)

Μετά την ολοκλήρωση της εκπαίδευσης του μοντέλου XGBoost, το Shield IDS εφαρμόζει αυτόματα SHAP για την ερμηνεία των αποφάσεών του. Η ανάγκη για δημιουργία και χρήση του SHAP προκύπτει από την ίδια τη φύση του XGBoost καθώς, παρότι είναι εξαιρετικά αποδοτικός, χαρακτηρίζεται από αδιαφάνεια, λειτουργώντας σαν μαύρο κουτί ως προς τους λόγους για τους οποίους οδηγήθηκε σε μια απόφαση και ως προς τη λογική των προβλέψεών του [29]. Το SHAP εφαρμόζεται ως post-hoc μέθοδος (δηλαδή μετά την εκπαίδευση) και ποσοτικοποιεί τη συνεισφορά κάθε χαρακτηριστικού στην πρόβλεψη [16].

Για την επεξηγησιμότητα μεταξύ των μεθόδων SHAP και LIME επιλέχθηκε, για λόγους που αναφέρονται στη συνέχεια, η μέθοδος SHAP. Όπως φαίνεται και στη σχετική βιβλιογραφία, το LIME λειτουργεί δημιουργώντας διαταραγμένες εκδοχές ενός δείγματος κι εκπαιδεύοντας ένα απλό, ερμηνεύσιμο μοντέλο (όπως μια γραμμική παλινδρόμηση) για να προσεγγίσει τη συμπεριφορά του αρχικού μοντέλου τοπικά, στην περιοχή γύρω από το συγκεκριμένο δείγμα [22]. Αυτό σημαίνει ότι τα αποτελέσματά του ενδέχεται να μεταβάλλονται από εκτέλεση σε εκτέλεση λόγω της τυχαιότητας στην παραγωγή των

διαταραγμένων δειγμάτων, γεγονός που μειώνει την αξιοπιστία και τη συνέπεια των ερμηνειών.

Από την άλλη πλευρά, το SHAP βασίζεται στη θεωρία παιγνίων και τις τιμές Shapley, εγγυώμενο τη δίκαιη κατανομή της συνεισφοράς κάθε χαρακτηριστικού σε μία πρόβλεψη, λαμβάνοντας υπόψη όλους τους πιθανούς συνδυασμούς χαρακτηριστικών [22], [40]. Επίσης, μέσω της υλοποίησης shap.TreeExplainer, το SHAP εκμεταλλεύεται άμεσα τη δομή δένδρων του XGBoost, παράγοντας ακριβείς και συνεπείς ερμηνείες. Η υλοποίηση Tree SHAP, που είναι ενσωματωμένη στο XGBoost, μπορεί να χρησιμοποιηθεί για να εξηγήσει τις προβλέψεις ταξινόμησης του συνόλου δοκιμών [24].

Το στάδιο του SHAP εκτελείται αυτόματα αμέσως μετά την εκπαίδευση του τελικού μοντέλου, εντός του TrainingWorker. Όπως φαίνεται και στο ακόλουθο απόσπασμα κώδικα, κατά την δειγματοληψία και τον υπολογισμό των τιμών, λαμβάνεται ένα τυχαίο δείγμα έως 500 εγγραφών από το σύνολο εκπαίδευσης (μετά το resampling) για λόγους υπολογιστικής αποδοτικότητας, διατηρώντας παράλληλα την αντιπροσωπευτικότητα.

```
Τυχαίο δείγμα έως 500 γραμμών από τα δεδομένα εκπαίδευσης
shap_sample = X_train.sample(n=min(500, len(X_train)), random_state=42)
# explainer για μοντέλα δέντρων πάνω στο εκπαιδευμένο μοντέλο
explainer = shap.TreeExplainer(best_model)
# Υπολογισμός SHAP τιμών για το δείγμα που επιλέχθηκε
shap_values = explainer.shap_values(shap_sample)
```

Στη συνέχεια, εφόσον επιστραφεί λίστα, ο κώδικας παίρνει τον μέσο όρο των απόλυτων τιμών.

```
if isinstance(shap_values, list): # Αν οι SHAP τιμές είναι λίστα
      num_classes = len(shap_values) # Υπολογισμός κλάσεων λίστας
      # Καταγραφή log (πολυταξινόμηση)
      self.log(f"Πολυταξινόμηση με {num_classes} κλάσεις (SHAP επιστρέφει
      λίστα)...")

      # Μ.Ο. απόλυτων SHAP τιμών για όλες τις κλάσεις
      mean_shap = np.mean([np.abs(sv) for sv in shap_values], axis=0)
```

```
        # Ορίσμός τιμών που θα χρησιμοποιηθούν στο γράφημα
        shap_values_to_plot = mean_shap

        # Νέο figure μεγέθους 10x8 ινστών
        plt.figure(figsize=(10, 8))
        # summary plot για τα 20 πιο σημαντικά χαρακτηριστικά
        shap.summary_plot(shap_values_to_plot, shap_sample, show=False,
        max_display=20)
        # Τίτλος γραφήματος
        plt.title("SHAP Summary Plot (Mean over all classes)", fontsize=14,
        weight='bold')
        plt.tight_layout() # Περιθώρια
        # Αποθήκευση γραφήματος ως εικόνα
        plt.savefig("shap_summary_plot.png", dpi=300, bbox_inches="tight")
        plt.close() # Κλείσιμο γραφήματος (απελευθέρωση μνήμης)
```

Αν δεν εντοπιστούν πολλαπλές κλάσεις (όπου οι SHAP τιμές επιστρέφονται ως απλός πίνακας), εκτελείται το ακόλουθο τμήμα κώδικα. Οι τρεις κατηγορίες γραφημάτων που παράγονται αυτόματα εμφανίζονται στο GUI μόλις ολοκληρωθεί η εκπαίδευση.

```
else: # Δυαδική ταξινόμηση
        shap_values_to_plot = shap_values # SHAP τιμές (όπως επιστράφηκαν)
        plt.figure(figsize=(10, 8)) # Νέο figure (10x8 ίντσες)
        # summary plot σημαντικότερων χαρακτηριστικών
        shap.summary_plot(shap_values_to_plot, shap_sample, show=False,
        max_display=20)
        # Τίτλος γραφήματος
        plt.title("SHAP Summary Plot", fontsize=14, weight='bold')
        plt.tight_layout() # Περιθώρεια
        # Αποθήκευση γραφήματος ως εικόνα
        plt.savefig("shap_summary_plot.png", dpi=300, bbox_inches="tight")
        plt.close() # Κλείσιμο γραφήματος (απελευθέρωση μνήμης)
```

```
plt.figure(figsize=(10, 8))
shap.summary_plot(shap_values_to_plot, shap_sample,
                  plot_type="bar", show=False, max_display=20)
plt.title("SHAP Bar Plot", fontsize=14, weight='bold')
plt.tight_layout()
plt.savefig("shap_bar_plot.png", dpi=300, bbox_inches="tight")
plt.close()
```

6.3.1. Ανάλυση σημαντικότητας χαρακτηριστικών

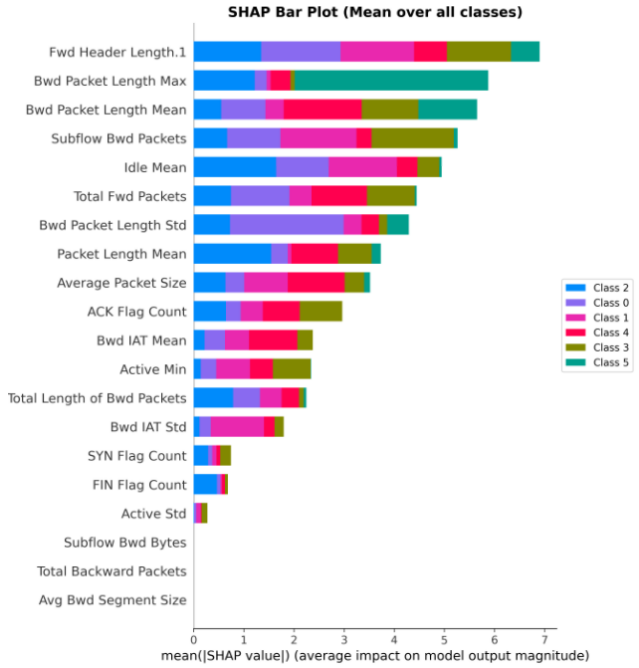
Η εφαρμογή παράγει το SHAP Bar Plot το οποίο παρουσιάζει τα χαρακτηριστικά ταξινομημένα βάσει μέσης απόλυτης τιμής SHAP (mean |SHAP value|). Δίνει μια πιο εύκολη κατανόηση ως προς το ποια είναι τα χαρακτηριστικά που είχαν τον μεγαλύτερο αντίκτυπο σε όλες τις προβλέψεις. Αυτό, είναι ιδιαίτερα χρήσιμο για τους ειδικούς τομέα και τους διαχειριστές δικτύων ώστε να μπορούν να επικεντρωθούν στα πιο κρίσιμα μοτίβα [15]. Ειδικότερα, για την περίπτωση της πολυκλασικής ταξινόμησης, εμφανίζεται ξεχωριστά η συνεισφορά ανά κλάση επίθεσης.

Παρακάτω θα κάνουμε μια σύντομη ανάλυση των εικοσιένα συνδυασμών που παρουσιάζει ο Πίνακας 6.

Για τον πρώτο συνδυασμό βλέπουμε από το διάγραμμα (στην Εικόνα 77) ότι το Fwd Header Length.1 εμφανίζεται ως το πιο σημαντικό χαρακτηριστικό συνολικά (mean |SHAP value| ≈ 7), με ισχυρή συνεισφορά κυρίως από τις κλάσεις 0 (μοβ) και 1 (ροζ). Αυτό σημαίνει ότι το μέγεθος της επικεφαλίδας των προωθούμενων πακέτων αποτελεί σημαντικό διαχωριστικό στοιχείο για την ταξινόμηση.

Ακολουθούν τα Bwd Packet Length Max και Bwd Packet Length Mean (≈6 και 5.7 αντίστοιχα), με εμφανή συνεισφορά από τις κλάσεις 5 και 4 αντίστοιχα, γεγονός που σημαίνει ότι το μέγεθος των πακέτων στην αντίστροφη κατεύθυνση (backward) αποτελεί χαρακτηριστικό για συγκεκριμένους τύπους επίθεσης.

Το Subflow Bwd Packets και το Idle Mean κατατάσσονται επίσης υψηλά (≈5.2 και 5.0), με το Idle Mean να επηρεάζεται κυρίως από την κλάση 1 (ροζ) και την κλάση 2 (μπλε), υποδηλώνοντας ότι ο χρόνος αδράνειας της σύνδεσης σχετίζεται με συγκεκριμένες κατηγορίες επιθέσεων.



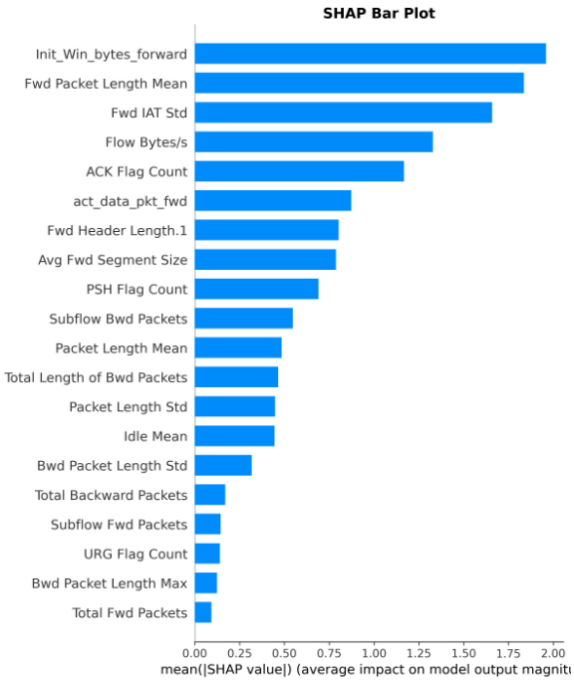
Εικόνα 77 Διάγραμμα Bar Plot για τον 1ο συνδυασμό.

Τα χαρακτηριστικά Total Fwd Packets, Bwd Packet Length Std και Packet Length Mean εμφανίζουν μέτρια σημαντικότητα (4.6 και 4.2 αντίστοιχα), ενώ χαρακτηριστικά όπως τα Subflow Bwd Bytes, Total Backwad Packets και Avg Bwd Segment Size εμφανίζουν σχεδόν μηδενική συνεισφορά, άρα έχουν αμελητέα επίδραση στις αποφάσεις του μοντέλου.

Για τον δεύτερο συνδυασμό (βλέπε Εικόνα 78) παρατηρούμε το Init_Win_bytes_forward εμφανίζεται ως το σημαντικότερο χαρακτηριστικό (mean |SHAP value| ≈ 2.0), υποδηλώνοντας ότι το αρχικό μέγεθος του παραθύρου TCP προς την προωθούμενη κατεύθυνση είναι ο πιο επιδραστικός παράγοντας διάκρισης ως προς τις αποφάσεις που θα ληφθούν. Αυτού του είδους η συμπεριφορά θεωρείται λογική, καθώς επιθέσεις όπως DDoS και port scanning συχνά παρουσιάζουν μη τυπικές τιμές TCP window.

Ακολουθούν τα χαρακτηριστικά Fwd Packet Length Mean (≈ 1.85) και Fwd IAT Std (≈ 1.65), που σχετίζονται με το μέγεθος και τη χρονική κατανομή των πακέτων στην προωθούμενη κατεύθυνση. Η υψηλή τυπική απόκλιση των χρονικών διαστημάτων (IAT Std) υποδηλώνει ότι η αστάθεια στους ρυθμούς αποστολής αποτελεί βασικό δείκτη επίθεσης.

Το Flow Bytes/s (≈ 1.3) και το ACK Flag Count (≈ 1.15) συμπληρώνουν την ομάδα των πιο σημαντικών χαρακτηριστικών, με τον ρυθμό Bytes ανά δευτερόλεπτο και τον αριθμό ACK flags να υποδεικνύουν ασυνήθιστα πρότυπα κυκλοφορίας.

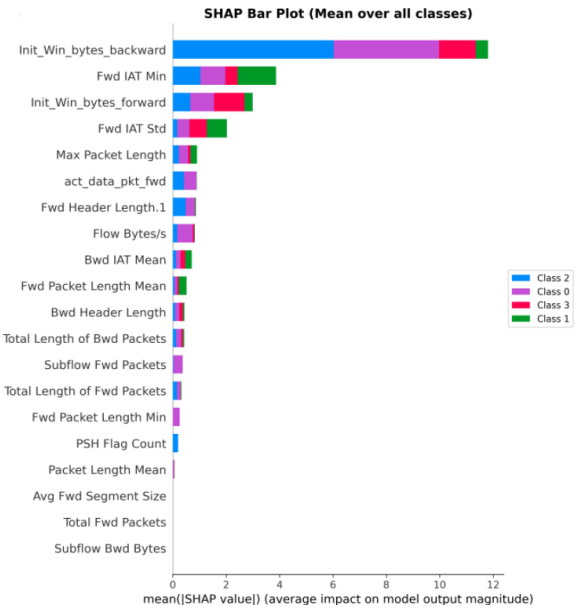


Εικόνα 78 Διάγραμμα Bar Plot για τον 2ο συνδυασμό.

Τα χαρακτηριστικά από act_data_pkt_fwd έως Idle Mean παρουσιάζουν μέτρια συνεισφορά (0.45–0.9), ενώ τα τελευταία πέντε (URG Flag Count, Bwd Packet Length Max, Subflow Fwd Packets, Total Backward Packets, Total Fwd Packets) έχουν σχεδόν αμελητέα επίδραση (< 0.15). Το μοντέλο βασίζεται κυρίως σε χαρακτηριστικά TCP παραθύρου, μεγέθους πακέτων και χρονικών διαστημάτων για τη διάκριση κανονικής κίνησης από επίθεση.

Για τον τρίτο συνδυασμό (Εικόνα 79) παρατηρούμε ότι το Init_Win_bytes_backward επικρατεί με διαφορά (mean |SHAP value| ≈ 12), με σημαντική συνεισφορά από την κλάση 2 (μπλε), ακολουθούμενη από κλάση 0 (μοβ) και κλάση 3 (κόκκινο). Αυτό υποδηλώνει ότι το αρχικό μέγεθος TCP παραθύρου στην αντίστροφη κατεύθυνση είναι ο παράγοντας απόφασης για τη διάκριση της κλάσης 2 από τις υπόλοιπες, πιθανώς λόγω χαρακτηριστικής συμπεριφοράς συγκεκριμένου τύπου επίθεσης.

Το Fwd IAT Min κατατάσσεται δεύτερο (≈ 4), με συνεισφορά από όλες τις κλάσεις, γεγονός που υποδηλώνει ότι ο ελάχιστος χρόνος μεταξύ της άφιξης δύο διαδοχικών πακέτων προς τα εμπρός αποτελεί διαγνωστικό για όλες τις κατηγορίες. Το Init_Win_bytes_forward (≈ 3) και το Fwd IAT Std (≈ 2.2) συμπληρώνουν το σύνολο των σημαντικών χαρακτηριστικών, με έμφαση στα παράθυρα TCP και την τυπική απόκλιση του χρόνου μεταξύ των πακέτων.

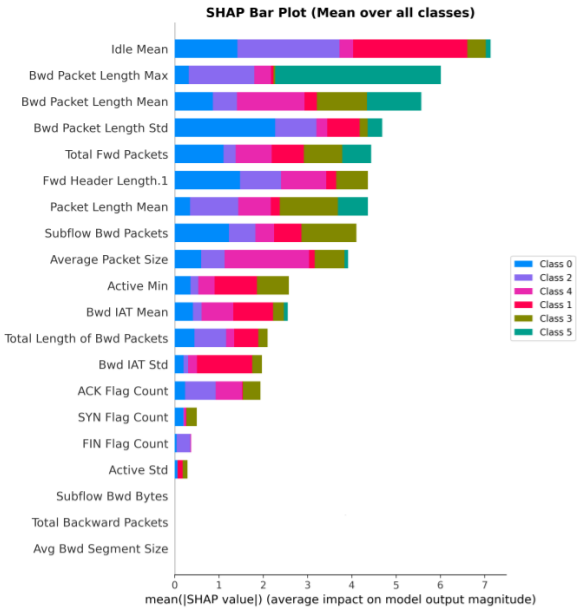


Εικόνα 79 Διάγραμμα Bar Plot για τον 3ο συνδυασμό.

Από το Max Packet Length και κάτω, η συνεισφορά πέφτει απότομα κάτω από 1.0, ενώ τα τελευταία χαρακτηριστικά (Avg Fwd Segment Size, Total Fwd Packets, Subflow Bwd Bytes) έχουν μηδενική επίδραση στις αποφάσεις του μοντέλου. Κατά συνέπεια, το μοντέλο αυτού του συνδυασμού βασίζεται κυρίως στα TCP παράθυρα και το χρόνο των πακέτων, με την κλάση 2 να επηρεάζει δυσανάλογα τη σημαντικότητα του Init_Win_bytes_backward.

Στον τέταρτο συνδυασμό (Εικόνα 80) παρατηρούμε ότι το Idle Mean κυριαρχεί με mean |SHAP value| ≈ 7.1, με κύρια συνεισφορά από την κλάση 1 (κόκκινο) και την κλάση 2 (μοβ). Αυτό υποδηλώνει ότι ο μέσος χρόνος αδράνειας μιας σύνδεσης είναι ο πιο σημαντικός παράγοντας καθώς συγκεκριμένες κατηγορίες επιθέσεων παρουσιάζουν χαρακτηριστικά πρότυπα αδράνειας που ξεχωρίζουν από την κανονική κίνηση. Τα Bwd Packet Length Max (≈ 6.0) και Bwd Packet Length Mean (≈ 5.7) κατατάσσονται δεύτερο και τρίτο αντίστοιχα, με έντονη παρουσία της κλάσης 5 (πετρόλ) και της κλάσης4 (ροζ) αντίστοιχα. Συνεπώς, το μέγεθος των πακέτων στην αντίστροφη κατεύθυνση αποδεικνύεται κρίσιμο για τη διάκριση συγκεκριμένων τύπων επίθεσης.

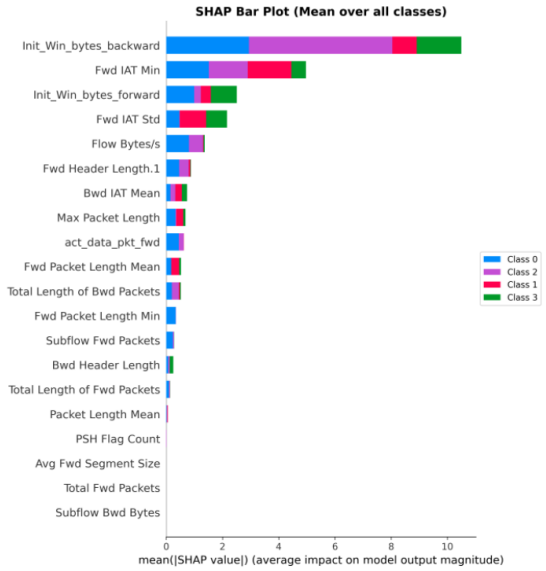
Το Bwd Packet Length Std (≈ 4.7) και το Total Fwd Packets (≈ 4.5) συμπληρώνουν την ομάδα των πέντε πρώτων σημαντικότερων χαρακτηριστικών, με συνεισφορά από πολλές κλάσεις γεγονός που τα καθιστά σημαντικά ως προς τη διάγνωση για το σύνολο των επιθέσεων.



Εικόνα 80 Διάγραμμα Bar Plot για τον 4ο συνδυασμό.

Τα χαρακτηριστικά Fwd Header Length.1, Packet Length Mean, Subflow Bwd Packets και Average Packet Size εμφανίζουν εξίσου αξιόλογη συνεισφορά (≈ 4–4.5), ενώ από SYN Flag Count και κάτω η επίδραση πέφτει κάτω από 0.6. Τα τελευταία τρία χαρακτηριστικά (Subflow Bwd Bytes, Total Backward Packets, Avg Bwd Segment Size) έχουν ουσιαστικά μηδενική επίδραση. Σε αντίθεση με τους προηγούμενους συνδυασμούς που βασίζονταν στα TCP παράθυρα, εδώ το δίνεται έμφαση στον χρόνο αδράνειας και τα στατιστικά μεγέθους πακέτων backward, υποδηλώνοντας διαφορετική δομή επιθέσεων στο σύνολο δεδομένων αυτού του συνδυασμού.

Για τον πέμπτο συνδυασμό (Εικόνα 81) παρατηρούμε ότι το Init_Win_bytes_backward κυριαρχεί ξανά με mean |SHAP value| ≈ 10.5, με κύρια συνεισφορά από τις κλάσεις 0 (μπλε) και 2 (μοβ), ενώ οι κλάσεις 1 (κόκκινο) και 3 (πράσινο) συνεισφέρουν λιγότερο. Παρατηρούμε ότι αυτό το χαρακτηριστικό εμφανίζεται πρώτο και σε αυτόν τον συνδυασμό, επιβεβαιώνοντας τη σταθερή διαγνωστική αξία του TCP backward παραθύρου σε διαφορετικούς συνδυασμούς του συνόλου δεδομένου. Το Fwd IAT Min κατατάσσεται δεύτερο (≈ 5.0) με ισορροπημένη συνεισφορά από όλες τις κλάσεις, υποδηλώνοντας ότι ο ελάχιστος χρόνος μεταξύ της άφιξης δύο διαδοχικών πακέτων προς τα εμπρός παραμένει γενικά διαγνωστικό.

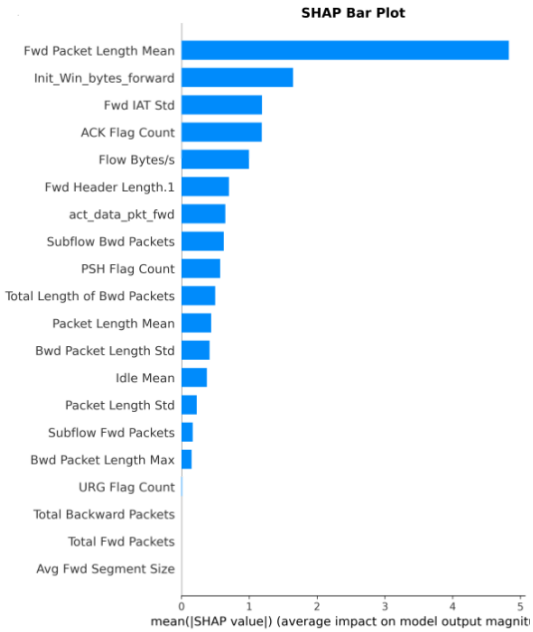


Εικόνα 81 Διάγραμμα Bar Plot για τον 5ο συνδυασμό.

Τα Init_Win_bytes_forward (≈ 2.7) και Fwd IAT Std (≈ 2.5) ακολουθούν, με κυρίαρχη συνεισφορά από τις κλάσεις 0 και 1 αντίστοιχα, αναδεικνύοντας ότι η μεταβλητότητα των χρονικών διαστημάτων και το μέγεθος TCP παραθύρου στην προωθούμενη κατεύθυνση διαφοροποιούν συγκεκριμένες κλάσεις. Το Flow Bytes/s (≈ 1.4) παρουσιάζει αξιόλογη συνεισφορά κυρίως από τις κλάσεις 0 και 2, ενώ από Fwd Header Length.1 και κάτω η επίδραση πέφτει κάτω από 1.0. Τα τελευταία features (PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets, Subflow Bwd Bytes) έχουν μηδενική επίδραση.

Για τον έκτο συνδυασμό (Εικόνα 82) παρατηρούμε ότι το Fwd Packet Length Mean κυριαρχεί με εξαιρετικά μεγάλη διαφορά (mean |SHAP value| ≈ 4.8), ξεπερνώντας κατά πολύ όλα τα υπόλοιπα χαρακτηριστικά. Αυτό σημαίνει ότι το μοντέλο βασίζεται κυρίως στο μέσο μέγεθος των πακέτων που αποστέλλονται για να αποφασίσει αν μια κίνηση είναι κανονική ή κακόβουλη. Ως εκ τούτου, οι επιθέσεις σε αυτόν τον συνδυασμό βασίζονται στο μέγεθος των πακέτων, κάτι που κάνει αυτό το χαρακτηριστικό εξαιρετικά εύκολα αναγνωρίσιμο από το μοντέλο, χωρίς να χρειάζεται να βασιστεί ιδιαίτερα σε άλλα χαρακτηριστικά.

Το Init_Win_bytes_forward (≈ 1.65) και το Fwd IAT Std (≈ 1.15) κατατάσσονται δεύτερο και τρίτο, ακολουθούμενα από τα ACK Flag Count (≈ 1.15) και Flow Bytes/s (≈ 1.0). Συνολικά, τα χαρακτηριστικά αυτά σχηματίζουν μια δευτερεύουσα ομάδα με σχετικά ισάξια συνεισφορά.

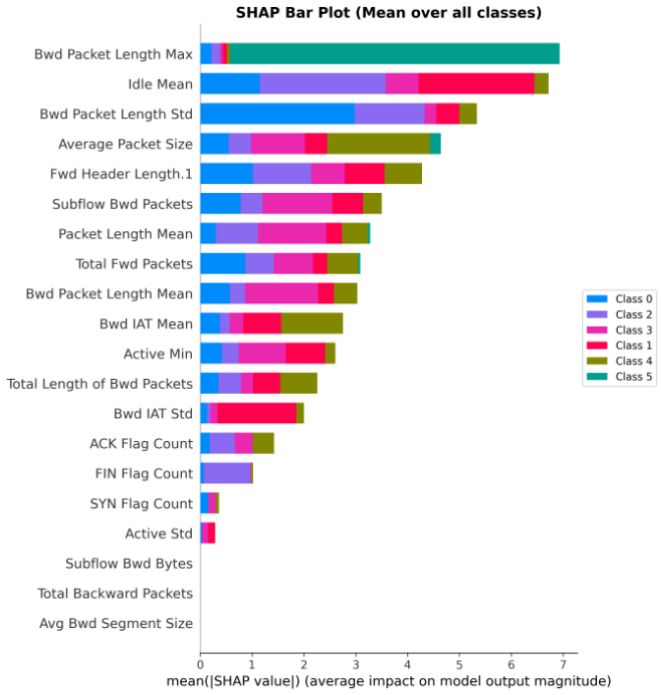


Εικόνα 82 Διάγραμμα Bar Plot για τον 6ο συνδυασμό.

Από το Fwd Header Length.1 μέχρι το Idle Mean παρατηρείται σταδιακή μείωση της σημαντικότητας (0.35–0.7), ενώ από το Packet Length Std και κάτω η επίδραση είναι κάτω από 0.2. Τα τελευταία τέσσερα χαρακτηριστικά (URG Flag Count, Total Backward Packets, Total Fwd Packets, Avg Fwd Segment Size) έχουν μηδενική επίδραση.

Για τον έβδομο συνδυασμό (Εικόνα 83) παρατηρούμε ότι τα Bwd Packet Length Max (≈ 7.0) και Idle Mean (≈ 6.7) κατατάσσονται πρώτα με σχεδόν ισάξια συνεισφορά, αλλά με πολύ διαφορετική σύνθεση κλάσεων. Το Bwd Packet Length Max επηρεάζεται κυρίως από την κλάση 5, ενώ το Idle Mean έχει ισορροπημένη συνεισφορά από τις κλάσεις 2 και 1, όπως αντίστοιχα και από τις 0 και 3. Αυτό σημαίνει ότι η κλάση 5 αναγνωρίζεται κυρίως από το μέγεθος των backward πακέτων, ενώ οι υπόλοιπες κλάσεις διακρίνονται περισσότερο από τον χρόνο αδράνειας.

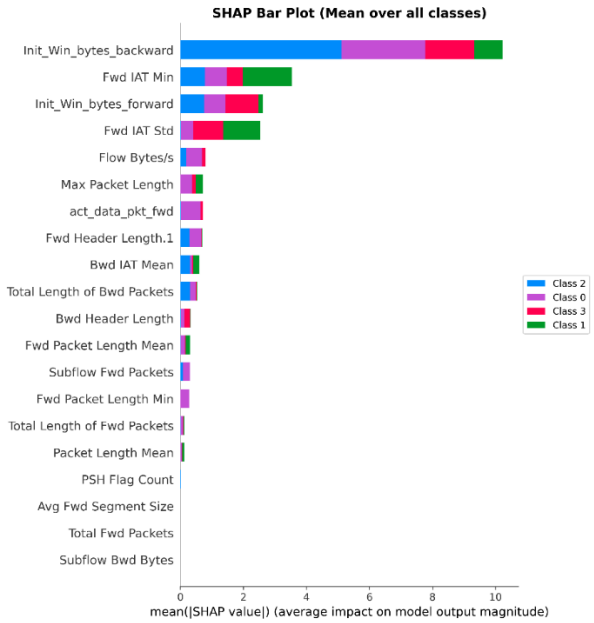
Το Bwd Packet Length Std (≈ 5.3) ακολουθεί με ισχυρή συνεισφορά από τη κλάση 0 και 2, υποδηλώνοντας ότι η μεταβλητότητα του μεγέθους backward πακέτων είναι σημαντική για τη διάκριση. Τα Average Packet Size (≈ 4.6) και Fwd Header Length.1 (≈ 4.3) παρουσιάζουν μια πιο ισορροπημένη συνεισφορά από πολλές κλάσεις, καθιστώντας τα γενικά διαγνωστικά χαρακτηριστικά. Από Subflow Bwd Packets μέχρι Total Length of Bwd Packets η συνεισφορά κυμαίνεται μεταξύ 2.3 και 3.5, ενώ από Bwd IAT Std και κάτω πέφτει σταδιακά. Τα τελευταία τρία features (Subflow Bwd Bytes, Total Backward Packets, Avg Bwd Segment Size) έχουν μηδενική επίδραση.



Εικόνα 83 Διάγραμμα Bar Plot για τον 7ο συνδυασμό.

Στον όγδοο συνδυασμό (φαίνεται στην Εικόνα 84) παρατηρούμε ότι το Init_Win_bytes_backward κυριαρχεί με mean |SHAP value| ≈ 10.3, με κύρια συνεισφορά από τις κλάσεις 2 και 0, υποδηλώνοντας ότι το αρχικό μέγεθος του TCP παραθύρου στην αντίστροφη κατεύθυνση είναι ο κύριος διαχωριστικός παράγοντας. Το Fwd IAT Min (≈ 3.5) κατατάσσεται δεύτερο με παρόμοια συνεισφορά από όλες τις κλάσεις, υποδηλώνοντας ότι το ελάχιστο χρονικό διάστημα μεταξύ προωθούμενων πακέτων είναι γενικά διαγνωστικό για το σύνολο των κλάσεων.

Τα Init_Win_bytes_forward (≈ 2.7) και Fwd IAT Std (≈ 2.6) ακολουθούν με επίσης παρόμοια συνεισφορά, με τη κλάση 1 να παίζει κυρίαρχο ρόλο στο Fwd IAT Std, υποδηλώνοντας ότι η μεταβλητότητα των χρονικών διαστημάτων αποτελεί ιδιαίτερο χαρακτηριστικό για αυτή την κλάση. Από το Flow Bytes/s (≈ 0.7) και κάτω η συνεισφορά πέφτει απότομα, ενώ τα τελευταία τέσσερα features (PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets, Subflow Bwd Bytes) έχουν μηδενική επίδραση στις αποφάσεις του μοντέλου.

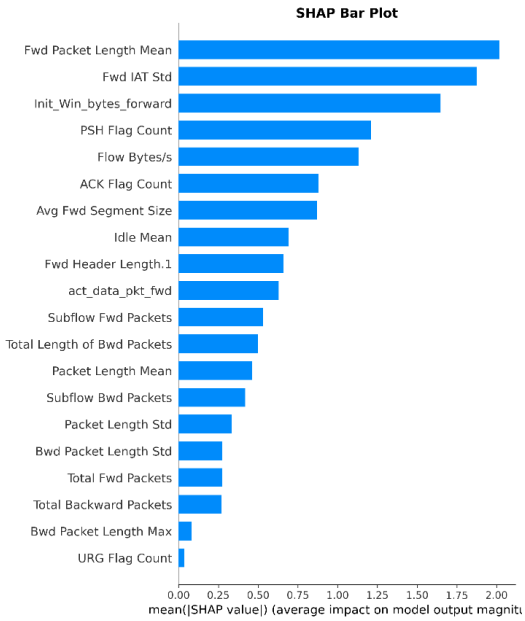


Εικόνα 84 Διάγραμμα Bar Plot για τον 8ο συνδυασμό.

Για τον ένατο συνδυασμό (όπως δείχνει η Εικόνα 85) παρατηρούμε ότι τα Fwd Packet Length Mean (≈ 2.0) και Fwd IAT Std (≈ 1.9) κατατάσσονται πρώτα με σχεδόν ισάξια συνεισφορά, υποδηλώνοντας ότι το μοντέλο βασίζεται εξίσου στο μέσο μέγεθος των προωθούμενων πακέτων και στη μεταβλητότητα των χρονικών διαστημάτων τους για να διακρίνει την κανονική κίνηση από την κακόβουλη.

Το Init_Win_bytes_forward (≈ 1.65) ακολουθεί τρίτο, δείχνοντας ότι το αρχικό μέγεθος TCP παραθύρου στην προωθούμενη κατεύθυνση παραμένει και εδώ σημαντικός παράγοντας. Αξιοσημείωτη είναι η παρουσία του PSH Flag Count (≈ 1.2) στην τέταρτη θέση, όπου το πλήθος των PSH flags υποδηλώνει ότι ο τρόπος με τον οποίο τα δεδομένα ωθούνται προς στην εφαρμογή αποτελεί χαρακτηριστικό μοτίβο των επιθέσεων αυτού του συνόλου δεδομένων.

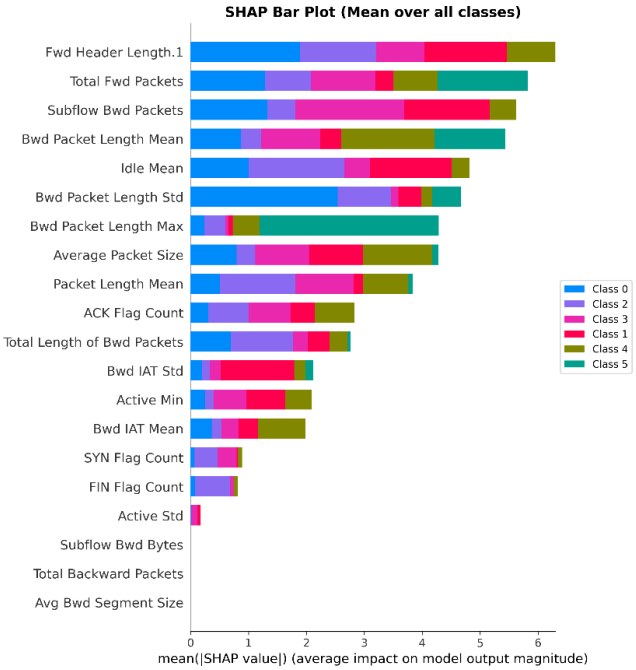
Στη συνέχεια βλέπουμε το Flow Bytes/s (≈ 1.1) και το ACK Flag Count (≈ 0.85), ενώ από Avg Fwd Segment Size μέχρι Total Backward Packets (≈ 0.28–0.85) παρατηρείται σταδιακή και ομαλή μείωση της συνεισφοράς γεγονός που υποδηλώνει ότι πολλά χαρακτηριστικά συμβάλλουν μέτρια, χωρίς κανένα να κυριαρχεί αποκλειστικά. Τα Bwd Packet Length Max και URG Flag Count έχουν σχεδόν μηδενική επίδραση.



Εικόνα 85 Διάγραμμα Bar Plot για τον 9ο συνδυασμό.

Όπως φαίνεται, στον δέκατο συνδυασμό (στην Εικόνα 86) το Fwd Header Length.1 (≈ 6.3) κυριαρχεί με παρόμοια συνεισφορά από σχεδόν όλες τις κλάσεις, με έμφαση στις 0 και 1. Αυτό δείχνει ότι το μέγεθος της επικεφαλίδας των προωθούμενων πακέτων αποτελεί γενικά διαγνωστικό χαρακτηριστικό για το σύνολο των κλάσεων. Τα Total Fwd Packets (≈ 5.8) και Subflow Bwd Packets (≈ 5.6) ακολουθούν με παρόμοια συνεισφορά, αναδεικνύοντας ότι ο αριθμός των πακέτων τόσο στην κατεύθυνση προς τα εμπρός όσο και στην αντίστροφη κατεύθυνση είναι εξίσου σημαντικός για τη διάκριση των κλάσεων. Αξιοσημείωτη είναι η ισχυρή παρουσία της κλάσης 5 στο Total Fwd Packets και της κλάσης 3 στο Subflow Bwd Packets.

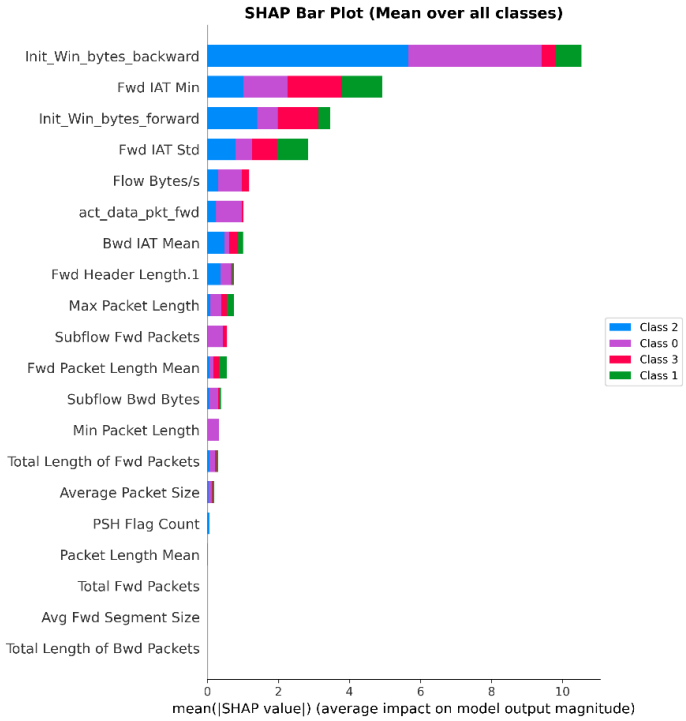
Το Bwd Packet Length Mean (≈ 5.4) και το Idle Mean (≈ 4.8) συμπληρώνουν την ομάδα των πέντε πρώτων χαρακτηριστικών, με το Idle Mean να επηρεάζεται κυρίως από τις κλάσεις 1 και 2. Τα Bwd Packet Length Std, Bwd Packet Length Max και Average Packet Size εμφανίζουν επίσης αξιόλογη συνεισφορά (4.2–4.7) με ισορροπημένη κατανομή κλάσεων. Από το Packet Length Mean μέχρι το Bwd IAT Mean η συνεισφορά κυμαίνεται μεταξύ 2.0 και 3.9, ενώ τα SYN/FIN Flag Count παρουσιάζουν μέτρια επίδραση (≈ 0.8) κυρίως από την κλάση 2. Τα τελευταία τρία χαρακτηριστικά (Subflow Bwd Bytes, Total Backward Packets, Avg Bwd Segment Size) έχουν μηδενική επίδραση.



Εικόνα 86 Διάγραμμα Bar Plot για τον 10ο συνδυασμό.

Στον ενδέκατο συνδυασμό (Εικόνα 87) το Init_Win_bytes_backward κυριαρχεί με mean |SHAP value| ≈ 10.5, με κύρια συνεισφορά από τις κλάσεις 2 και 0, ενώ οι κλάσεις 1 και 3 συνεισφέρουν πολύ λιγότερο. Αυτό υποδηλώνει ότι το αρχικό μέγεθος TCP παραθύρου στην αντίστροφη κατεύθυνση είναι καθοριστικός παράγοντας κυρίως για τη διάκριση των κλάσεων 2 και 0. Το Fwd IAT Min (≈ 5.0) ακολουθεί με πιο ισορροπημένη συνεισφορά από όλες τις κλάσεις, δείχνοντας ότι το ελάχιστο χρονικό διάστημα μεταξύ προωθούμενων πακέτων είναι γενικά διαγνωστικό για το σύνολο των κλάσεων.

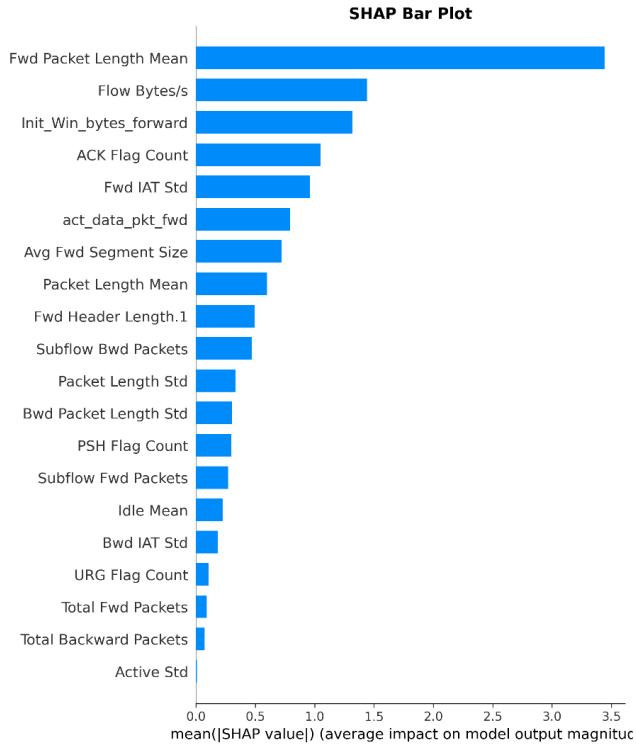
Τα Init_Win_bytes_forward (≈ 3.5) και Fwd IAT Std (≈ 3.0) ακολουθούν, με κυρίαρχη συνεισφορά από την κλάση 2 στο πρώτο και από την κλάση 1 στο δεύτερο, υποδηλώνοντας ότι αυτά τα χαρακτηριστικά είναι ιδιαίτερα για τις συγκεκριμένες κλάσεις. Από το Flow Bytes/s (≈ 1.1) και κάτω η συνεισφορά πέφτει, με χαρακτηριστικά όπως τα act_data_pkt_fwd, Bwd IAT Mean, Fwd Header Length.1 να παρουσιάζουν μέτρια επίδραση (≈ 0.8–1.0), ενώ τα τελευταία πέντε (Packet Length Mean, Total Fwd Packets, Avg Fwd Segment Size, Total Length of Bwd Packets, PSH Flag Count) έχουν ουσιαστικά μηδενική επίδραση.



Εικόνα 87 Διάγραμμα Bar Plot για τον 11ο συνδυασμό.

Για τον δωδέκατο συνδυασμό (παρουσιάζεται στην Εικόνα 88) παρατηρούμε ότι το Fwd Packet Length Mean κυριαρχεί με mean |SHAP value| ≈ 3.45, ξεπερνώντας σημαντικά όλα τα υπόλοιπα χαρακτηριστικά. Αυτό σημαίνει ότι το μοντέλο βασίζεται κυρίως στο μέσο μέγεθος των προωθούμενων πακέτων για να διακρίνει την κανονική κίνηση από την κακόβουλη. Τα Flow Bytes/s (≈ 1.45) και Init_Win_bytes_forward (≈ 1.35) κατατάσσονται δεύτερο και τρίτο κατά σειρά, ακολουθούμενα από τα ACK Flag Count (≈ 1.05) και Fwd IAT Std (≈ 1.0). Αυτά τα χαρακτηριστικά σχηματίζουν μια δευτερεύουσα ομάδα με παρόμοια συνεισφορά, υποδηλώνοντας ότι ο ρυθμός των bytes, το TCP παράθυρο και τα χρονικά χαρακτηριστικά φέρουν συμπληρωματικό ρόλο.

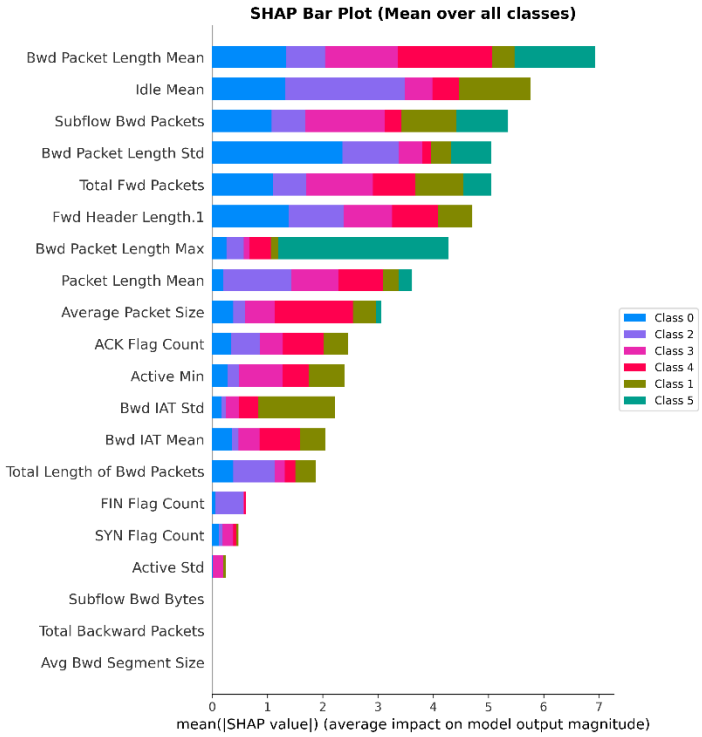
Από act_data_pkt_fwd (≈ 0.78) μέχρι Bwd IAT Std (≈ 0.17) παρατηρείται σταδιακή και ομαλή μείωση της συνεισφοράς, με χαρακτηριστικά όπως τα Avg Fwd Segment Size, Packet Length Mean, Fwd Header Length.1 να διατηρούν αξιόλογη αλλά δευτερεύουσα επίδραση. Ακολουθούν τα τελευταία τρία χαρακτηριστικά (URG Flag Count, Total Fwd Packets, Total Backward Packets, Active Std) που έχουν σχεδόν μηδενική επίδραση.



Εικόνα 88 Διάγραμμα Bar Plot για τον 12ο συνδυασμό.

Για τον δέκατο τρίτο συνδυασμό (Εικόνα 89) παρατηρούμε ότι το Bwd Packet Length Mean κυριαρχεί (≈ 7.0) με ολική συνεισφορά από όλες τις κλάσεις και ειδικότερα από τις κλάσεις 4 και 5. Αυτό δείχνει ότι το μέσο μέγεθος των backward πακέτων αποτελεί τον κυριότερο διαχωριστικό παράγοντα για το σύνολο των κλάσεων. Το Idle Mean (≈ 5.8) ακολουθεί με κύρια συνεισφορά από τις κλάσεις 0 και 2, υποδηλώνοντας ότι ο χρόνος αδράνειας της σύνδεσης είναι ιδιαίτερα χαρακτηριστικός για αυτές τις περιπτώσεις. Τα Subflow Bwd Packets (≈ 5.4) και Bwd Packet Length Std (≈ 5.1) ακολουθούν επίσης με συνολική κατανομή κλάσεων, αναδεικνύοντας ότι τόσο ο αριθμός όσο και η μεταβλητότητα των backward πακέτων είναι σημαντικά χαρακτηριστικά.

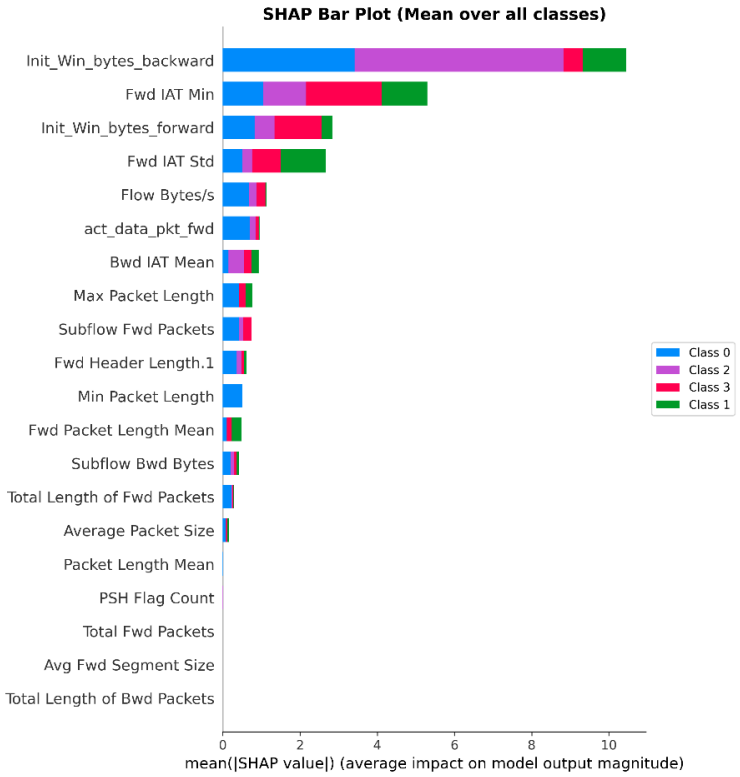
Τα Total Fwd Packets (≈ 5.0) και Fwd Header Length.1 (≈ 4.7) εμφανίζουν επίσης υψηλή συνεισφορά με ισορροπημένη κατανομή κλάσεων, ενώ το Bwd Packet Length Max (≈ 4.3) επηρεάζεται κυρίως από την κλάση 5. Από το Packet Length Mean μέχρι το Total Length of Bwd Packets η συνεισφορά κυμαίνεται μεταξύ 3.6 και 1.9, ενώ τα FIN/SYN Flag Count παρουσιάζουν μέτρια επίδραση (≈ 0.5–0.6) κυρίως από την κλάση 2. Τα τελευταία τρία χαρακτηριστικά (Subflow Bwd Bytes, Total Backward Packets, Avg Bwd Segment Size) έχουν μηδενική επίδραση.



Εικόνα 89 Διάγραμμα Bar Plot για τον 13ο συνδυασμό.

Για τον δέκατο τέταρτο συνδυασμό (Εικόνα 90) παρατηρούμε ότι το Init_Win_bytes_backward κυριαρχεί (≈ 10.5) με κύρια συνεισφορά από τις κλάσεις 0 και 2, ενώ οι κλάσεις 3 και 1 συνεισφέρουν ελάχιστα. Αυτό σημαίνει ότι το αρχικό μέγεθος TCP παραθύρου στην αντίστροφη κατεύθυνση είναι καθοριστικός παράγοντας κυρίως για τη διάκριση των κλάσεων 0 και 2. Το Fwd IAT Min (≈ 5.2) ακολουθεί με ισορροπημένη συνεισφορά από όλες τις κλάσεις, με ιδιαίτερη παρουσία της κλάσης 3, υποδηλώνοντας ότι το ελάχιστο χρονικό διάστημα μεταξύ των προωθούμενων πακέτων λειτουργεί διαγνωστικά για το σύνολο των κλάσεων.

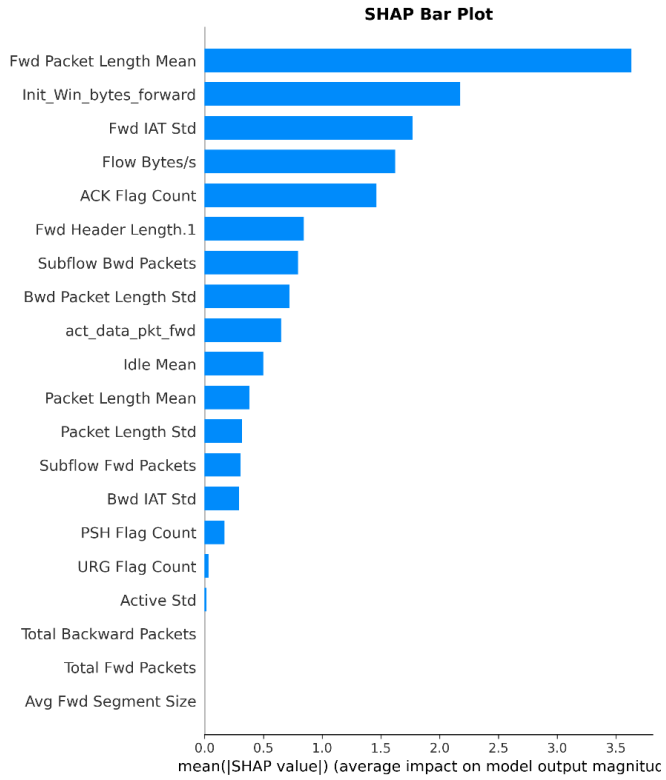
Τα Init_Win_bytes_forward (≈ 2.9) και Fwd IAT Std (≈ 2.8) ακολουθούν με παρόμοια συνεισφορά, με την κλάση 3 να κυριαρχεί στο πρώτο και την κλάση 1 στο δεύτερο, υποδηλώνοντας ότι αυτά τα χαρακτηριστικά διακρίνουν ιδιαίτερα αυτές τις κλάσεις. Από το Flow Bytes/s (≈ 1.0) και κάτω η συνεισφορά πέφτει απότομα, με χαρακτηριστικά όπως τα act_data_pkt_fwd, Bwd IAT Mean, Max Packet Length να παρουσιάζουν μέτρια επίδραση (≈ 0.7–0.9), ενώ τα τελευταία πέντε (Packet Length Mean, PSH Flag Count, Total Fwd Packets, Avg Fwd Segment Size, Total Length of Bwd Packets) έχουν ουσιαστικά μηδενική επίδραση.



Εικόνα 90 Διάγραμμα Bar Plot για τον 14ο συνδυασμό.

Για τον δέκατο πέμπτο συνδυασμό (φαίνεται στην Εικόνα 91) παρατηρούμε ότι το Fwd Packet Length Mean κυριαρχεί (≈ 3.6), δείχνοντας ότι το μέσο μέγεθος των προωθούμενων πακέτων παραμένει ο κύριος διαχωριστικός παράγοντας για τη διάκριση μεταξύ της κανονικής κίνησης από την κακόβουλη. Ακολουθούν τα Init_Win_bytes_forward (≈ 2.15), Fwd IAT Std (≈ 1.75) και Flow Bytes/s (≈ 1.6), σχηματίζοντας μια ισχυρή δευτερεύουσα ομάδα συνεισφοράς. Αξιοσημείωτη είναι η παρουσία του ACK Flag Count (≈ 1.5) στην πέμπτη θέση, υποδηλώνοντας ότι ο αριθμός των ACK σημαιών αποτελεί σημαντικό δείκτη ανίχνευσης επίθεσης.

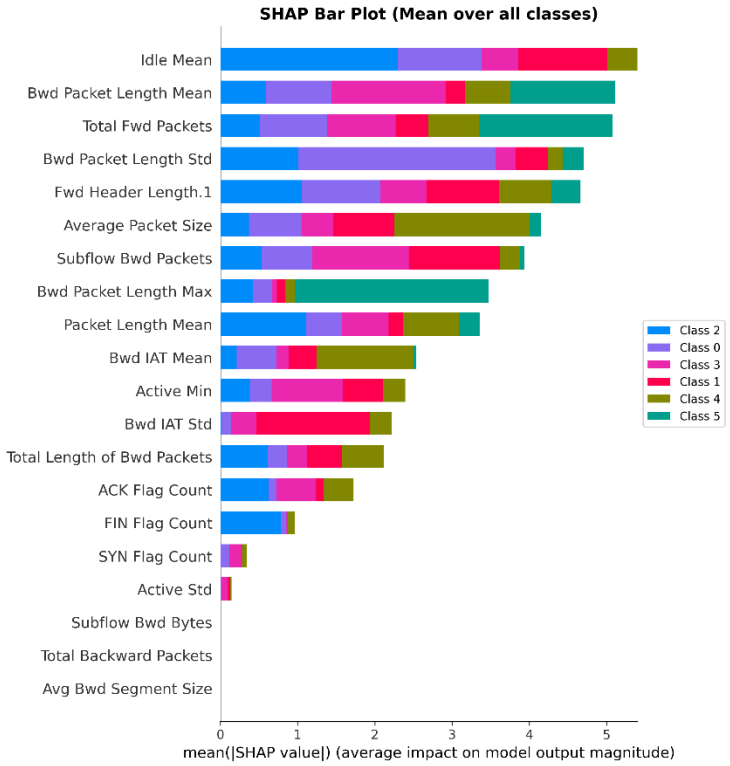
Από Fwd Header Length.1 (≈ 0.82) μέχρι Bwd IAT Std (≈ 0.28) παρατηρείται σταδιακή και ομαλή μείωση της συνεισφοράς, με χαρακτηριστικά όπως τα Subflow Bwd Packets, Bwd Packet Length Std, act_data_pkt_fwd και Idle Mean να διατηρούν αξιόλογη αλλά δευτερεύουσα επίδραση. Το PSH Flag Count (≈ 0.17) έχει οριακή συνεισφορά, ενώ τα τελευταία τέσσερα χαρακτηριστικά (URG Flag Count, Active Std, Total Backward Packets, Total Fwd Packets, Avg Fwd Segment Size) έχουν σχεδόν μηδενική επίδραση.



Εικόνα 91 Διάγραμμα Bar Plot για τον 15ο συνδυασμό.

Για τον δέκατο έκτο συνδυασμό (Εικόνα 92) παρατηρούμε ότι το Idle Mean (≈ 5.3) κυριαρχεί με συνεισφορά από όλες τις κλάσεις, με ιδιαίτερη έμφαση στις κλάσεις 1 και 2, υποδηλώνοντας ότι ο χρόνος αδράνειας της σύνδεσης αποτελεί τον κυριότερο διαχωριστικό παράγοντα για αυτόν τον συνδυασμό. Τα Bwd Packet Length Mean (≈ 5.1) και Total Fwd Packets (≈ 5.1) ακολουθούν με σχεδόν ισάξια συνεισφορά, με το Bwd Packet Length Mean να επηρεάζεται κυρίως από τις κλάσεις 3 και 5, ενώ το Total Fwd Packets παρουσιάζει ισορροπημένη κατανομή από όλες τις κλάσεις. Αυτό σημαίνει ότι τόσο το μέγεθος των backward πακέτων όσο και ο αριθμός των forward πακέτων είναι εξίσου σημαντικά.

Τα Bwd Packet Length Std (≈ 4.7) και Fwd Header Length.1 (≈ 4.6) εμφανίζουν επίσης υψηλή συνεισφορά με συνολική κατανομή κλάσεων. Αξιοσημείωτη είναι η παρουσία του Bwd Packet Length Max (≈ 3.5) με κυρίαρχη συνεισφορά από την κλάση 5, γεγονός που δείχνει ότι αυτή η κλάση παρουσιάζει ιδιαίτερα μεγάλα backward πακέτα. Από Packet Length Mean μέχρι ACK Flag Count η συνεισφορά κυμαίνεται μεταξύ 1.6 και 3.3, ενώ τα FIN/SYN Flag Count και Active Std παρουσιάζουν αρκετά χαμηλή επίδραση (< 0.4). Τα τελευταία τρία χαρακτηριστικά (Subflow Bwd Bytes, Total Backward Packets, Avg Bwd Segment Size) έχουν μηδενική επίδραση.

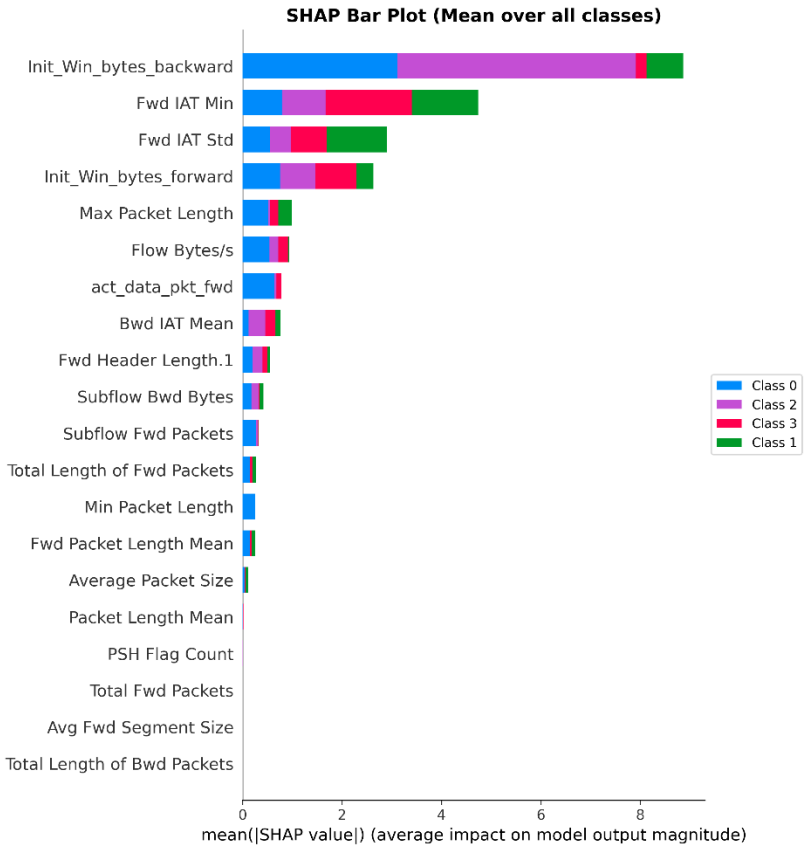


Εικόνα 92 Διάγραμμα Bar Plot για τον 16ο συνδυασμό.

Στον δέκατο έβδομο συνδυασμό (Εικόνα 93) παρατηρούμε ότι κυρίαρχο είναι και πάλι το Init_Win_bytes_backward (≈ 8.7) με βασικότερη συνεισφορά από τις κλάσεις 0 και 2, ενώ οι κλάσεις 3 και 1 συνεισφέρουν ελάχιστα, δείχνοντας ότι το αρχικό μέγεθος TCP παραθύρου στην αντίστροφη κατεύθυνση είναι καθοριστικός παράγοντας για τη διάκριση των κλάσεων 0 και 2.

Το Fwd IAT Min (≈ 4.8) ακολουθεί με ισορροπημένη συνεισφορά από όλες τις κλάσεις, υποδηλώνοντας ότι το ελάχιστο χρονικό διάστημα μεταξύ προωθούμενων πακέτων είναι γενικά διαγνωστικό για το σύνολο των κλάσεων. Τα Fwd IAT Std (≈ 3.0) και Init_Win_bytes_forward (≈ 2.8) ακολουθούν με παρόμοια συνεισφορά και ισορροπημένη κατανομή κλάσεων.

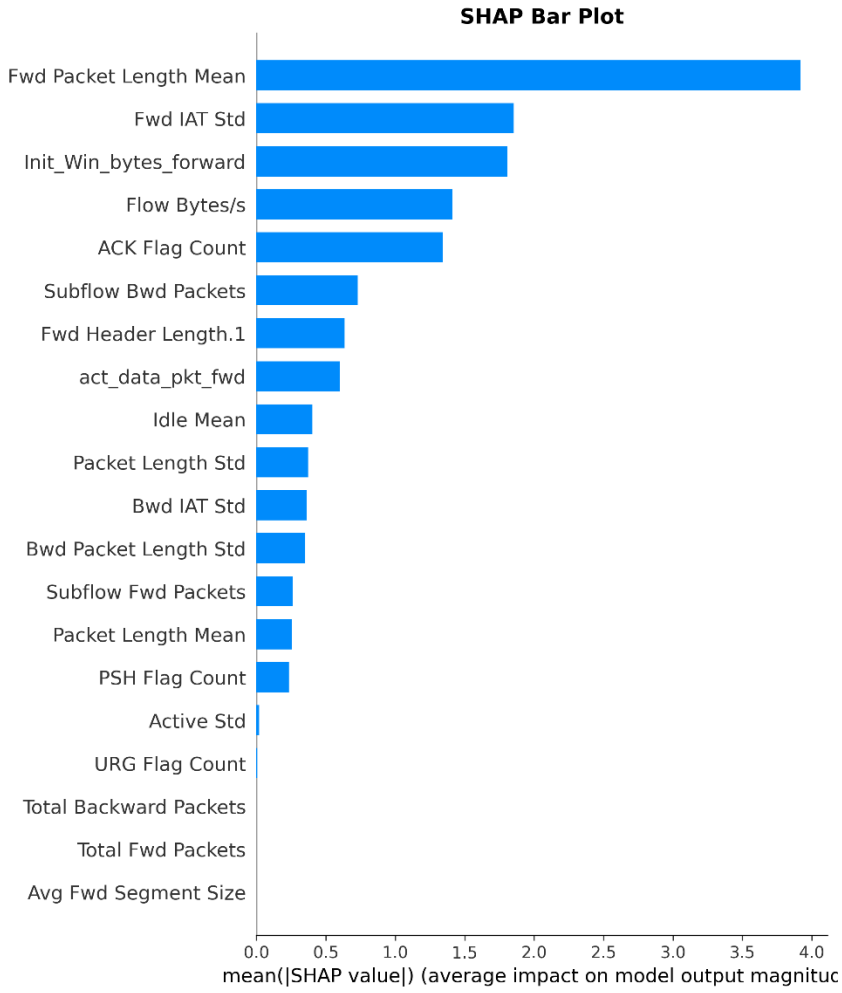
Από Max Packet Length (≈ 1.0) και κάτω η συνεισφορά πέφτει απότομα, με χαρακτηριστικά όπως τα Flow Bytes/s, act_data_pkt_fwd, Bwd IAT Mean να παρουσιάζουν μέτρια επίδραση (≈ 0.7–1.0) και με τα τελευταία πέντε (Packet Length Mean, PSH Flag Count, Total Fwd Packets, Avg Fwd Segment Size, Total Length of Bwd Packets) να έχουν ουσιαστικά μηδενική επίδραση.



Εικόνα 93 Διάγραμμα Bar Plot για τον 17ο συνδυασμό.

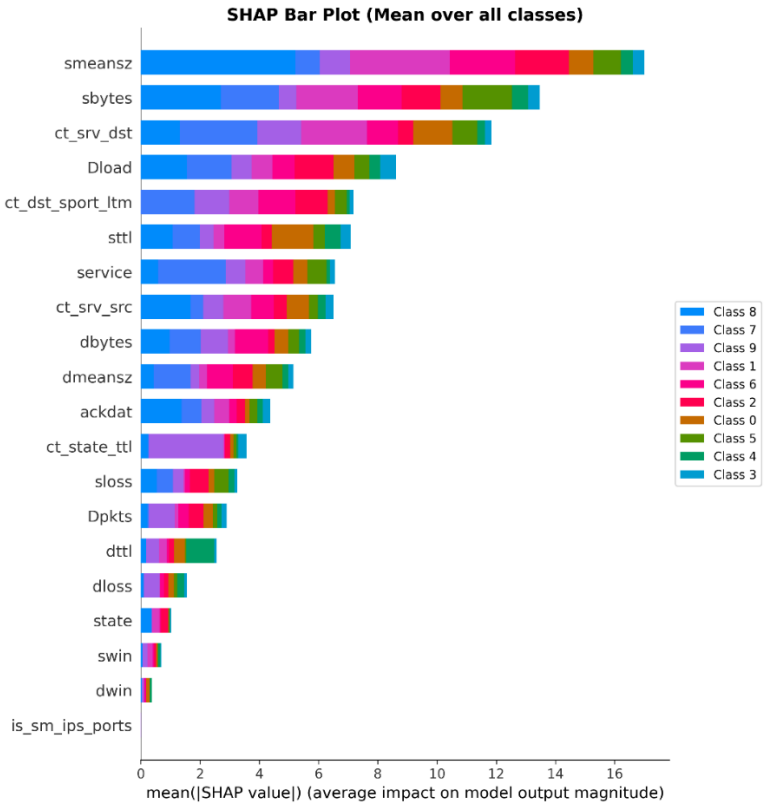
Για τον δέκατο όγδοο συνδυασμό (βλέπε Εικόνα 94) παρατηρούμε ότι επικρατεί το Fwd Packet Length Mean (≈ 3.95), υποδηλώνοντας ότι το μέσο μέγεθος των προωθούμενων πακέτων είναι ο κύριος διαχωριστικός παράγοντας για τη διάκριση κανονικής κίνησης από επίθεση. Τα Fwd IAT Std (≈ 1.85) και Init_Win_bytes_forward (≈ 1.82) ακολουθούν με σχεδόν ισάξια συνεισφορά, σχηματίζοντας μια ισχυρή δευτερεύουσα ομάδα μαζί με το Flow Bytes/s (≈ 1.4) και το ACK Flag Count (≈ 1.35). Τα πέντε αυτά χαρακτηριστικά μαζί αποτελούν τον πυρήνα λήψης απόφασης του μοντέλου.

Από το Subflow Bwd Packets (≈ 0.7) μέχρι το PSH Flag Count (≈ 0.2) παρατηρείται σταδιακή και ομαλή μείωση της συνεισφοράς, με χαρακτηριστικά όπως τα Fwd Header Length.1, act_data_pkt_fwd, Idle Mean να διατηρούν αξιόλογη αλλά δευτερεύουσα επίδραση. Τα τελευταία τέσσερα χαρακτηριστικά (Active Std, URG Flag Count, Total Backward Packets, Total Fwd Packets, Avg Fwd Segment Size) έχουν σχεδόν μηδενική επίδραση.



Εικόνα 94 Διάγραμμα Bar Plot για τον 18ο συνδυασμό.

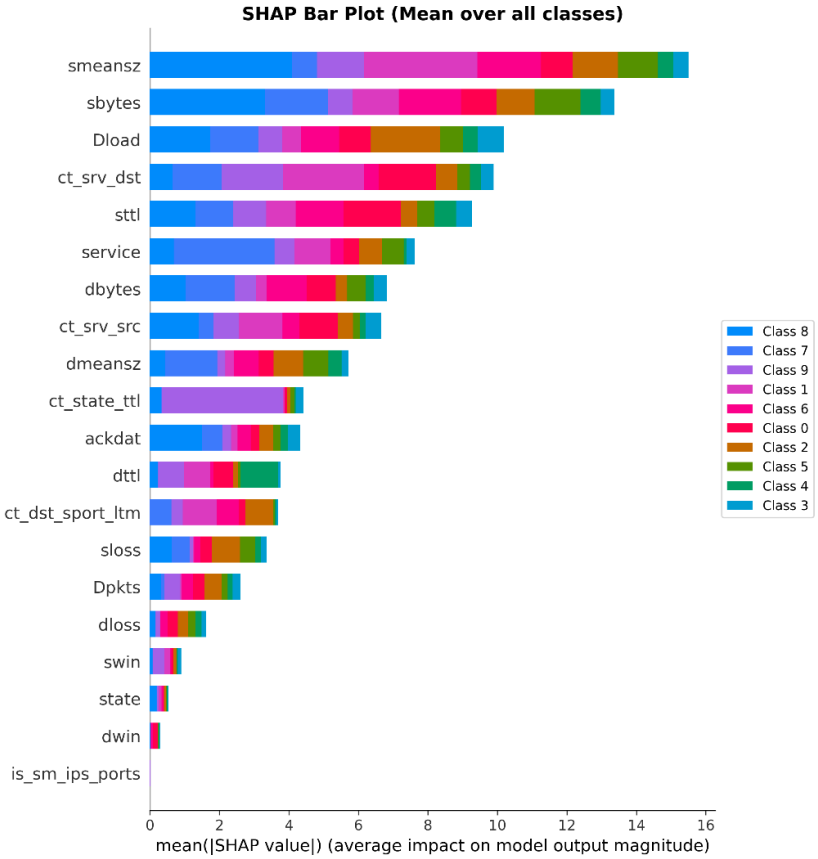
Για τον δέκατο ένατο συνδυασμό (φαίνεται στην Εικόνα 95) παρατηρούμε ότι το smeansz (≈ 17.0) επικρατεί με συνεισφορά από όλες τις κλάσεις, με ιδιαίτερη έμφαση στις 8, 9 και 6. Το smeansz αντιπροσωπεύει το μέσο μέγεθος των πακέτων πηγής, αποτελώντας τον κυριότερο διαχωριστικό παράγοντα για το σύνολο των κλάσεων. Ακολουθούν τα sbytes (≈ 13.5) και ct_srv_dst (≈ 11.8), τα οποία αντιπροσωπεύουν αντίστοιχα τα bytes πηγής και τον αριθμό συνδέσεων προς τον server προορισμού. Η υψηλή συνεισφορά του ct_srv_dst σημαίνει ότι η συμπεριφορά επανάληψης των συνδέσεων προς τον ίδιο προορισμό αποτελεί χαρακτηριστικό μοτίβο επίθεσης. Το Dload (≈ 8.7) και τα ct_dst_sport_ltm, sttl (≈ 7.2) παρουσιάζουν επίσης υψηλή συνεισφορά, με τα service, ct_srv_src, dbytes, dmeansz να ακολουθούν (5.0–6.5). Από ackdat μέχρι dttl η συνεισφορά κυμαίνεται μεταξύ 4.3 και 2.5, ενώ τα dloss, state, swin, dwin παρουσιάζουν χαμηλή επίδραση (< 1.5). Το is_sm_ips_ports έχει μηδενική επίδραση.



Εικόνα 95 Διάγραμμα Bart Plot για τον 19ο συνδυασμό.

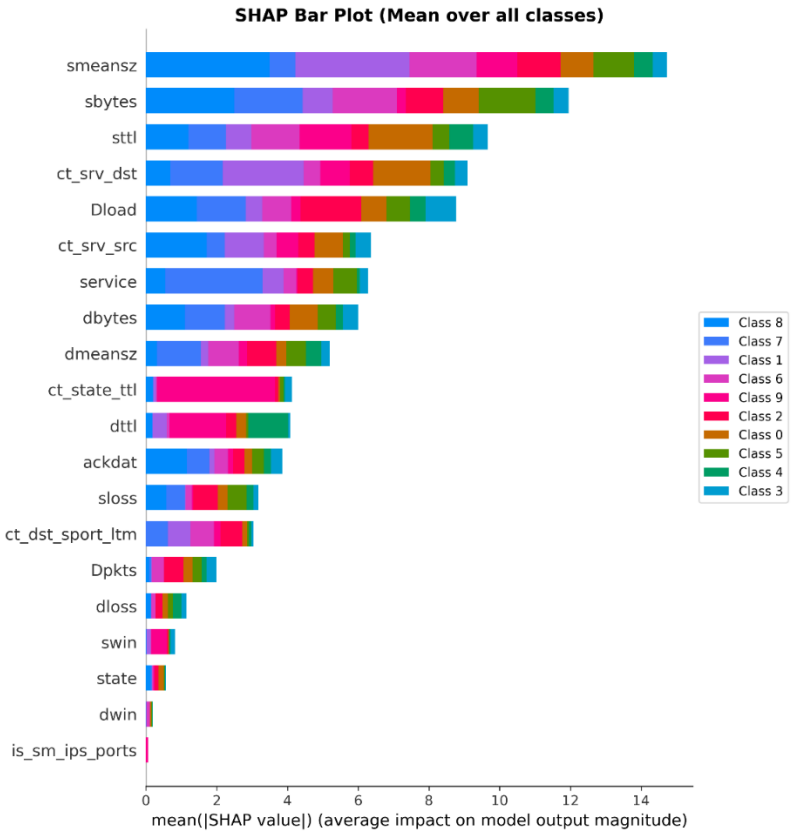
Στον εικοστό συνδυασμό (Εικόνα 96) παρατηρούμε ότι το smeansz (≈ 15.5) κυριαρχεί και πάλι με συνεισφορά από όλες τις κλάσεις, επιβεβαιώνοντας ότι το μέσο μέγεθος των πακέτων πηγής αποτελεί τον κυριότερο διαχωριστικό παράγοντα για αυτό το σύνολο δεδομένων. Ακολουθούν τα sbytes (≈ 13.3) και Dload (≈ 10.2), με το Dload να ανεβαίνει στην τρίτη θέση σε αυτόν τον συνδυασμό, υποδηλώνοντας ότι εδώ ο ρυθμός λήψης δεδομένων προορισμού παίζει πιο σημαντικό ρόλο. Το ct_srv_dst (≈ 10.0) και sttl (≈ 9.3) ακολουθούν με υψηλή συνεισφορά από αρκετές κλάσεις.

Το service (≈ 7.4) παρουσιάζει ισχυρή συνεισφορά κυρίως από την κλάση 7 (σκούρο μπλε), δείχνοντας ότι ο τύπος υπηρεσίας είναι ιδιαίτερα χαρακτηριστικός για αυτές τις κλάσεις. Τα dbytes, ct_srv_src και dmeansz εμφανίζουν επίσης αξιόλογη συνεισφορά (5.6–6.9), από το ct_state_ttl (≈ 4.2) μέχρι το Dpkts (≈ 2.5) η συνεισφορά μειώνεται σταδιακά, ενώ τα dloss, swin, state, dwin παρουσιάζουν χαμηλή επίδραση (< 1.2). Το is_sm_ips_ports έχει μηδενική επίδραση.



Εικόνα 96 Διάγραμμα Bar Plot για τον 20ο συνδυασμό.

Για τον εικοστό πρώτο συνδυασμό (Εικόνα 97) παρατηρούμε ότι και εδώ το smeansz (≈ 14.8) επικρατεί με συνεισφορά από όλες τις κλάσεις, επιβεβαιώνοντας ότι το μέσο μέγεθος των πακέτων πηγής αποτελεί σταθερά τον κυριότερο διαχωριστικό παράγοντα για αυτό το σύνολο δεδομένων. Το sbytes (≈ 12.0) ακολουθεί δεύτερο, ενώ αξιοσημείωτη είναι η άνοδος του sttl (≈ 9.7) στην τρίτη θέση με το Time-To-Live των πακέτων πηγής να αποκτά μεγαλύτερη σημασία σε αυτόν τον συνδυασμό. Το ct_srv_dst (≈ 9.0) και Dload (≈ 8.7) ακολουθούν με ισορροπημένη κατανομή κλάσεων. Τα ct_srv_src (≈ 6.4), service (≈ 6.3) και dbytes (≈ 6.0) παρουσιάζουν παρόμοια συνεισφορά, ενώ το dmeansz (≈ 5.2) ολοκληρώνει την ομάδα των σημαντικών χαρακτηριστικών. Από ct_state_ttl (≈ 4.1) μέχρι ct_dst_sport_ltm (≈ 3.1) η συνεισφορά παραμένει αξιόλογη, ενώ τα Dpkts, dloss, swin, state παρουσιάζουν χαμηλή επίδραση (< 2.0). Το dwin (≈ 0.15) και is_sm_ips_ports (≈ 0.05) έχουν σχεδόν μηδενική επίδραση.



Εικόνα 97 Διάγραμμα Bar Plot για τον 21ο συνδυασμό.

6.3.2. Ερμηνεία προβλέψεων

Για την ερμηνεία των προβλέψεων του μοντέλου, η εφαρμογή παράγει το συγκεντρωτικό SHAP Summary Plot για όλες τις κλάσεις μαζί αλλά και ξεχωριστό για κάθε κλάση, στα οποία απεικονίζεται όχι μόνο ποια χαρακτηριστικά είναι σημαντικά, αλλά και προς ποια κατεύθυνση επηρεάζουν τις αποφάσεις του μοντέλου.

Το SHAP Summary Plot συγκεντρώνει τις SHAP τιμές όλων των δειγμάτων για κάθε χαρακτηριστικό. Στον οριζόντιο άξονα παρατίθενται τα χαρακτηριστικά ταξινομημένα κατά φθίνουσα σειρά με βάση τη συνολική τους σημασία, η οποία υπολογίζεται ως η μέση απόλυτη τιμή SHAP. Ο κάθετος άξονας δείχνει την τιμή SHAP κάθε δείγματος, θετική τιμή (δεξιά) σημαίνει ότι το χαρακτηριστικό αυξάνει την πιθανότητα ταξινόμησης ως επίθεση, ενώ αρνητική τιμή (αριστερά) τη μειώνει. Η χρωματική γραμμή (κόκκινη για υψηλή και μπλε για χαμηλή) αποκαλύπτει τη συσχέτιση μεταξύ της αρχικής τιμής του χαρακτηριστικού και της επίδρασής του, με κάθε κουκκίδα να αντιπροσωπεύει μια παρουσία δεδομένων [22]. Έτσι για παράδειγμα, ένα χαρακτηριστικό με κόκκινα σημεία

δεξιά του μηδενός σημαίνει ότι υψηλές τιμές του αυξάνουν την πιθανότητα ανίχνευσης επίθεσης.

Ακολουθεί ανάλυση των είκοσι ένα συνδυασμών όπως αναφέρει ο Πίνακας 6.

Στον πρώτο συνδυασμό παρατηρούμε ότι στο συνολικό Summary Plot (Εικόνα 98), το Fwd Header Length.1 κυριαρχεί με τα περισσότερα σημεία αριστερά του μηδενός με κυρίως μπλε χρώμα, υποδηλώνοντας ότι χαμηλές τιμές της επικεφαλίδας των προωθούμενων πακέτων μειώνουν συνολικά την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Bwd Packet Length Max παρουσιάζει διπλό μοτίβο με μπλε σημεία αριστερά και κόκκινα δεξιά, που σημαίνει ότι τόσο οι πολύ χαμηλές όσο και οι πολύ υψηλές τιμές οδηγούν σε αναγνώριση επίθεσης. Το Idle Mean με κόκκινα σημεία δεξιά σημαίνει ότι υψηλές τιμές αδράνειας αυξάνουν συνολικά την πιθανότητα αναγνώρισης κακόβουλης κίνησης.

Για την κλάση 0 (Εικόνα 99) η οποία είναι η κανονική (benign) το Bwd Packet Length Std κυριαρχεί με κόκκινα/μοβ σημεία αριστερά (≈ -6), που σημαίνει ότι υψηλή μεταβλητότητα στο μέγεθος backward πακέτων μειώνει την πιθανότητα αναγνώρισης της κίνησης ως κανονική το οποίο είναι λογικό, καθώς η κανονική κίνηση τείνει να έχει σταθερό μέγεθος πακέτων. Τα Total Fwd Packets και Subflow Bwd Packets με κόκκινα σημεία δεξιά υποδηλώνουν ότι οι υψηλές τιμές τους αυξάνουν την πιθανότητα αναγνώρισης ως κανονική κίνηση. Το Idle Mean με κόκκινα σημεία αριστερά υποδηλώνει ότι υψηλή αδράνεια μειώνει την πιθανότητα αναγνώρισης ως BENIGN.

Για την κλάση 1 (Εικόνα 100) η οποία είναι η DoS GoldenEye το Idle Mean παρουσιάζει το πιο εκτεταμένο εύρος με κόκκινα σημεία δεξιά (έως +6.5), επιβεβαιώνοντας ότι υψηλός χρόνος αδράνειας αυξάνει σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Bwd IAT Std με μπλε σημεία δεξιά σημαίνει ότι χαμηλή μεταβλητότητα backward IAT αυξάνει την πιθανότητα αναγνώρισης ως DoS GoldenEye. Τα Subflow Bwd Packets και Fwd Header Length.1 με μπλε σημεία αριστερά υποδηλώνουν ότι χαμηλές τιμές τους μειώνουν την πιθανότητα αναγνώρισης της συγκεκριμένης κλάσης.

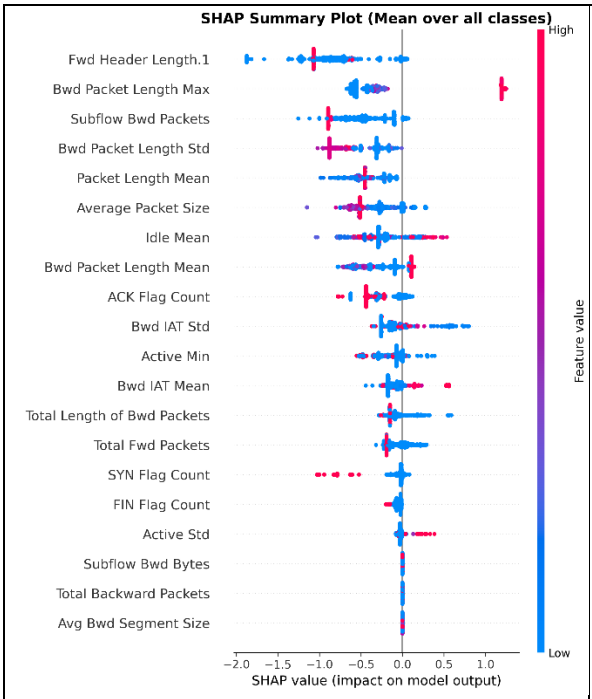
Για την κλάση 2 (Εικόνα 101) η οποία αντιστοιχεί στο είδος επίθεσης DoS Hulk, το Idle Mean κυριαρχεί με κόκκινα σημεία δεξιά (έως +5), επιβεβαιώνοντας ότι υψηλή αδράνεια αυξάνει την πιθανότητα αναγνώρισης κίνησης ως κακόβουλη. Το Bwd Packet Length Max παρουσιάζει μπλε σημεία δεξιά (έως +4), δηλαδή χαμηλές τιμές του αυξάνουν την πιθανότητα αναγνώρισης της κλάσης υποδηλώνοντας ότι η επίθεση παράγει μικρά

backward πακέτα. Το SYN Flag Count με κόκκινα σημεία αριστερά δείχνει ότι υψηλός αριθμός SYN σημαιών μειώνει την πιθανότητα αναγνώρισης της κλάσης, διαφοροποιώντας την από επιθέσεις τύπου SYN flood.

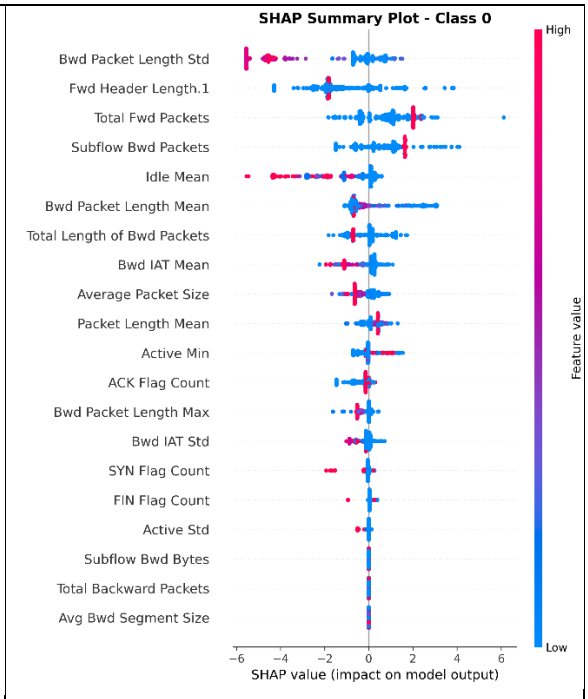
Για την κλάση 3 (Εικόνα 102) η οποία αντιστοιχεί στο είδος επίθεσης DoS Slowhttptest το Active Min παρουσιάζει κόκκινα σημεία δεξιά του κεντρικού άξονα (έως +4.5), υποδηλώνοντας ότι υψηλές τιμές ελάχιστου χρόνου ενεργοποίησης αυξάνουν την πιθανότητα αναγνώρισης της κλάσης και άρα, της κίνησης ως επίθεση. Το Bwd IAT Std με κόκκινα σημεία δεξιά (έως +4) δείχνει ότι η υψηλή μεταβλητότητα backward IAT αυξάνει επίσης την πιθανότητα αναγνώρισής της. Τα Subflow Bwd Packets και Fwd Header Length.1 με μπλε σημεία αριστερά υποδηλώνουν ότι χαμηλές τιμές τους μειώνουν την πιθανότητα αναγνώρισης της κίνησης ως DoS Slowhttptest.

Για την κλάση 4 (Εικόνα 103) η οποία αντιστοιχεί στην DoS Slowloristo Bwd IAT Mean με κόκκινα σημεία δεξιά (έως +4) σημαίνει ότι ο υψηλός μέσος χρόνος backward IAT αυξάνει την πιθανότητα αναγνώρισης της συγκεκριμένης κλάσης. Τα Bwd Packet Length Mean και Average Packet Size παρουσιάζουν διπολικό μοτίβο, υποδηλώνοντας ότι τόσο πολύ χαμηλές όσο και πολύ υψηλές τιμές τους αυξάνουν την πιθανότητα αναγνώρισης ως DoS Slowloris. Η ACK Flag Count με κόκκινα σημεία αριστερά υποδηλώνει ότι υψηλός αριθμός ACK flags μειώνει την πιθανότητα αναγνώρισής της.

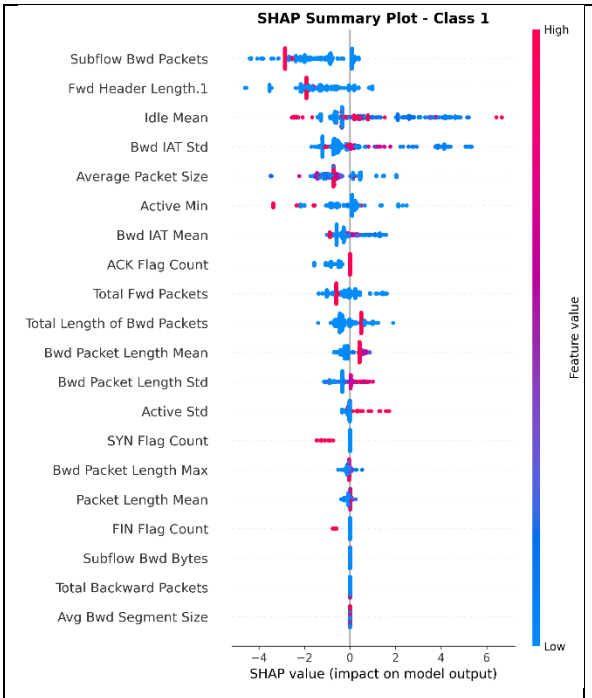
Για την κλάση 5 (Εικόνα 104) η οποία αντιστοιχεί στην Heartbleed) το Bwd Packet Length Max παρουσιάζει εξαιρετικά διπλό μοτίβο το οποίο είναι μπλε σημεία αριστερά (≈ -2.5) και κόκκινα δεξιά (≈ +6) υποδηλώνοντας ότι τόσο οι πολύ χαμηλές όσο και οι πολύ υψηλές τιμές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Τα υπόλοιπα χαρακτηριστικά εμφανίζουν πολύ μικρό εύρος SHAP τιμών (< ±1.5), υποδηλώνοντας ότι το μοντέλο βασίζεται σχεδόν αποκλειστικά στο Bwd Packet Length Max για την αναγνώριση Heartbleed.



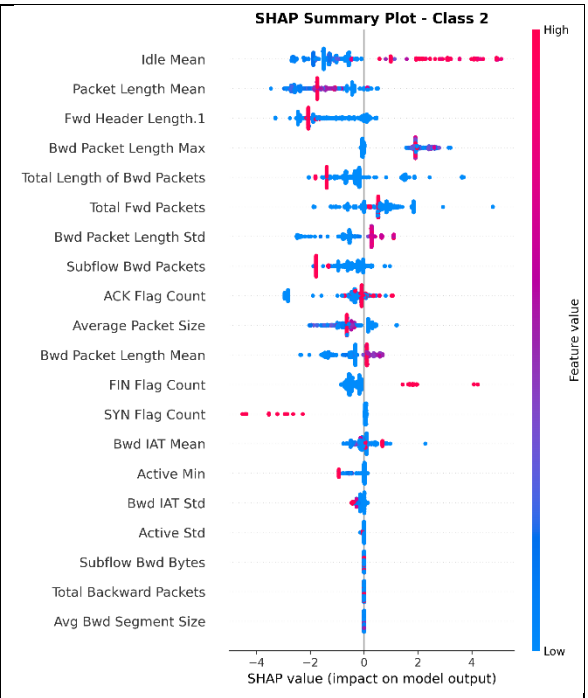
Εικόνα 98 SHAP Summary Plot, όλες οι κλάσεις για τον 1ο συνδυασμό.



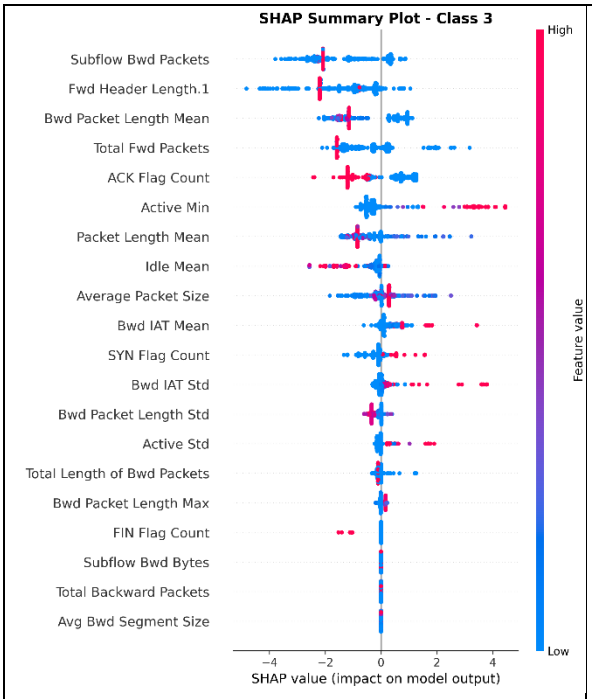
Εικόνα 99 SHAP Summary Plot, κλάση 0 για τον 1ο συνδυασμό.



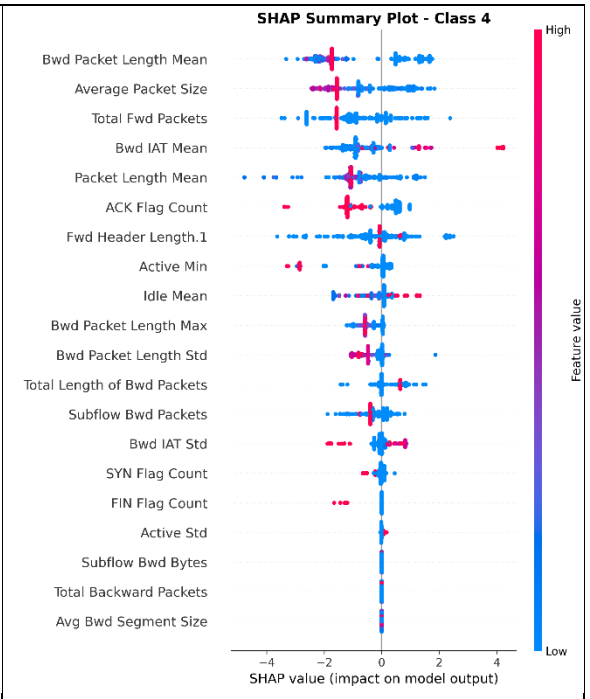
Εικόνα 100 SHAP Summary Plot, κλάση 1 για τον 1ο συνδυασμό.



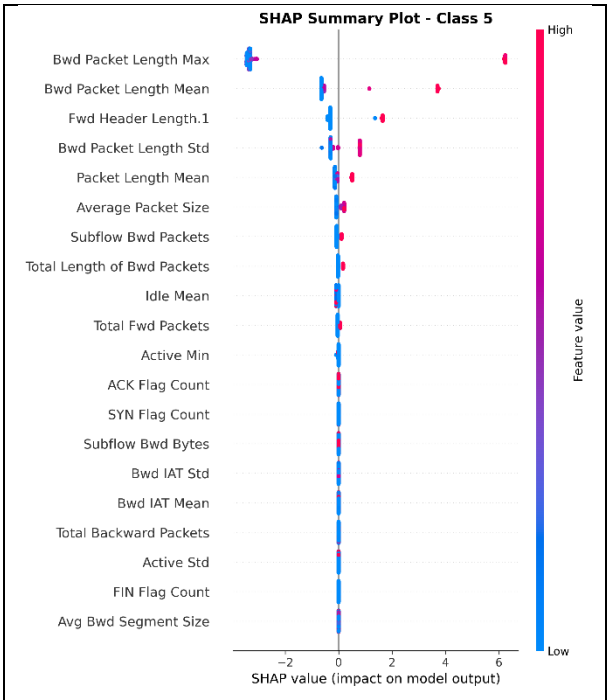
Εικόνα 101 SHAP Summary Plot, κλάση 2 για τον 1ο συνδυασμό.



Εικόνα 102 SHAP Summary Plot, κλάση 3 για τον 1ο συνδυασμό.



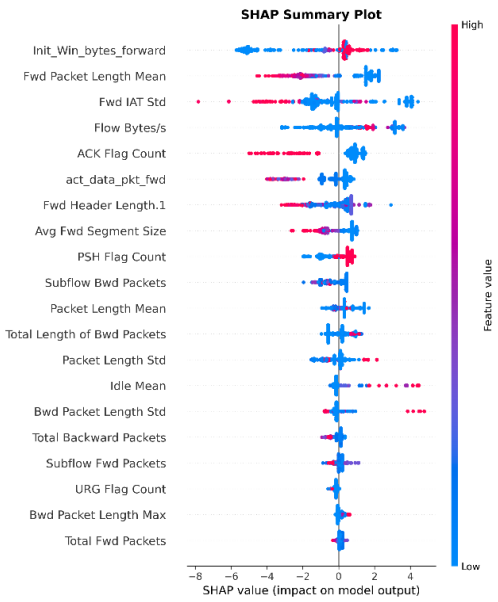
Εικόνα 103 SHAP Summary Plot, κλάση 4 για τον 1ο συνδυασμό.



Εικόνα 104 SHAP Summary Plot, κλάση 5 για τον 1ο συνδυασμό.

Για τον δεύτερο συνδυασμό παρατηρούμε ότι στο συνολικό Summary Plot (φαίνεται στην Εικόνα 105) το Init_Win_bytes_forward κυριαρχεί με το μεγαλύτερο εύρος SHAP τιμών. Τα μπλε σημεία εκτείνονται αριστερά (έως -8), υποδηλώνοντας ότι χαμηλές τιμές TCP παραθύρου μειώνουν την πιθανότητα αναγνώρισης κακόβουλης κίνησης, ενώ τα κόκκινα σημεία δεξιά (έως +1.5) υποδηλώνουν ότι υψηλές τιμές ελαφρώς αυξάνουν αυτή την πιθανότητα. Το Fwd Packet Length Mean εμφανίζει κόκκινα σημεία αριστερά και μπλε δεξιά (έως +1.5), υποδηλώνοντας διπολικό μοτίβο — χαμηλές τιμές αυξάνουν και υψηλές μειώνουν την πιθανότητα αναγνώρισης επίθεσης. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (έως -4), υποδηλώνοντας ότι υψηλή μεταβλητότητα χρονικών διαστημάτων προωθούμενων πακέτων μειώνει την πιθανότητα αναγνώρισης επίθεσης, αντίθετα δηλαδή από το αναμενόμενο, υποδηλώνοντας ότι η κανονική κίνηση έχει πιο ακανόνιστους χρόνους αποστολής.

Το ACK Flag Count εμφανίζει κόκκινα σημεία αριστερά (έως -5), υποδηλώνοντας ότι υψηλός αριθμός ACK σημαιών μειώνει την πιθανότητα αναγνώρισης επίθεσης. Το Flow Bytes/s με μοβ/κόκκινα σημεία δεξιά υποδηλώνει ότι υψηλός ρυθμός bytes αυξάνει την πιθανότητα αναγνώρισης. Τα Idle Mean και Bwd Packet Length Std παρουσιάζουν κόκκινα σημεία δεξιά (έως +4-5), υποδηλώνοντας ότι υψηλές τιμές αδράνειας και μεταβλητότητας backward πακέτων αυξάνουν σημαντικά την πιθανότητα αναγνώρισης επίθεσης. Τα τελευταία features (Bwd Packet Length Max, Total Fwd Packets) εμφανίζουν σχεδόν μηδενική επίδραση.



Εικόνα 105 SHAP Summary Plot, όλες οι κλάσεις για τον 2ο συνδυασμό.

Για τον τρίτο συνδυασμό, παρατηρούμε ότι στο συνολικό Summary Plot (όπως δείχνει η Εικόνα 106) το Init_Win_bytes_backward κυριαρχεί με εξαιρετικά χαρακτηριστικό μοτίβο στα κόκκινα σημεία αριστερά (≈ -3.5) γεγονός που σημαίνει ότι οι υψηλές τιμές TCP backward παραθύρου μειώνουν συνολικά την πιθανότητα αναγνώρισης κακόβουλης κίνησης, ενώ τα μπλε σημεία δεξιά υποδηλώνουν ότι χαμηλές τιμές την αυξάνουν. Το Fwd IAT Min εμφανίζει πυκνή συγκέντρωση σημείων σε μικρό εύρος γύρω από το μηδέν, υποδηλώνοντας ομοιόμορφη αλλά μέτρια επίδραση. Τα Init_Win_bytes_forward και Bwd IAT Mean με κόκκινα σημεία αριστερά υποδηλώνουν ότι υψηλές τιμές τους μειώνουν την πιθανότητα αναγνώρισης επίθεσης.

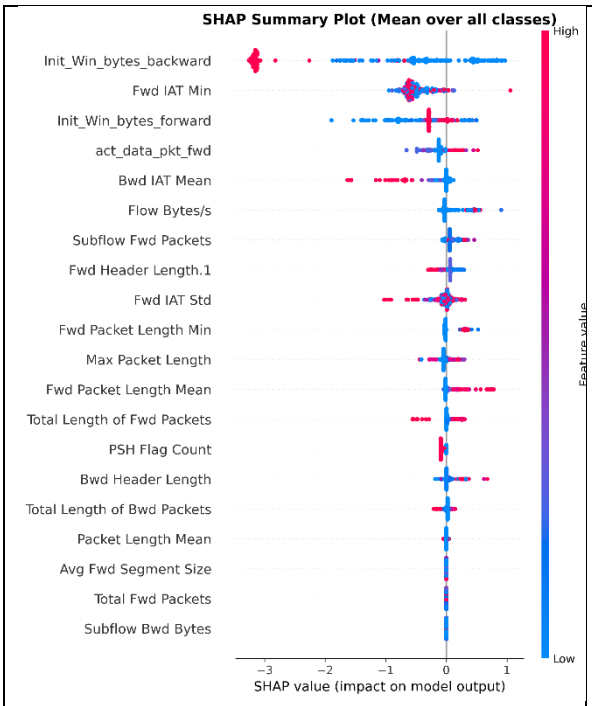
Για την κλάση 0 που είναι η κανονική κίνηση (φαίνεται στην Εικόνα 107) το Init_Win_bytes_backward παρουσιάζει κόκκινα σημεία αριστερά (≈ -5.5) και μπλε σημεία δεξιά (έως +7.5), υποδηλώνοντας ότι χαμηλές τιμές TCP backward παραθύρου αυξάνουν δραστικά την πιθανότητα αναγνώρισης ως BENIGN, ενώ υψηλές τιμές την μειώνουν. Το Fwd IAT Min με συσσωρευμένα μπλε σημεία δεξιά δείχνει ότι χαμηλές τιμές ελάχιστου χρονικού διαστήματος αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Init_Win_bytes_forward παρουσιάζει διπολικό μοτίβο με σημεία και στις δύο κατευθύνσεις.

Για την κλάση 1 που αντιστοιχεί στην Web Attack Brute Force (παρουσιάζεται στην Εικόνα 108) το Fwd IAT Min κυριαρχεί με εξαιρετικά πυκνή συγκέντρωση σημείων αριστερά (έως -4.5) κυρίως μπλε χρώματος, υποδηλώνοντας ότι χαμηλές τιμές ελάχιστου χρονικού διαστήματος μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd IAT Std παρουσιάζει διπολικό μοτίβο με κόκκινα σημεία δεξιά (έως +3.5), υποδηλώνοντας ότι υψηλή μεταβλητότητα χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης. Το Fwd Packet Length Mean με κόκκινα σημεία δεξιά (έως +3.5) δείχνει ότι το υψηλό μέσο μέγεθος προωθούμενων πακέτων αυξάνει την πιθανότητα αναγνώρισης αυτής της επίθεσης.

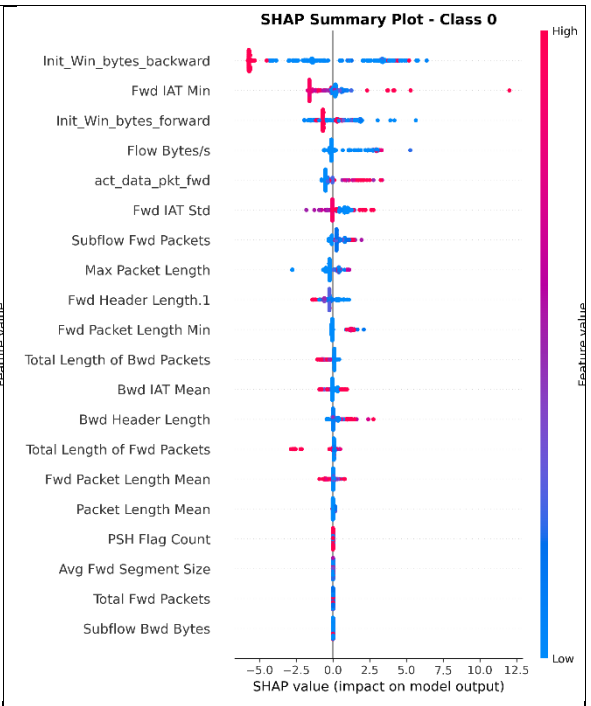
Για την κλάση 2 που αντιστοιχεί στην Web Attack SQL Injection (όπως δείχνει η Εικόνα 109) το Init_Win_bytes_backward παρουσιάζει το πιο εκτεταμένο εύρος με μπλε σημεία αριστερά (≈ -8) και μπλε σημεία δεξιά (έως +7) υποδηλώνοντας ότι χαμηλές τιμές TCP backward παραθύρου λειτουργούν και στις δύο κατευθύνσεις, αποκαλύπτοντας μια πιο πολύπλοκη σχέση με την κλάση. Το Fwd IAT Min με πυκνά μπλε σημεία γύρω από το μηδέν δείχνει ομοιόμορφη αλλά μικρή επίδραση. Τα υπόλοιπα χαρακτηριστικά εμφανίζουν

πολύ μικρό εύρος SHAP τιμών (< ±1.5), γεγονός που υποδεικνύει ότι η SQL Injection δεν έχει ισχυρά διαχωριστικά χαρακτηριστικά πέραν του TCP παραθύρου.

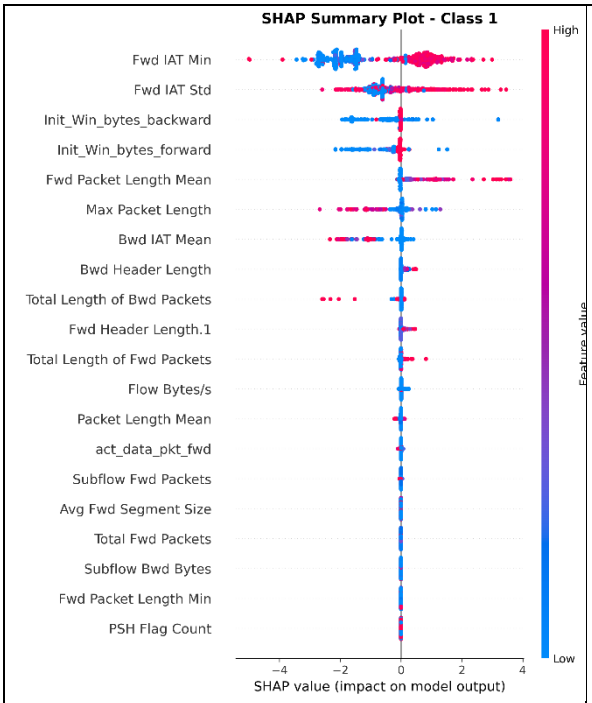
Για την κλάση 3 που είναι η Web Attack XSS (Εικόνα 110) το Init_Win_bytes_backward εμφανίζει κόκκινα σημεία δεξιά (έως +1.0) και μπλε σημεία αριστερά (≈ -3), υποδηλώνοντας ότι υψηλές τιμές TCP backward παραθύρου έχουν ήπια θετική επίδραση στην αναγνώριση XSS, σε αντίθεση με τις πιο χαμηλές. Το Init_Win_bytes_forward με μπλε σημεία αριστερά (έως -6) δείχνει ότι υψηλές τιμές TCP forward παραθύρου μειώνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης, ενώ το Fwd IAT Std με κόκκινα σημεία δεξιά (έως +2.5) υποδηλώνει ότι υψηλή μεταβλητότητα χρονικών διαστημάτων αυξάνει την πιθανότητα.



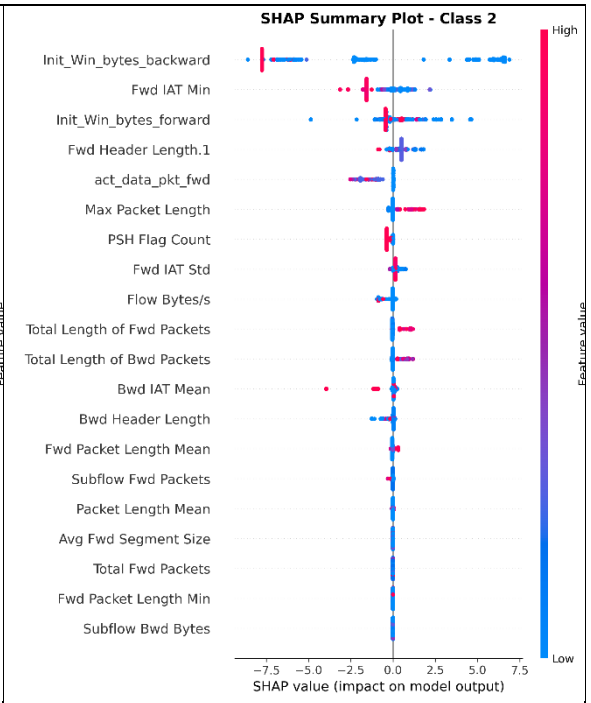
Εικόνα 106 SHAP Summary Plot, όλες οι κλάσεις για τον 3ο συνδυασμό.



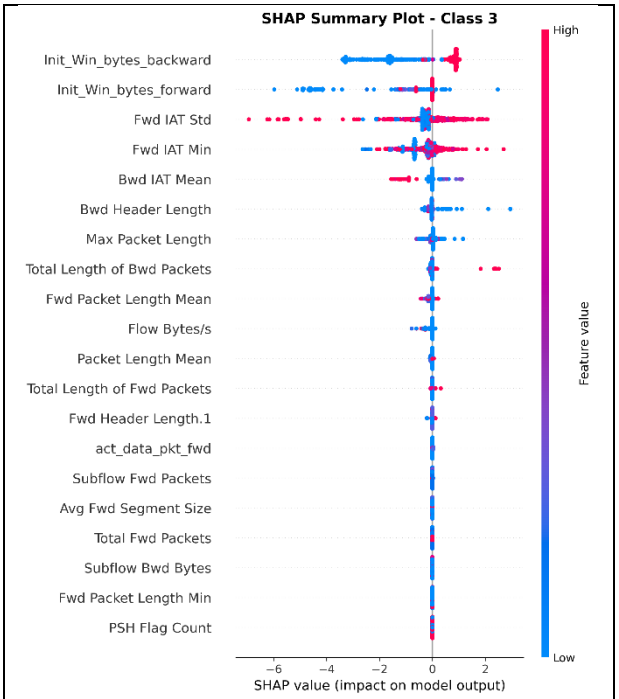
Εικόνα 107 SHAP Summary Plot, κλάση 0 για τον 3ο συνδυασμό.



Εικόνα 108 SHAP Summary Plot, κλάση 1 για τον 3ο συνδυασμό.



Εικόνα 109 SHAP Summary Plot, κλάση 2 για τον 3ο συνδυασμό.



Εικόνα 110 SHAP Summary Plot, κλάση 3 για τον 3ο συνδυασμό.

Στον τέταρτο συνδυασμό, φαίνεται ότι στο συνολικό Summary Plot (Εικόνα 111) το Idle Mean κυριαρχεί με μπλε σημεία αριστερά (≈ -1.0) και κόκκινα σημεία δεξιά (≈ +0.5), παρουσιάζοντας διπλό μοτίβο. Οι χαμηλές τιμές αδράνειας μειώνουν και οι υψηλές αυξάνουν την πιθανότητα αναγνώρισης της κακόβουλης κίνησης. Το Bwd Packet Length Max εμφανίζει κόκκινα σημεία δεξιά (≈ +1.0) που σημαίνει ότι οι υψηλές τιμές backward πακέτων αυξάνουν επίσης την πιθανότητα αναγνώρισης. Το Bwd IAT Std με μπλε σημεία δεξιά (έως +0.8) δείχνει ότι και η χαμηλή μεταβλητότητα backward IAT αυξάνει την πιθανότητα αναγνώρισης της κακόβουλης κίνησης.

Για την κλάση 0 που είναι η κανονική κίνηση (Εικόνα 112) το Bwd Packet Length Std κυριαρχεί με κόκκινα/μοβ σημεία αριστερά (≈ -6), υποδηλώνοντας ότι υψηλή μεταβλητότητα backward πακέτων μειώνει την πιθανότητα αναγνώρισης της κλάσης. Το Idle Mean με κόκκινα σημεία αριστερά (≈ -6) υποδηλώνει ότι υψηλή αδράνεια μειώνει την πιθανότητα αναγνώρισης της κίνησης ως κανονική. Τα Subflow Bwd Packets και Total Fwd Packets με κόκκινα σημεία δεξιά (έως +2) υποδηλώνουν ότι υψηλές τιμές τους αυξάνουν την πιθανότητα αναγνώρισης της κλάσης.

Για την κλάση 1 που αντιστοιχεί στην DoS GoldenEye (Εικόνα 113) το Idle Mean κυριαρχεί με εκτεταμένο εύρος στα μπλε σημεία αριστερά (≈ -6) και στα κόκκινα δεξιά (έως +6) που σημαίνει ότι τόσο οι πολύ χαμηλές όσο και οι πολύ υψηλές τιμές αδράνειας είναι διαγνωστικές για DoS GoldenEye. Το Bwd IAT Std με μπλε σημεία δεξιά (έως +6) υποδηλώνει ότι χαμηλή μεταβλητότητα backward IAT αυξάνει την πιθανότητα αναγνώρισης της κλάσης. Το Active Min με κόκκινα σημεία αριστερά υποδηλώνει ότι υψηλές τιμές ελάχιστου χρόνου ενεργοποίησης μειώνουν την πιθανότητα αναγνώρισης.

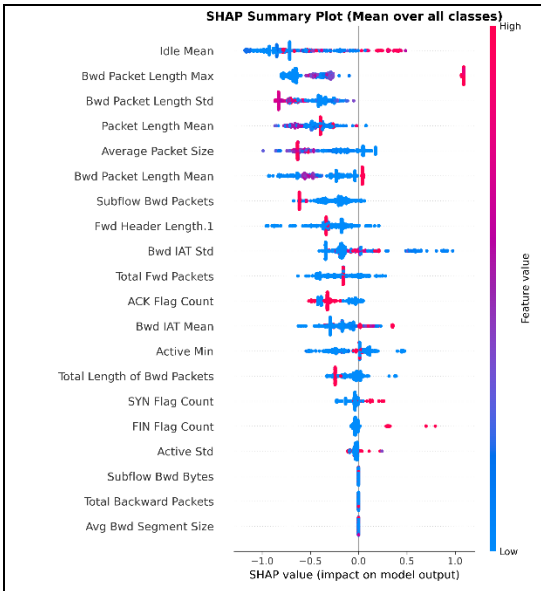
Για την κλάση 2 που είναι η DoS Hulk (Εικόνα 114) το Idle Mean παρουσιάζει μπλε σημεία αριστερά (≈ -4) και κόκκινα δεξιά (έως +6.5), υποδηλώνοντας ότι υψηλή αδράνεια αυξάνει σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Bwd Packet Length Max με μπλε σημεία δεξιά (έως +2.5) δείχνει ότι χαμηλές τιμές backward πακέτων μειώνουν την πιθανότητα αναγνώρισης. Το FIN Flag Count εμφανίζει κόκκινα σημεία δεξιά (έως +6.5) που σημαίνει ότι υψηλός αριθμός FIN σημαιών είναι ισχυρός δείκτης της συγκεκριμένης κακόβουλης επίθεσης.

Για την κλάση 3 (Εικόνα 115) η οποία αντιστοιχεί στην DoS Slowhttptest το Packet Length Mean εμφανίζει κόκκινα/μοβ σημεία αριστερά (≈ -2.5) και μπλε δεξιά, παρουσιάζοντας

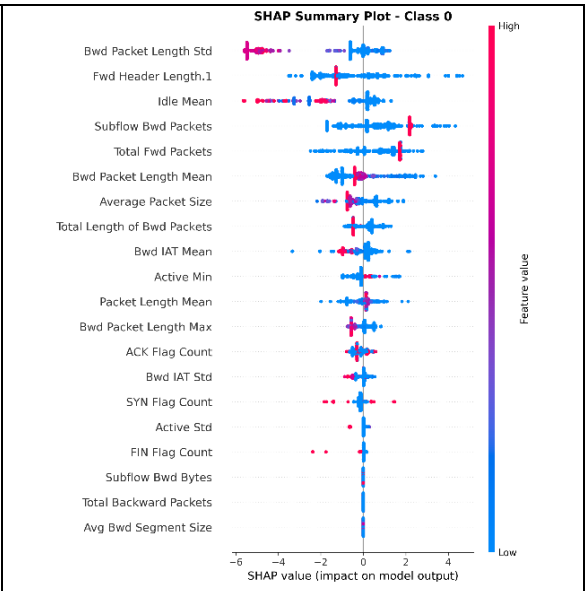
διπλό μοτίβο καθώς τόσο οι χαμηλές όσο και οι υψηλές τιμές μεγέθους πακέτων είναι διαγνωστικές. Το Active Min με κόκκινα σημεία δεξιά (έως +4) δείχνει ότι ο υψηλός ελάχιστος χρόνος ενεργοποίησης αυξάνει την πιθανότητα αναγνώρισης της κλάσης. Το Bwd IAT Std με κόκκινα σημεία δεξιά (έως +4) επιβεβαιώνει ότι και η υψηλή μεταβλητότητα backward IAT αυξάνει την πιθανότητα αναγνώρισής της.

Για την κλάση 4 που αντιστοιχεί στην DoS Slowloris (όπως δείχνει η Εικόνα 116) το Average Packet Size κυριαρχεί με μπλε/μοβ σημεία αριστερά (≈ -4.5) και μπλε σημεία δεξιά, παρουσιάζοντας ακόμα ενα διπολικό μοτίβο. Το Bwd Packet Length Mean με μοβ σημεία αριστερά (≈ -2) και μπλε δεξιά δείχνει ενα παρόμοιο μοτίβο. Το Bwd IAT Mean με κόκκινα σημεία δεξιά (έως +4.5) σημαίνει ότι ο υψηλός μέσος χρόνος backward IAT αυξάνει την πιθανότητα αναγνώρισης DoS Slowloris επιθέσεων.

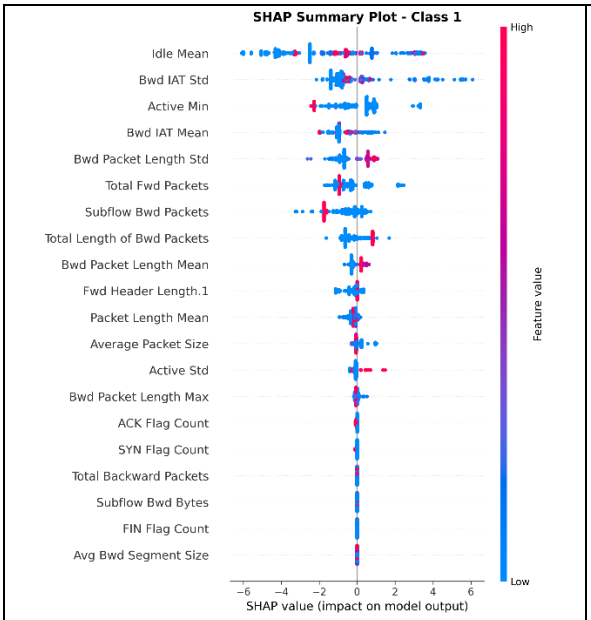
Για την κλάση 5 που είναι η Heartbleed (φαίνεται στην Εικόνα 117) το Bwd Packet Length Max παρουσιάζει ξανά το χαρακτηριστικό διπλό μοτίβο. Τα μπλε σημεία αριστερά (≈ -4.5) και τα κόκκινα δεξιά (≈ +5.5) επιβεβαιώνουν ότι τόσο οι πολύ χαμηλές όσο και οι πολύ υψηλές τιμές backward πακέτων αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Τα υπόλοιπα χαρακτηριστικά εμφανίζουν πολύ μικρό εύρος SHAP τιμών (< ±0.5), επιβεβαιώνοντας ότι το μοντέλο βασίζεται σχεδόν αποκλειστικά στο Bwd Packet Length Max για την αναγνώριση της Heartbleed επίθεσης.



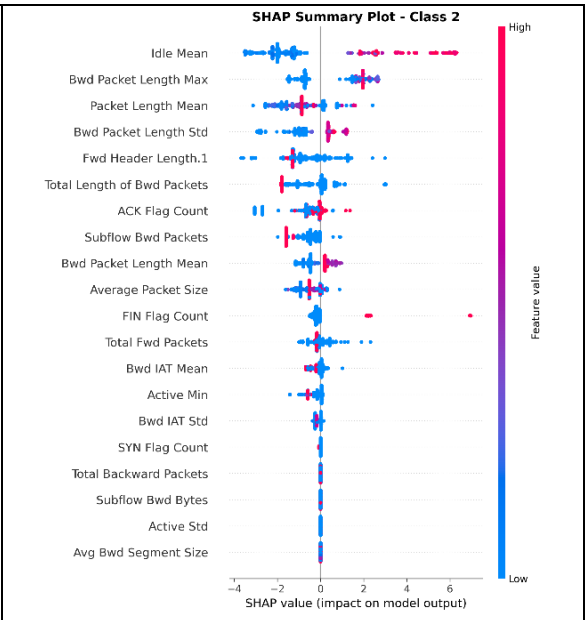
Εικόνα 111 SHAP Summary Plot, όλες οι κλάσεις για τον 4ο συνδυασμό.



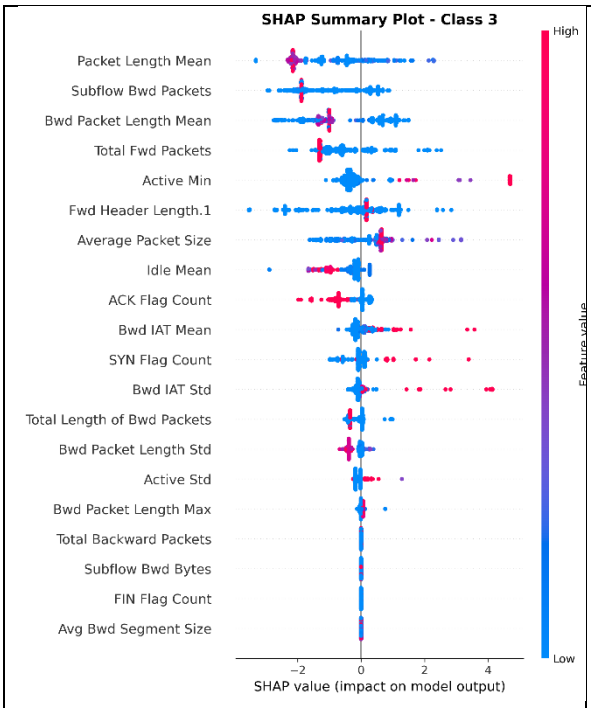
Εικόνα 112 SHAP Summary Plot, κλάση 0 για τον 4ο συνδυασμό.



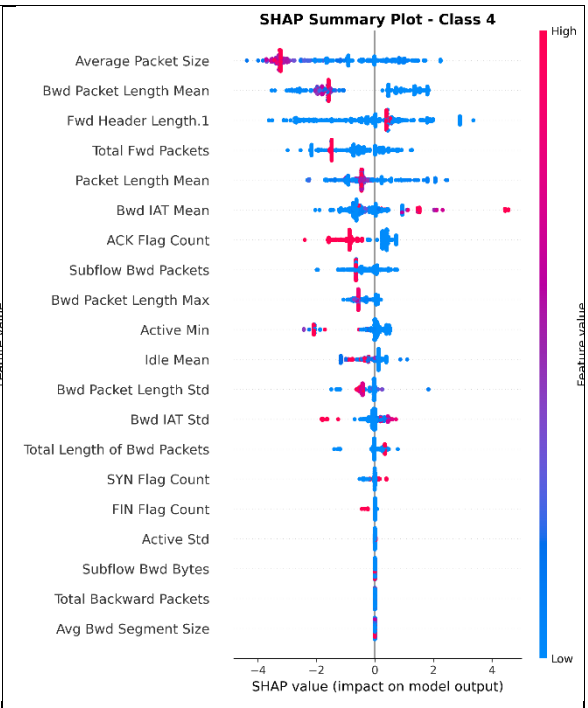
Εικόνα 113 SHAP Summary Plot, κλάση 1 για τον 4ο συνδυασμό.



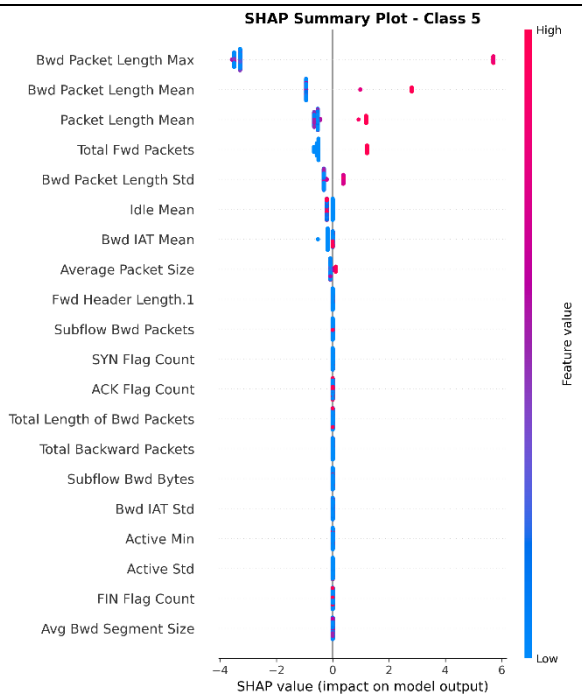
Εικόνα 114 SHAP Summary Plot, κλάση 2 για τον 4ο συνδυασμό.



Εικόνα 115 SHAP Summary Plot, κλάση 3 για τον 4ο συνδυασμό.



Εικόνα 116 SHAP Summary Plot, κλάση 4 για τον 4ο συνδυασμό.



Εικόνα 117 SHAP Summary Plot, κλάση 5 για τον 4ο συνδυασμό.

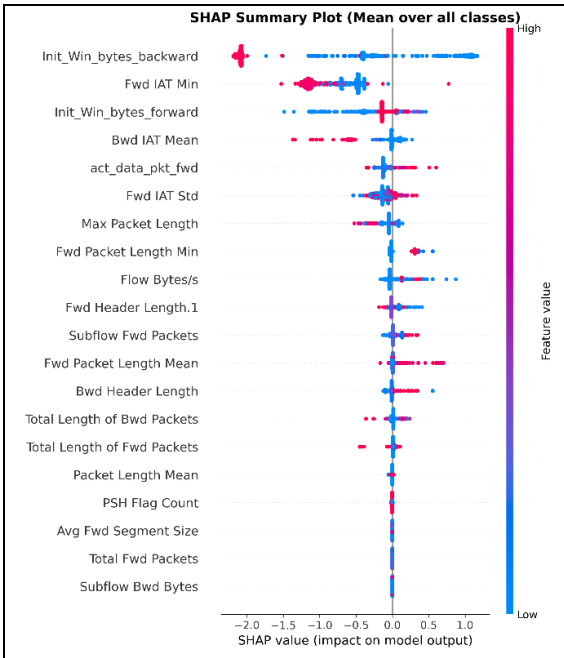
Στον πέμπτο συνδυασμό φαίνεται ότι στο συνολικό Summary Plot (βλέπε Εικόνα 118) το Init_Win_bytes_backward κυριαρχεί με κόκκινα σημεία αριστερά (≈ -2.5) και μπλε σημεία δεξιά (έως +1.2), παρουσιάζοντας ακόμα ένα διπλό μοτίβο. Οι υψηλές τιμές μειώνουν και χαμηλές αυξάνουν την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Fwd IAT Min εμφανίζει πυκνή συγκέντρωση κυρίως μπλε σημείων στα αριστερά, που σημαίνει ότι οι χαμηλές τιμές ελάχιστου χρονικού διαστήματος μειώνουν την πιθανότητα αναγνώρισης της επίθεσης. Τα Bwd IAT Mean και Init_Win_bytes_forward με κόκκινα σημεία αριστερά υποδηλώνουν ότι οι υψηλές τιμές τους μειώνουν την πιθανότητα αναγνώρισης.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 119) το Init_Win_bytes_backward παρουσιάζει κόκκινα σημεία αριστερά (≈ -5.5) και μπλε σημεία δεξιά (έως +5.5), υποδηλώνοντας ότι οι χαμηλές τιμές TCP backward παραθύρου αυξάνουν δραστικά την πιθανότητα αναγνώρισης της κίνησης ως BENIGN. Το Fwd IAT Min με κόκκινα σημεία αριστερά (≈ -3) δείχνει ότι οι χαμηλές τιμές ελάχιστου χρονικού διαστήματος αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Flow Bytes/s με σημεία δεξιά υποδηλώνει ότι και ο υψηλός ρυθμός bytes σχετίζεται με κανονική κίνηση.

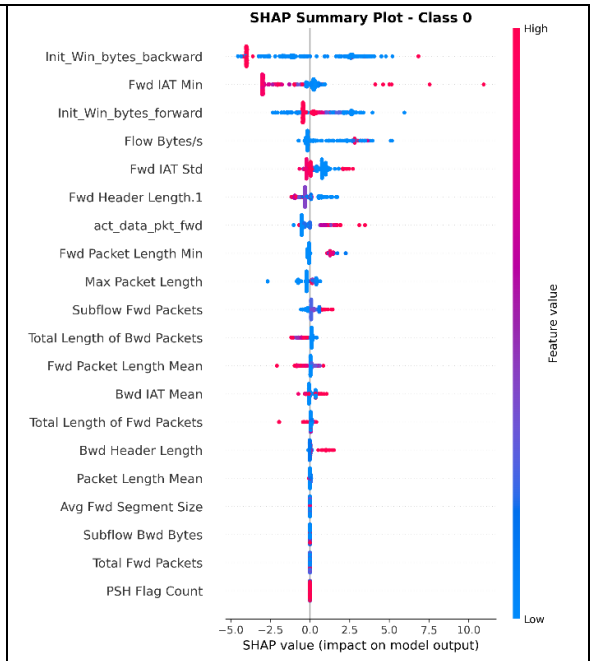
Για την κλάση 1 (Εικόνα 120) η οποία αντιστοιχεί στην Web Attack Brute Force, το Fwd IAT Min κυριαρχεί με κόκκινα σημεία δεξιά (≈ 3.0) δείχνοντας ότι οι υψηλές τιμές ελάχιστου χρονικού διαστήματος μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd IAT Std παρουσιάζει διπλό μοτίβο με κόκκινα σημεία αριστερά (≈ -2.5) και μπλε δεξιά (έως +2.5), γεγονός που υποδεικνύει ότι και η υψηλή μεταβλητότητα χρονικών διαστημάτων μειώνει την πιθανότητα αναγνώρισης. Το Fwd Packet Length Mean με κόκκινα σημεία δεξιά (έως +3.5) υποδηλώνει ότι το υψηλό μέσο μέγεθος προωθούμενων πακέτων αυξάνει την πιθανότητα αναγνώρισης αυτής της επίθεσης.

Για την κλάση 2 που αντιστοιχεί στην Web Attack SQL Injection (Εικόνα 121), το Init_Win_bytes_backward εμφανίζει κόκκινα σημεία αριστερά (≈ -5.5) και μπλε σημεία δεξιά (έως +10.0), παρουσιάζοντας ισχυρό διπλό μοτίβο, σύμφωνα με το οποίο οι χαμηλές τιμές TCP backward παραθύρου αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Fwd IAT Min από την άλλη, με κόκκινα σημεία αριστερά (≈ -2.5), υποδηλώνει ότι οι υψηλές τιμές ελάχιστου χρονικού διαστήματος μειώνουν την πιθανότητα αναγνώρισης. Τα υπόλοιπα χαρακτηριστικά εμφανίζουν πολύ μικρό εύρος SHAP τιμών (< ±1.0), που σημαίνει ότι η SQL Injection αναγνωρίζεται κυρίως μέσω του TCP backward παραθύρου.

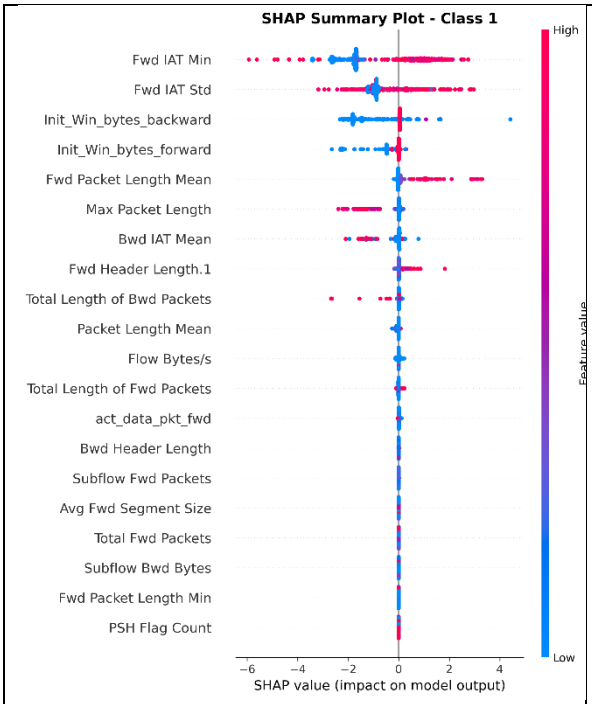
Για την κλάση 3 που είναι η Web Attack XSS (Εικόνα 122), το Init_Win_bytes_backward εμφανίζει κόκκινα σημεία δεξιά (≈ +1.3), από τα οποία μπορούμε να αντιληφθούμε ότι οι υψηλές τιμές TCP backward παραθύρου έχουν θετική επίδραση στην αναγνώριση της κλάσης. Το Init_Win_bytes_forward με κόκκινα σημεία αριστερά (έως -5) υποδηλώνει ότι υψηλές τιμές TCP forward παραθύρου μειώνουν την πιθανότητα αναγνώρισης. Το Fwd IAT Std με κόκκινα σημεία δεξιά και αριστερά υποδηλώνει ότι υψηλή μεταβλητότητα χρονικών διαστημάτων μπορεί να λειτουργήσει είτε υπέρ είτε κατά της πιθανότητας αναγνώρισης XSS.



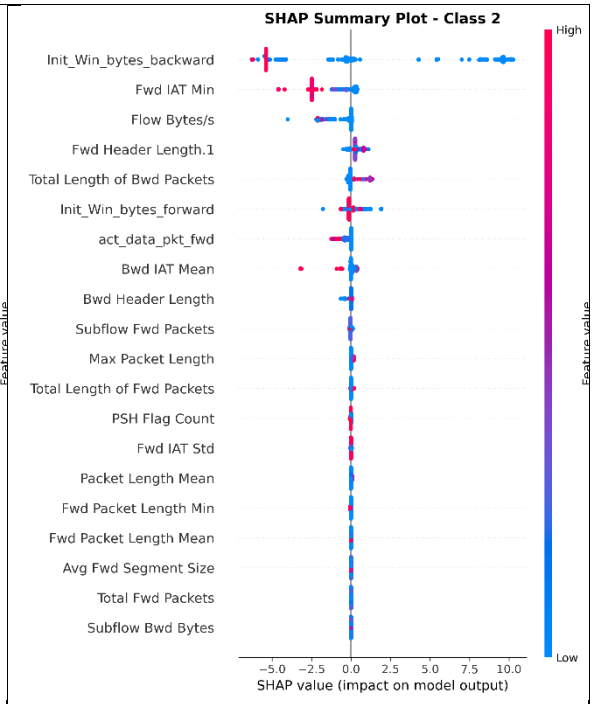
Εικόνα 118 SHAP Summary Plot, όλες οι κλάσεις για τον 5ο συνδυασμό.



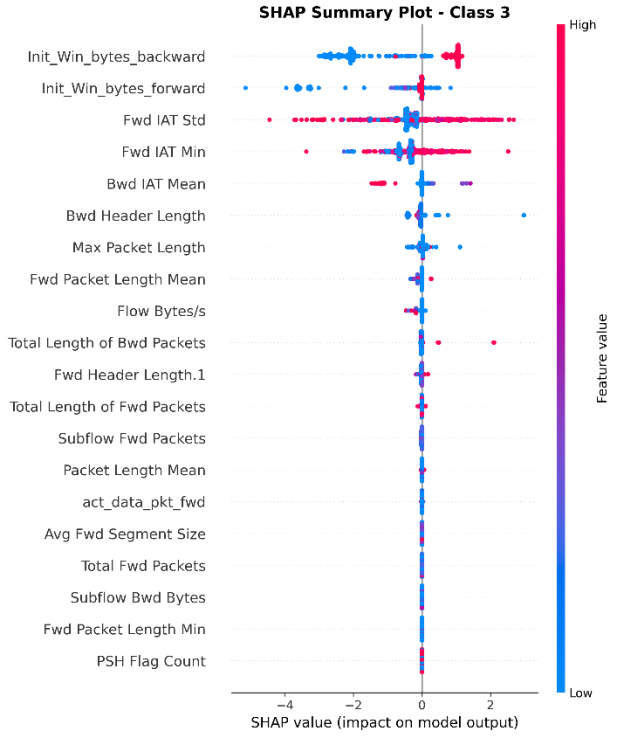
Εικόνα 119 SHAP Summary Plot, κλάση 0 για τον 5ο συνδυασμό.



Εικόνα 120 SHAP Summary Plot, κλάση 1 για τον 5ο συνδυασμό.

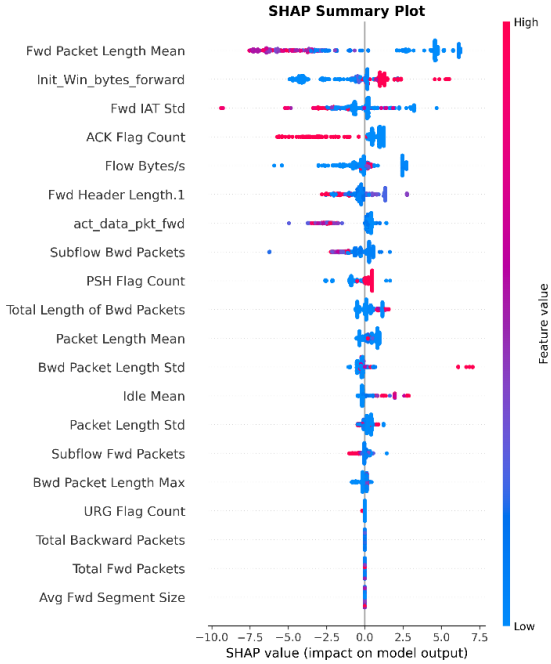


Εικόνα 121 SHAP Summary Plot, κλάση 2 για τον 5ο συνδυασμό.



Εικόνα 122 SHAP Summary Plot, κλάση 3 για τον 5ο συνδυασμό.

Στον έκτο συνδυασμό φαίνεται ότι στο συνολικό Summary Plot (Εικόνα 123) το Fwd Packet Length Mean κυριαρχεί με διπλό μοτίβο. Τα μοβ/κόκκινα σημεία αριστερά (έως -8) και μπλε σημεία δεξιά (έως +7), υποδηλώνοντας ότι οι πολύ χαμηλές μεγέθους πακέτων αυξάνουν την πιθανότητα αναγνώρισης κακόβουλης κίνησης, ενώ οι υψηλές τιμές τη μειώνουν. Το Init_Win_bytes_forward με μπλε σημεία αριστερά (≈ -5.0) και κόκκινα δεξιά (έως +6.0) παρουσιάζει επίσης διπλό μοτίβο. Οι χαμηλές τιμές μειώνουν και υψηλές αυξάνουν την πιθανότητα αναγνώρισης. Το ACK Flag Count με κόκκινα σημεία αριστερά (≈ -6.0) υποδηλώνει ότι υψηλός αριθμός ACK flags μειώνει την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Bwd Packet Length Std με κόκκινα σημεία δεξιά (έως +7.5) υποδηλώνει ότι υψηλή μεταβλητότητα backward πακέτων αυξάνει σημαντικά την πιθανότητα αναγνώρισης επίθεσης. Τα τελευταία χαρακτηριστικά (URG Flag Count, Total Backward Packets, Total Fwd Packets, Avg Fwd Segment Size) εμφανίζουν σχεδόν μηδενική επίδραση.



Εικόνα 123 SHAP Summary Plot, όλες οι κλάσεις για τον 6ο συνδυασμό.

Στον έβδομο συνδυασμό, στο συνολικό SHAP Summary Plot (Εικόνα 124) παρατηρούμε ότι το Bwd Packet Length Max κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας έντονο διπλό μοτίβο όπου μπλε σημεία αριστερά (≈ −0.8) και κόκκινα σημεία δεξιά (έως +1.6), δείχνουν ότι χαμηλές τιμές μέγιστου μεγέθους πακέτου προς τα πίσω μειώνουν ενώ υψηλές αυξάνουν συνολικά την πιθανότητα ανίχνευσης. Το Idle Mean εμφανίζει πυκνή συγκέντρωση σημείων τόσο αριστερά (μπλε, ≈ −1.0) όσο και δεξιά (κόκκινα, ≈ +0.5), που σημαίνει ότι οι τιμές του επηρεάζουν με διαφορετικό τρόπο ανά κλάση. Το Bwd Packet Length Std με κόκκινα σημεία αριστερά (≈ −1.2) και κάποια λίγα μπλε δεξιά δείχνει ότι η υψηλή τυπική απόκλιση των πακέτων προς τα πίσω μειώνει την πιθανότητα ανίχνευσης. Τα features Fwd Header Length.1, Average Packet Size και Packet Length Mean εμφανίζουν μέτρια αλλά ομοιόμορφη συμβολή, ενώ χαρακτηριστικά όπως το Avg Bwd Segment Size και το Total Backward Packets παρουσιάζουν πολύ μικρό εύρος SHAP τιμών, συμβάλλοντας ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 125), το Bwd Packet Length Std κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας πυκνή συγκέντρωση μπλε και μοβ σημείων αριστερά (≈ −3 έως −7), υποδηλώνοντας ότι οι

χαμηλές τιμές τυπικής απόκλισης του μεγέθους των πακέτων προς τα πίσω μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Idle Mean εμφανίζει διπλό μοτίβο με κόκκινα σημεία αριστερά και μπλε σημεία γύρω από το μηδέν, δείχνοντας ότι οι υψηλές τιμές χρόνου αδράνειας μειώνουν την πιθανότητα αναγνώρισης. Το Fwd Header Length.1 και τα Total Fwd Packets παρουσιάζουν κυρίως θετικές SHAP τιμές (σημεία δεξιά), υποδηλώνοντας ότι ο υψηλός αριθμός προωθούμενων πακέτων και το μεγάλο μέγεθος επικεφαλίδας αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Τα χαρακτηριστικά ACK Flag Count, SYN Flag Count και FIN Flag Count εμφανίζουν μικρό εύρος τιμών με σημεία κοντά στο μηδέν, συμβάλλοντας ελάχιστα στη διάκριση της κανονικής κίνησης.

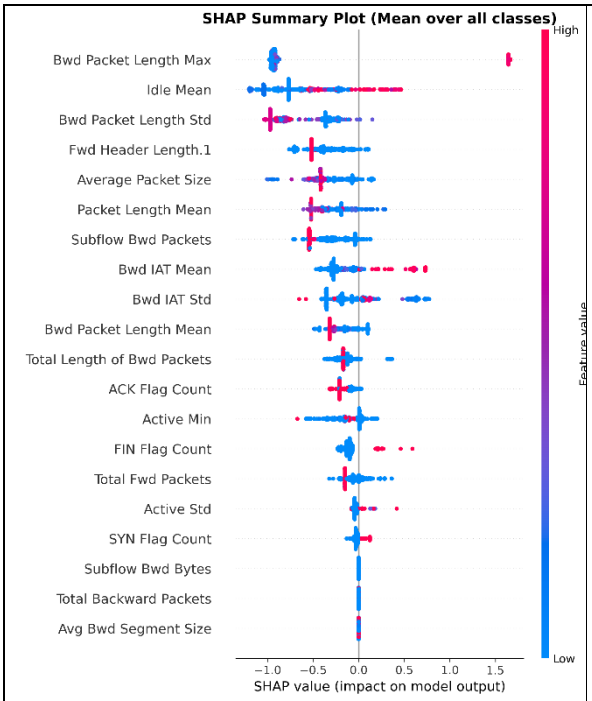
Για την κλάση 1 που αντιστοιχεί στην επίθεση DoS GoldenEye (φαίνεται στην Εικόνα 126), το Idle Mean επικρατεί με έντονο διπλό μοτίβο, καθώς κόκκινα σημεία αριστερά (≈ −5) και μπλε σημεία δεξιά (έως +3), γεγονός που σημαίνει ότι οι υψηλές τιμές μέσου χρόνου αδράνειας μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Bwd IAT Std εμφανίζει μπλε σημεία δεξιά (έως +5), υποδηλώνοντας ότι η χαμηλή τυπική απόκλιση του χρονικού διαστήματος μεταξύ των πακέτων προς τα πίσω αυξάνει σημαντικά την πιθανότητα αναγνώρισης αυτής της επίθεσης. Το Fwd Header Length.1 παρουσιάζει κόκκινα σημεία κοντά στο μηδέν και μπλε δεξιά (έως +3.5), υποδηλώνοντας ότι οι χαμηλές τιμές μήκους της επικεφαλίδας σχετίζονται θετικά με της κλάση. Τα χαρακτηριστικά SYN Flag Count, ACK Flag Count, FIN Flag Count και Avg Bwd Segment Size συγκεντρώνονται κοντά στο μηδέν, δείχνοντας ότι παρέχουν ελάχιστη συμβολή στην αναγνώριση αυτής της κλάσης.

Για την κλάση 2 που αντιστοιχεί στην επίθεση DoS Hulk (όπως δείχνει η Εικόνα 127), το Idle Mean κυριαρχεί με ισχυρό θετικό μοτίβο, όπου κόκκινα σημεία δεξιά (έως +6.0) και μπλε σημεία αριστερά (≈ −3.0), υποδεικνύουν ότι οι υψηλές τιμές του μέσου χρόνου αδράνειας αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της συγκεκριμένης κλάσης. Το Bwd Packet Length Std παρουσιάζει μπλε σημεία αριστερά (≈ −4) και κόκκινα / μοβ σημεία κοντά στο μηδέν, υποδηλώνοντας ότι η χαμηλή τυπική απόκλιση του μεγέθους των πακέτων προς τα πίσω μειώνει την πιθανότητα ανίχνευσης. Το FIN Flag Count εμφανίζει διάσπαρτα κόκκινα σημεία δεξιά (έως +6), που σημαίνει ότι ο υψηλός αριθμός των FIN σημαιών αποτελεί χαρακτηριστικό δείκτη αυτής της κλάσης. Τα χαρακτηριστικά Total Backward Packets, Subflow Bwd Bytes και Avg Bwd Segment Size εμφανίζουν SHAP τιμές πολύ κοντά στο μηδέν, συμβάλλοντας ουσιαστικά ελάχιστα στην αναγνώριση.

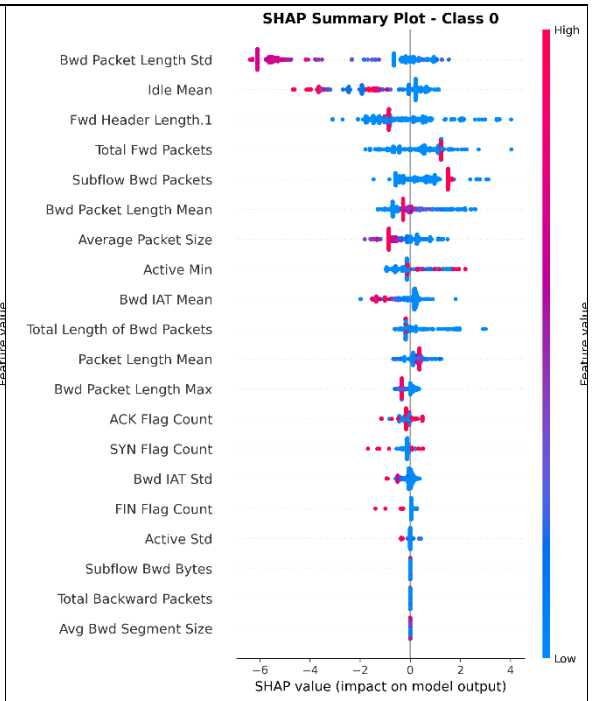
Για την κλάση 3 που αντιστοιχεί στην επίθεση DoS Slowhttptest (Εικόνα 128), το Bwd Packet Length Mean κυριαρχεί παρουσιάζοντας ακόμα ένα διπλό μοτίβο, όπου κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (έως +1.9), δείχνουν ότι οι υψηλές τιμές του μέσου μεγέθους πακέτων προς τα πίσω μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Subflow Bwd Packets εμφανίζει μπλε σημεία αριστερά (έως −4), υποδηλώνοντας ότι ο χαμηλός αριθμός των πακέτων subflow προς τα πίσω μειώνει την πιθανότητα ανίχνευσης. Το Average Packet Size παρουσιάζει κόκκινα σημεία δεξιά (≈ +1.9), που σημαίνει ότι το υψηλό μέσο μέγεθος του πακέτου αυξάνει την πιθανότητα αναγνώρισης αυτής της επίθεσης, όπως ισχύει και για το Active Min που εμφανίζει κόκκινα σημεία δεξιά του κεντρικού άξονα (έως +5 περίπου).

Για την κλάση 4 που αντιστοιχεί στην επίθεση DoS Slowloris (Εικόνα 129), το Average Packet Size παρουσιάζεται με ακόμα ένα διπλό μοτίβο σύμφωνα με το οποίο, τα μπλε σημεία αριστερά (≈ −5) και τα κόκκινα σημεία δεξιά (έως +3.0), δείχνουν ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτου μειώνουν ενώ οι υψηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Bwd IAT Mean εμφανίζει κόκκινα σημεία δεξιά (έως +6.5), από τα οποία προκύπτει ότι οι υψηλές τιμές του μέσου χρονικού διαστήματος μεταξύ των πακέτων προς τα πίσω αυξάνουν σημαντικά την πιθανότητα ανίχνευσής της. Το Fwd Header Length.1 παρουσιάζει κόκκινα σημεία κοντά στο μηδέν και μπλε αριστερά (≈ -2.0), υποδηλώνοντας ότι οι χαμηλές τιμές του αυτού χαρακτηριστικού μειώνουν την πιθανότητα ανίχνευσης.

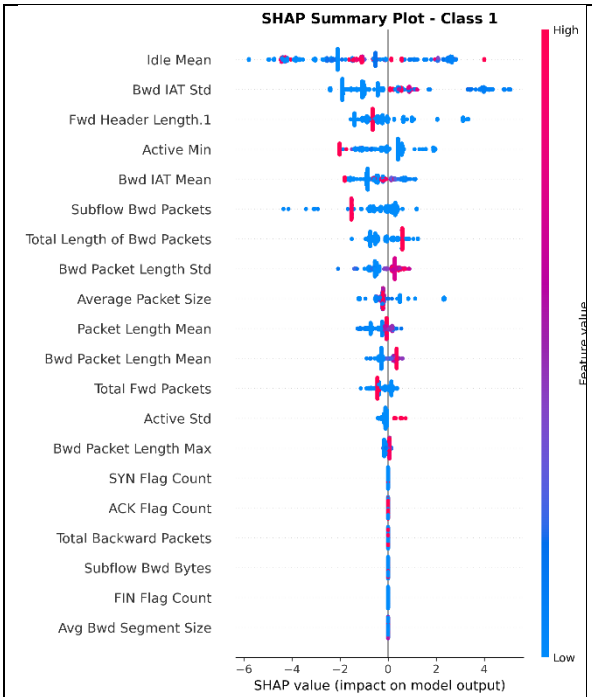
Για την κλάση 5 που αντιστοιχεί στην επίθεση Heartbleed (Εικόνα 130), το Bwd Packet Length Max παρουσιάζεται με ένα εξαιρετικά ισχυρό μοτίβο, στο οποίο φαίνονται μπλε σημεία αριστερά (≈ −5) και ένα μεμονωμένο κόκκινο σημείο δεξιά (≈ +10), που σημαίνει ότι ένα μόνο πακέτο με εξαιρετικά υψηλό μέγιστο μέγεθος προς τα πίσω αποτελεί καθοριστικό δείκτη για την αναγνώριση της κλάσης, ενώ οι χαμηλές τιμές μειώνουν σημαντικά την πιθανότητα ανίχνευσης. Το Average Packet Size παρουσιάζει μικρές θετικές SHAP τιμές (σημεία δεξιά, ≈ +0.5), ενώ όλα τα υπόλοιπα χαρακτηριστικά (Packet Length Mean, Total Fwd Packets, SYN/ACK/FIN Flag Count, Idle Mean, κ.λπ.) εμφανίζουν πρακτικά μηδενικές τιμές. Αυτό σημαίνει ότι η ανίχνευση της κλάσης βασίζεται σχεδόν αποκλειστικά στο χαρακτηριστικό μέγεθος του backward πακέτου, αντικατοπτρίζοντας τον μηχανισμό της επίθεσης που εκμεταλλεύεται μεγάλου μεγέθους TLS απαντήσεις (για διαρροή μνήμης).



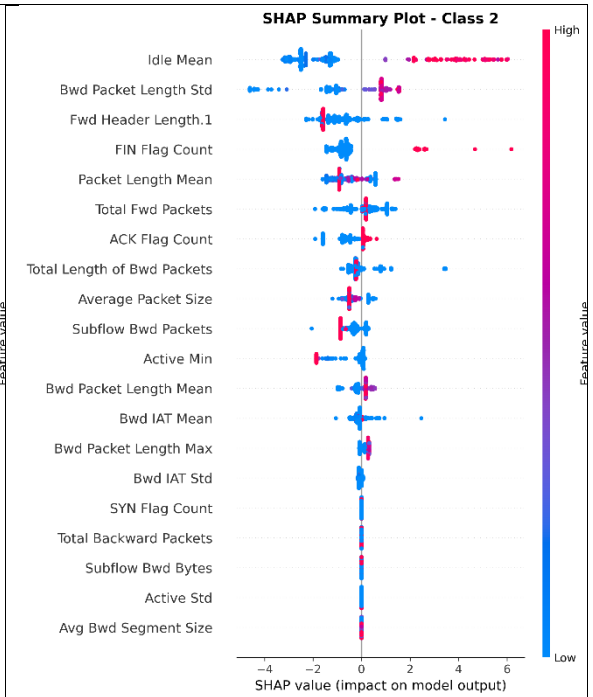
Εικόνα 124 SHAP Summary Plot, όλες οι κλάσεις για τον 7ο συνδυασμό.



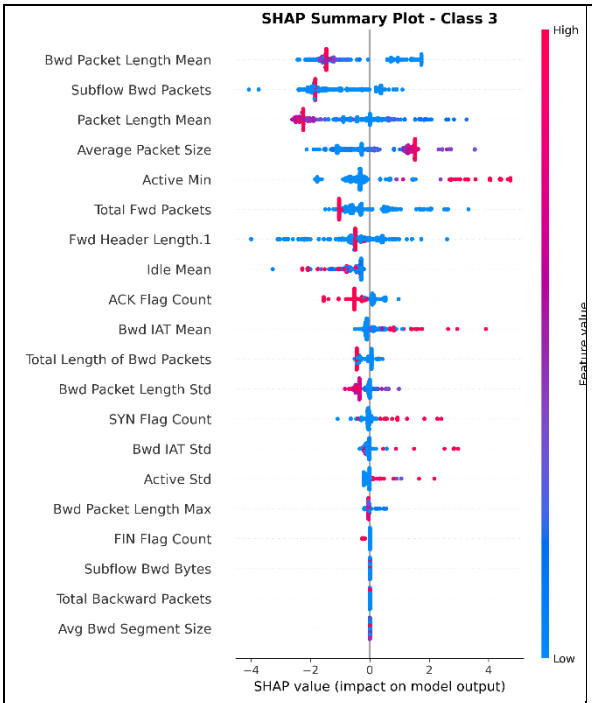
Εικόνα 125 SHAP Summary Plot, κλάση 0 για τον 7ο συνδυασμό.



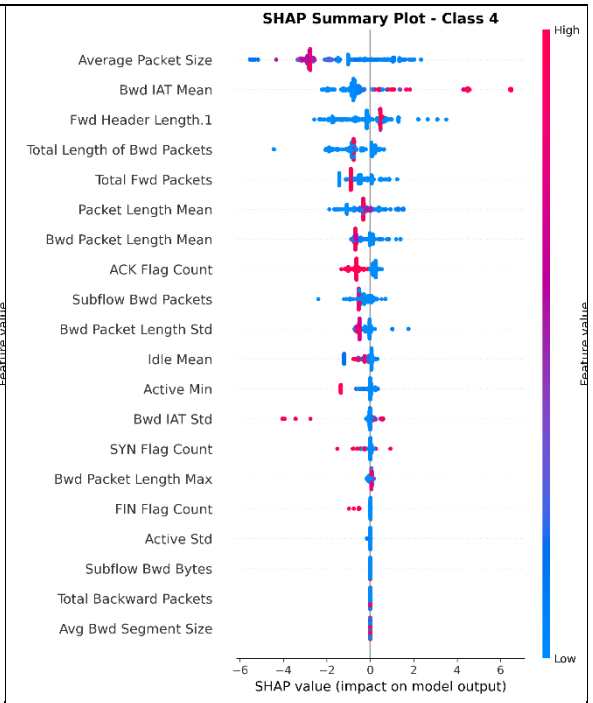
Εικόνα 126 SHAP Summary Plot, κλάση 1 για τον 7ο συνδυασμό.



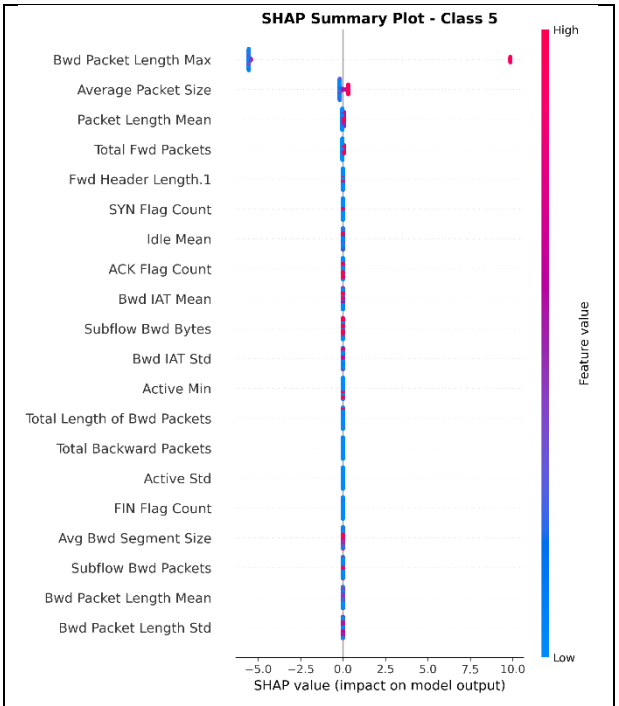
Εικόνα 127 SHAP Summary Plot, κλάση 2 για τον 7ο συνδυασμό.



Εικόνα 128 SHAP Summary Plot, κλάση 3 για τον 7ο συνδυασμό.



Εικόνα 129 SHAP Summary Plot, κλάση 4 για τον 7ο συνδυασμό.



Εικόνα 130 SHAP Summary Plot, κλάση 5 για τον 7ο συνδυασμό.

Στο συνολικό SHAP Summary Plot του όγδοου συνδυασμού (βλέπε Εικόνα 131) παρατηρούμε ότι το Init_Win_bytes_backward κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας έντονο διπλό μοτίβο, όπου κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (έως +0.7), υποδηλώνουν ότι οι υψηλές τιμές του TCP backward παραθύρου μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα ανίχνευσης κακόβουλης κίνησης. Το Fwd IAT Min εμφανίζει πυκνή συγκέντρωση μπλε σημείων κυρίως αριστερά του κεντρικού άξονα, υποδηλώνοντας ότι οι χαμηλές τιμές του ελάχιστου χρονικού διαστήματος μεταξύ των προωθούμενων πακέτων μειώνουν συνολικά την πιθανότητα αναγνώρισης επίθεσης. Το Init_Win_bytes_forward με κυρίως μπλε σημεία αριστερά δείχνει ότι οι χαμηλές τιμές TCP forward παραθύρου σχετίζονται αρνητικά με την ανίχνευση. Το Fwd IAT Std και το act_data_pkt_fwd παρουσιάζουν μέτρια συμβολή με σημεία κοντά στο μηδέν, ενώ χαρακτηριστικά όπως τα Subflow Bwd Bytes, Total Fwd Packets και PSH Flag Count παρουσιάζουν πρακτικά μηδενικές SHAP τιμές, συμβάλλοντας ελάχιστα στη συνολική απόφαση του μοντέλου.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (φαίνεται στην Εικόνα 132), επικρατεί το Init_Win_bytes_backward παρουσιάζοντας έντονο διπλό μοτίβο, όπου κόκκινα σημεία αριστερά (≈ −4.0) και μπλε σημεία δεξιά (έως +4.0), δείχνοντας ότι οι χαμηλές τιμές TCP backward παραθύρου αυξάνουν δραστικά την πιθανότητα αναγνώρισης ως BENIGN, ενώ υψηλές τιμές τη μειώνουν. Το Fwd IAT Min παρουσιάζει επίσης διπλό μοτίβο με κόκκινα σημεία αριστερά (≈ −2.0) και δεξιά (έως +9.0), που σημαίνει ότι τόσο οι χαμηλές όσο και οι πολύ υψηλές τιμές του ελάχιστου χρονικού διαστήματος αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Init_Win_bytes_forward εμφανίζει κόκκινα σημεία αριστερά (≈ −1.0) και μπλε σημεία δεξιά (≈ +4.0), παρουσιάζοντας ανάλογη συμπεριφορά με το backward παράθυρο. Το Flow Bytes/s με μπλε σημεία γύρο από το μηδέν και δεξιά του υποδηλώνει ότι υψηλός ρυθμός bytes σχετίζεται με κανονική κίνηση. Τα χαρακτηριστικά PSH Flag Count, Total Fwd Packets και Subflow Bwd Bytes πρακτικά εμφανίζουν μηδενικές SHAP τιμές, συμβάλλοντας ελάχιστα στη διάκριση της κανονικής κίνησης.

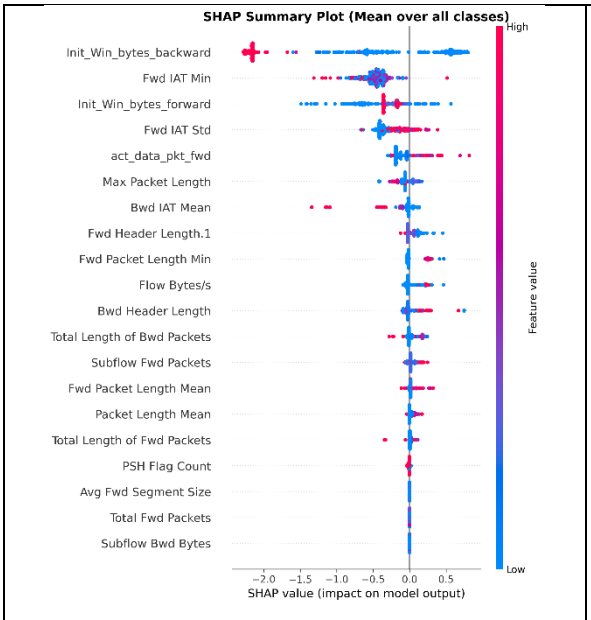
Για την κλάση 1 που αντιστοιχεί στην επίθεση Web Attack Brute Force (Εικόνα 133), το Fwd IAT Min εμφανίζει κόκκινα σημεία αριστερά (≈ −5.0) και μπλε / μοβ σημεία δεξιά (έως +2.0), υποδηλώνοντας ότι οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd IAT

Std παρουσιάζει διπλό μοτίβο με κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (έως +3.5), που σημαίνει ότι η υψηλή μεταβλητότητα των χρονικών διαστημάτων μειώνει ενώ η χαμηλή αυξάνει την πιθανότητα αναγνώρισης, αντικατοπτρίζοντας τον τακτικό, επαναληπτικό ρυθμό αποστολής των αιτημάτων που χαρακτηρίζει αυτή την επίθεση. Το Fwd Packet Length Mean με κόκκινα σημεία δεξιά (≈ +2.0) δείχνει ότι το υψηλό μέσο μέγεθος προωθούμενων πακέτων αυξάνει την πιθανότητα αναγνώρισης. Τα features PSH Flag Count, Subflow Bwd Bytes, Total Fwd Packets και Fwd Packet Length Min εμφανίζουν τιμές SHAP συγκεντρωμένες στο μηδέν, συμβάλλοντας ουσιαστικά ελάχιστα στην αναγνώριση της κλάσης.

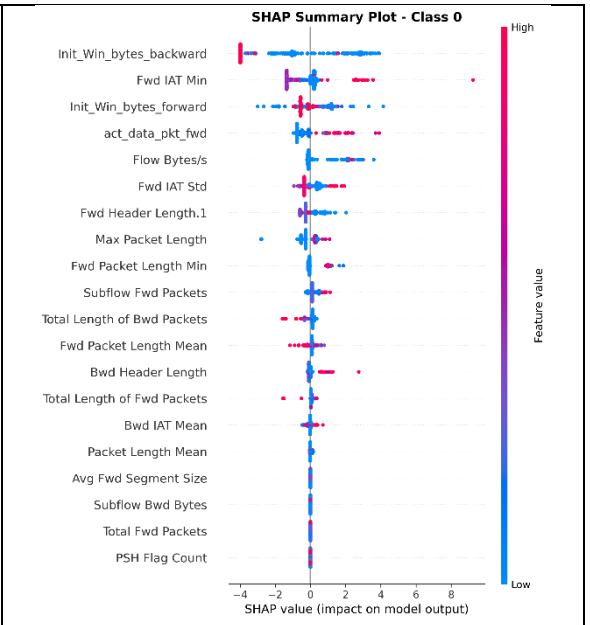
Για την κλάση 2 που αντιστοιχεί στην επίθεση Web Attack SQL Injection (Εικόνα 134), το Init_Win_bytes_backward επικρατεί με κόκκινα σημεία αριστερά (≈ −6.0) και μπλε σημεία δεξιά (έως +8.0), υποδηλώνοντας ότι οι χαμηλές τιμές του TCP backward παραθύρου αυξάνουν σημαντικά την πιθανότητα αναγνώρισης SQL Injection, ενώ οι υψηλές τιμές τη μειώνουν. Το Fwd IAT Min παρουσιάζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (≈ +0.5), δείχνοντας ότι οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος μειώνουν την πιθανότητα ανίχνευσης. Το Init_Win_bytes_forward εμφανίζει κυρίως μπλε σημεία δεξιά (≈ +2.0), υποδηλώνοντας ότι οι χαμηλές τιμές TCP forward παραθύρου έχουν ήπια θετική επίδραση. Τα υπόλοιπα χαρακτηριστικά (PSH Flag Count, Fwd Packet Length Mean, Max Packet Length, Subflow Bwd Bytes κ.λπ.) εμφανίζουν πολύ μικρό εύρος SHAP τιμών (< ±0.5), γεγονός που σημαίνει ότι η συγκεκριμένη κλάση αναγνωρίζεται κυρίως μέσω των χαρακτηριστικών του TCP παραθύρου.

Για την κλάση 3 που αντιστοιχεί στην επίθεση Web Attack XSS (Εικόνα 135), το Init_Win_bytes_backward εμφανίζει διπλό μοτίβο με μπλε σημεία αριστερά (≈ -3.5) και κόκκινα σημεία δεξιά (≈ +1.6), που σημαίνει ότι οι χαμηλές τιμές TCP backward παραθύρου έχουν μειώνουν ενώ οι υψηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Init_Win_bytes_forward παρουσιάζει μπλε σημεία αριστερά (≈ −4) και κόκκινα σημεία κοντά στο μηδέν, υποδηλώνοντας ότι και οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν την πιθανότητα αναγνώρισης. Το Fwd IAT Std επικρατεί με κόκκινα σημεία δεξιά (έως +2.5), κάνοντας σαφές ότι η υψηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει σημαντικά την πιθανότητα αναγνώρισης της XSS. Το Fwd IAT Min παρουσιάζει επίσης κόκκινα σημεία δεξιά (≈ +2.0), υποδηλώνοντας ότι και οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος αυξάνουν την πιθανότητα αναγνώρισης. Τα

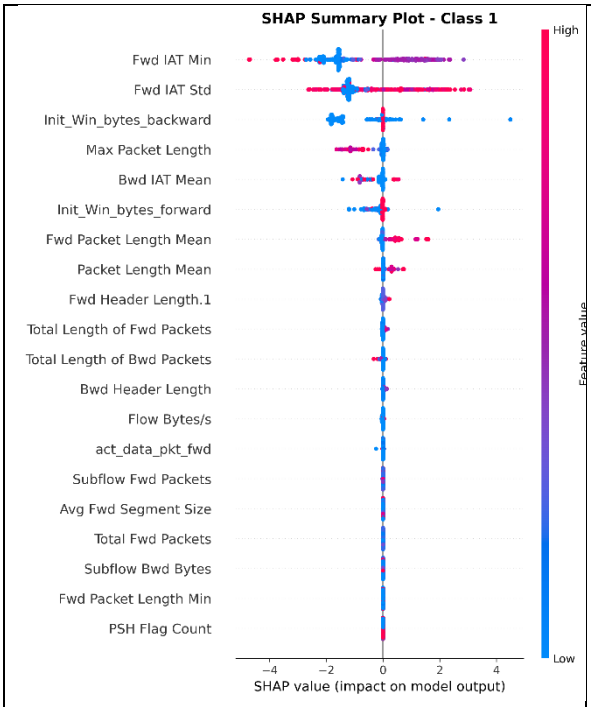
χαρακτηριστικά PSH Flag Count, Subflow Bwd Bytes, Fwd Packet Length Min και Total Fwd Packets εμφανίζουν μηδενικές SHAP τιμές, συμβάλλοντας ελάχιστα στην αναγνώριση αυτής της επίθεσης.



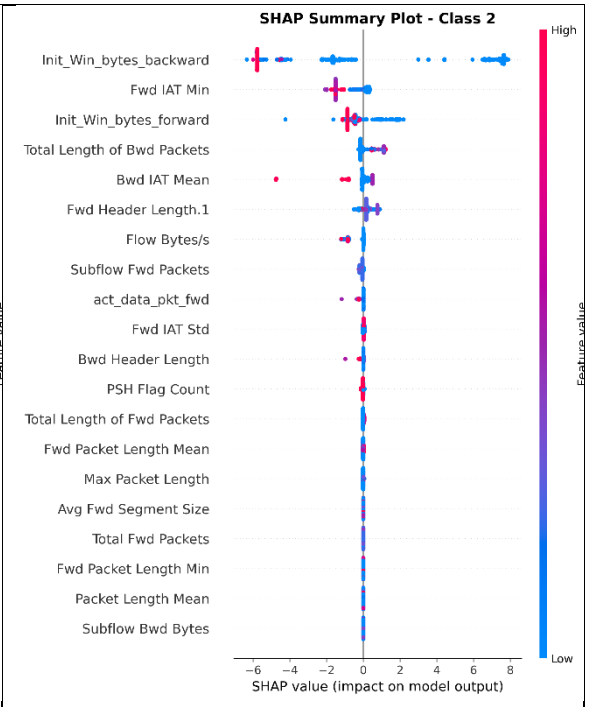
Εικόνα 131 SHAP Summary Plot, όλες οι κλάσεις για τον 8ο συνδυασμό.



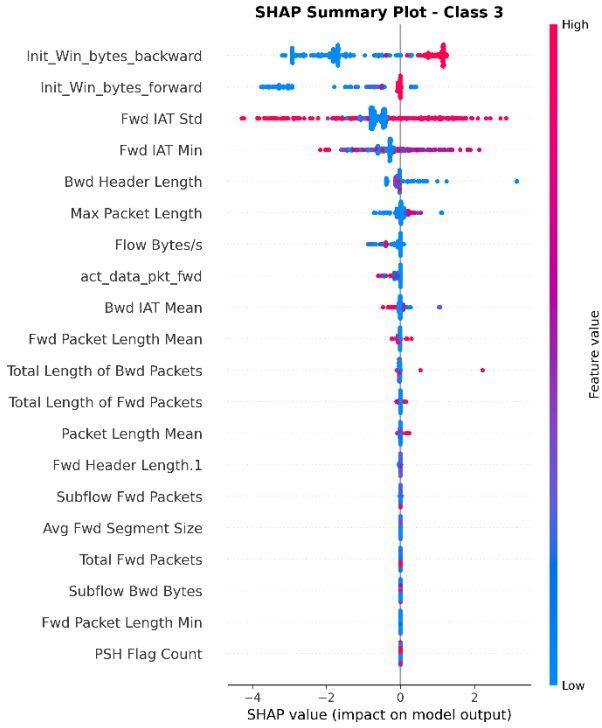
Εικόνα 132 SHAP Summary Plot, κλάση 0 για τον 8ο συνδυασμό.



Εικόνα 133 SHAP Summary Plot, κλάση 1 για τον 8ο συνδυασμό.



Εικόνα 134 SHAP Summary Plot, κλάση 2 για τον 8ο συνδυασμό.

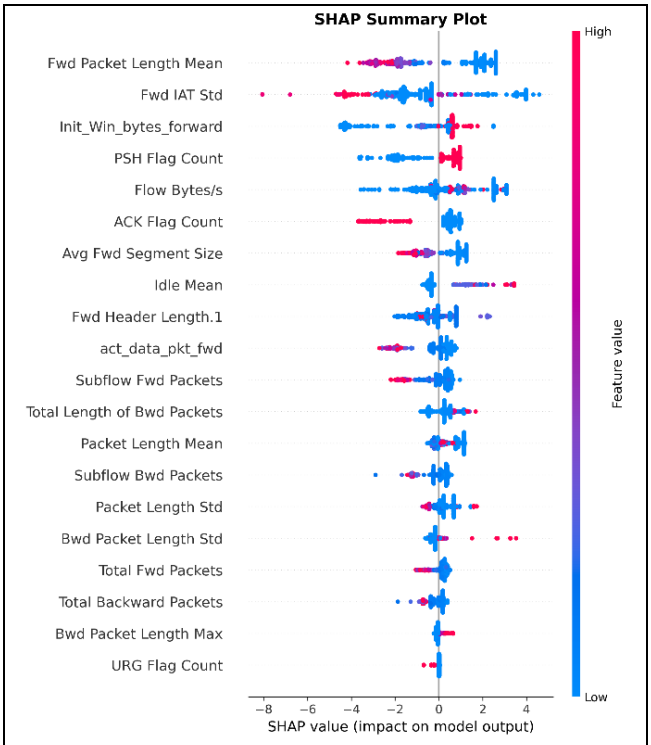


Εικόνα 135 SHAP Summary Plot, κλάση 3 για τον 8ο συνδυασμό.

Στο συνολικό SHAP Summary Plot του ένατου συνδυασμού (αποτυπώνεται στην Εικόνα 136) παρατηρούμε ότι το Fwd Packet Length Mean επικρατεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας ένα σύνθετο μοτίβο, σύμφωνα με το οποίο μπλε σημεία δεξιά με πυκνή συγκέντρωση που εκτείνεται έως +2.5 με μια μικρή διακοπή στο ≈ +1.5, αλλά και μπλε σημεία αριστερά (≈ −1), υποδηλώνουν ότι οι χαμηλές τιμές των μέσου μεγέθους προωθούμενων πακέτων μπορούν να έχουν είτε θετική είτε αρνητική επίδραση στην αναγνώριση κακόβουλης κίνησης ανάλογα με το σύνολο. Τα κόκκινα σημεία αριστερά (≈ −2.0 έως −5.0) δείχνουν ότι οι υψηλές τιμές μειώνουν την πιθανότητα αναγνώρισης επίθεσης. Το Fwd IAT Std εμφανίζει επίσης διπλό μοτίβο με κόκκινα σημεία αριστερά (≈ έως −8.0) και μπλε σημεία δεξιά (έως +4), υποδηλώνοντας ότι η υψηλή τυπική απόκλιση των χρονικών διαστημάτων μεταξύ των προωθούμενων πακέτων μειώνει σημαντικά την πιθανότητα αναγνώρισης. Αυτό σημαίνει ουσιαστικά ότι οι κακόβουλες ροές χαρακτηρίζονται από πιο ομοιόμορφο και τακτικό ρυθμό αποστολής πακέτων.

Το Init_Win_bytes_forward παρουσιάζει κυρίως μπλε σημεία αριστερά (≈ −4.6), υποδηλώνοντας ότι οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν την πιθανότητα ανίχνευσης. Το PSH Flag Count εμφανίζει μπλε σημεία αριστερά (≈ −2.0) και κόκκινα

σημεία δεξιά (≈ +1.0), κάνοντας ξεκάθαρο ότι ο υψηλός αριθμός των PSH σημαιών αυξάνει την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Flow Bytes/s παρουσιάζει μπλε σημεία αριστερά και διάσπαρτα κόκκινα σημεία δεξιά, δείχνοντας ότι ο εξαιρετικά υψηλός ρυθμός bytes ανά δευτερόλεπτο αυξάνει σημαντικά την πιθανότητα αναγνώρισης επίθεσης. Το ACK Flag Count εμφανίζει κόκκινα σημεία αριστερά (≈ −3.0) και μπλε σημεία δεξιά (≈ +1.0), υποδηλώνοντας ότι ο υψηλός αριθμός ACK σημαιών μειώνει την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Avg Fwd Segment Size παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (≈ +1.0), ενώ το Idle Mean με κόκκινα σημεία δεξιά (έως +3.5) υποδηλώνει ότι ο υψηλός μέσος χρόνος αδράνειας αυξάνει την πιθανότητα αναγνώρισης επίθεσης. Το Fwd Header Length.1 και το act_data_pkt_fwd εμφανίζουν μέτρια συμβολή με την πλειοψηφία των σημείων κοντά στο μηδέν, ενώ τα χαρακτηριστικά URG Flag Count, Bwd Packet Length Max, Total Backward Packets και Total Fwd Packets παρουσιάζουν πολύ μικρό εύρος SHAP τιμών, συμβάλλοντας ελάχιστα στη διάκριση της κίνησης.



Εικόνα 136 SHAP Summary Plot, όλες οι κλάσεις για τον 9ο συνδυασμό.

Στο δέκατο συνδυασμό, στο συνολικό SHAP Summary Plot (Εικόνα 137) παρατηρούμε ότι το Fwd Header Length.1 κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας κυρίως μπλε σημεία αριστερά του κεντρικού άξονα, υποδηλώνοντας ότι οι χαμηλές τιμές του μήκους των επικεφαλίδων των προωθούμενων πακέτων μειώνουν συνολικά την πιθανότητα ανίχνευσης. Το Bwd Packet Length Std εμφανίζει κόκκινα και μπλε σημεία συγκεντρωμένα γύρω από το −1.0, υποδηλώνοντας ομοιόμορφα αρνητική επίδραση ανεξαρτήτως της τιμής. Το Bwd Packet Length Max παρουσιάζει διπλό μοτίβο με μπλε σημεία αριστερά (≈ −0.4) και κόκκινα σημεία δεξιά (έως +0.7), δείχνοντας ότι οι υψηλές τιμές μέγιστου μεγέθους των πακέτων προς τα πίσω αυξάνουν την πιθανότητα ανίχνευσης. Το Idle Mean εμφανίζει κόκκινα σημεία δεξιά (έως +0.7) υποδηλώνοντας ότι υψηλές τιμές χρόνου αδράνειας συμβάλλουν θετικά στην αναγνώριση επίθεσης, σε αντίθεση με τις χαμηλές. Τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size παρουσιάζουν μηδενικές SHAP τιμές, συμβάλλοντας ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 138), το Bwd Packet Length Std παρουσιάζει κυρίως κόκκινα και μωβ σημεία αριστερά (≈ −6.0), υποδηλώνοντας ότι υψηλές τιμές τυπικής απόκλισης του μεγέθους των backward πακέτων μειώνουν δραστικά την πιθανότητα αναγνώρισης ως της κλάσης. Το Fwd Header Length.1 παρουσιάζει διπλό μοτίβο με μπλε και κόκκινα σημεία αριστερά και μπλε σημεία δεξιά (έως +4.5), που σημαίνει γενικά ότι οι χαμηλές τιμές μήκους επικεφαλίδας αυξάνουν την πιθανότητα κανονικής κίνησης. Τα Subflow Bwd Packets και Total Fwd Packets δείχνουν ότι οι χαμηλές τιμές του αριθμού των πακέτων σχετίζονται θετικά με BENIGN κίνηση και το Idle Mean, παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −5) και μπλε σημεία κοντά στο μηδέν, υποδηλώνει ότι οι υψηλές τιμές του χρόνου αδράνειας μειώνουν σημαντικά την πιθανότητα αναγνώρισης κανονικής κίνησης. Τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size εμφανίζουν πρακτικά μηδενικές SHAP τιμές, συμβάλλοντας ελάχιστα στη διάκριση.

Για την κλάση 1 που αντιστοιχεί στην επίθεση DoS GoldenEye (Εικόνα 139), το Subflow Bwd Packets κυριαρχεί παρουσιάζοντας μπλε σημεία αριστερά (έως −6.0), υποδηλώνοντας ότι χαμηλές τιμές αριθμού backward πακέτων subflow μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd Header Length.1 παρουσιάζει επίσης μπλε σημεία αριστερά (≈ −3) και κόκκινα σημεία κοντά στο μηδέν, με ανάλογη ερμηνεία. Το Idle Mean

εμφανίζει διαφορετικό μοτίβο με μπλε σημεία γύρω στο −1.0 και κόκκινα σημεία δεξιά (διάσπαρτα έως +8.0), υποδηλώνοντας ότι οι υψηλές τιμές χρόνου αδράνειας αυξάνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης. Το Bwd IAT Std παρουσιάζει μπλε σημεία αριστερά και κόκκινα δεξιά (≈ +0.5), υποδηλώνοντας διπλό μοτίβο επίδρασης ανάλογα με την τιμή. Τα χαρακτηριστικά FIN Flag Count, Subflow Bwd Bytes και Avg Bwd Segment Size παρουσιάζουν ουσιαστικά μηδενική επίδραση.

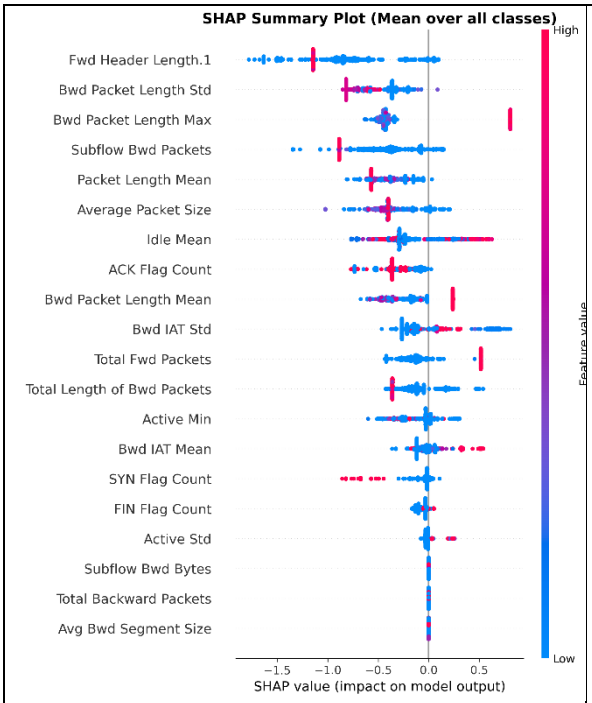
Για την κλάση 2 που αντιστοιχεί στην επίθεση DoS Hulk (Εικόνα 140), το Idle Mean εμφανίζεται με ισχυρό διπλό μοτίβο, όπου μπλε σημεία αριστερά (έως −3.0) και κόκκινα σημεία δεξιά (έως +5.0), σημαίνουν ότι οι υψηλές τιμές του μέσου χρόνου αδράνειας αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd Header Length.1 παρουσιάζει μπλε σημεία αριστερά (≈ −2.5), υποδηλώνοντας ότι οι χαμηλές τιμές του μήκους της επικεφαλίδας μειώνουν την πιθανότητα ανίχνευσης. Το SYN Flag Count εμφανίζει αποκλειστικά κόκκινα σημεία αριστερά (≈ −6.0), δείχνοντας ότι οι υψηλές τιμές SYN σημαιών μειώνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης, ενώ το FIN Flag Count παρουσιάζει κόκκινα σημεία δεξιά (≈ +2.5), υποδηλώνοντας ότι υψηλός αριθμός FIN σημαιών αποτελεί χαρακτηριστικό δείκτη αυτής της επίθεσης. Τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size πρακτικά εμφανίζουν μηδενική SHAP τιμή.

Για την κλάση 3 που αντιστοιχεί στην επίθεση DoS Slowhttptest (Εικόνα 141), το Subflow Bwd Packets παρουσιάζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (≈ +0.5), υποδηλώνοντας διπλό μοτίβο ανάλογα με την τιμή. Το Total Fwd Packets εμφανίζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά, δείχνοντας ότι οι υψηλές τιμές αριθμού των προωθούμενων πακέτων μειώνουν την πιθανότητα ανίχνευσης της κλάσης. Το Average Packet Size παρουσιάζει κόκκινα σημεία δεξιά (έως +1.5) και μπλε σημεία αριστερά, όπως επίσης το Active Min εμφανίζει κόκκινα σημεία δεξιά (έως +4.5), υποδηλώνοντας ότι η υψηλή ελάχιστη διάρκεια της ενεργής περιόδου αποτελεί χαρακτηριστικό δείκτη αυτής της κλάσης. Το Bwd IAT Std παρουσιάζει κόκκινα σημεία δεξιά (έως +4.5), που σημαίνει ότι η υψηλή μεταβλητότητα των χρονικών διαστημάτων των backward πακέτων σχετίζεται με την επίθεση αυτή. Τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size δεν επηρεάζουν την εκτίμηση.

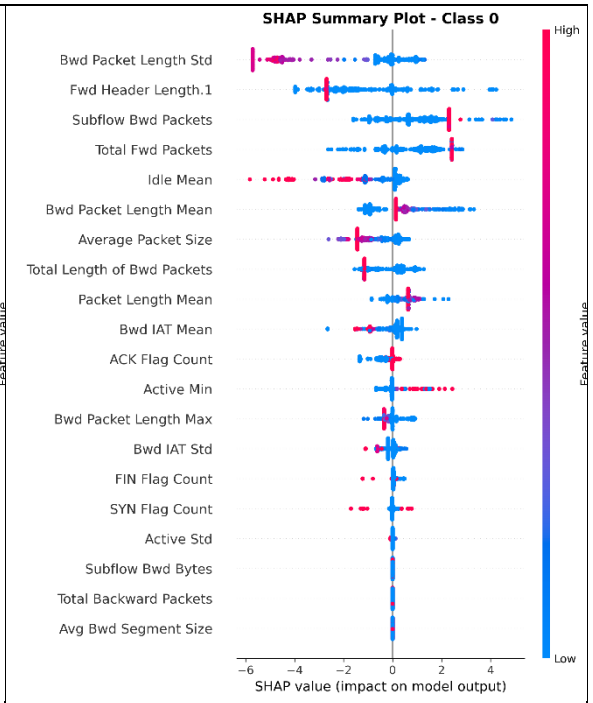
Για την κλάση 4 που αντιστοιχεί στην επίθεση DoS Slowloris (Εικόνα 142), το Bwd Packet Length Mean κυριαρχεί παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −2.0) και

μπλε σημεία δεξιά (≈ +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των backward πακέτων μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Ομοίως συμβαίνει και με το Average Packet Size το οποίο εμφανίζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (≈ +0.5). Το Bwd IAT Mean παρουσιάζει κόκκινα σημεία δεξιά (έως +4.0), δείχνοντας ότι οι υψηλές τιμές του μέσου χρονικού διαστήματος των backward πακέτων ενισχύουν την αναγνώριση της κλάσης. Το ACK Flag Count παρουσιάζει κόκκινα σημεία αριστερά (≈ −3.5) και μπλε σημεία δεξιά, υποδηλώνοντας ότι ο υψηλός αριθμός ACK σημαιών μειώνει την πιθανότητα αναγνώρισης.

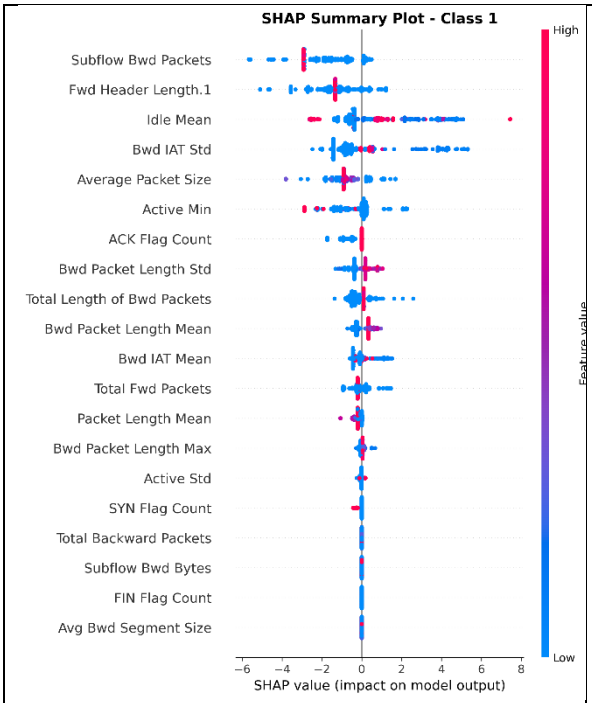
Για την κλάση 5 που αντιστοιχεί στην επίθεση Heartbleed (Εικόνα 143), το Bwd Packet Length Max κυριαρχεί με ένα αρκετά ισχυρό μοτίβο, με βάση το οποίο μπλε σημεία αριστερά (≈ −2.5) και ένα μεμονωμένο κόκκινο σημείο δεξιά (≈ +5.0), σημαίνουν ότι ένα μόνο πακέτο με εξαιρετικά υψηλό μέγιστο μέγεθος προς τα πίσω αποτελεί καθοριστικό δείκτη αναγνώρισης, ενώ οι χαμηλές τιμές μειώνουν σημαντικά αυτή την πιθανότητα. Το Total Fwd Packets και το Bwd Packet Length Mean παρουσιάζουν μπλε σημεία αριστερά (≈ −1) και μεμονωμένα κόκκινα σημεία δεξιά (≈ +3.5), υποδηλώνοντας ότι συγκεκριμένες ακραίες τιμές αποτελούν δευτερεύοντες δείκτες ανίχνευσης. Όλα τα υπόλοιπα χαρακτηριστικά (SYN/ACK Flag Count, Idle Mean, Fwd Header Length.1, Active Min, Bwd IAT Mean κ.λπ.) εμφανίζουν πρακτικά μηδενικές SHAP τιμές, κάνοντας σαφές ότι η ανίχνευση της κλάσης βασίζεται σχεδόν αποκλειστικά στο χαρακτηριστικό μέγεθος του backward πακέτου, αντικατοπτρίζοντας τον μηχανισμό της επίθεσης που εκμεταλλεύεται υπερμεγέθη TLS response για διαρροή μνήμης.



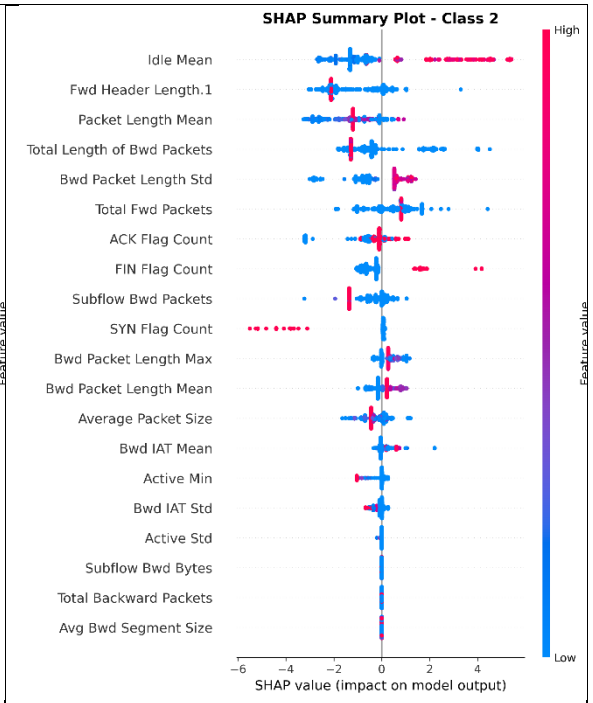
Εικόνα 137 Summary Plot, όλες οι κλάσεις για τον 10ο συνδυασμό.



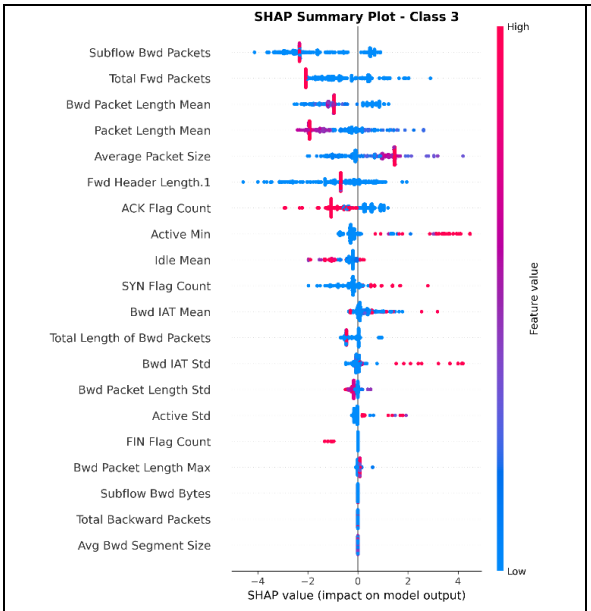
Εικόνα 138 SHAP Summary Plot, κλάση 0 για τον 10ο συνδυασμό.



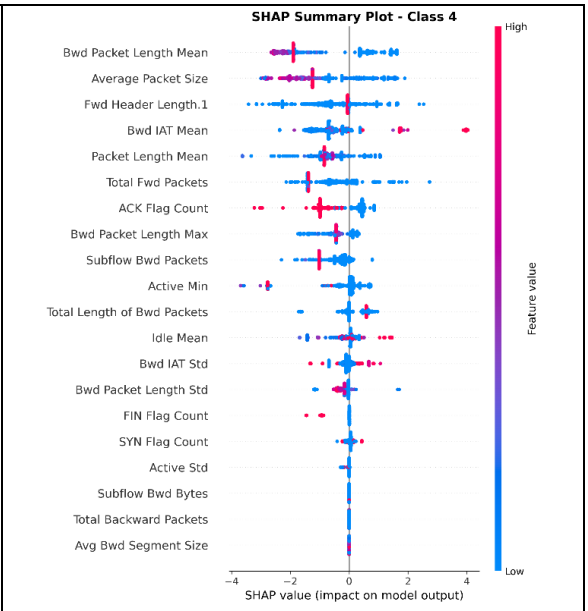
Εικόνα 139 SHAP Summary Plot, κλάση 1 για τον 10ο συνδυασμό.



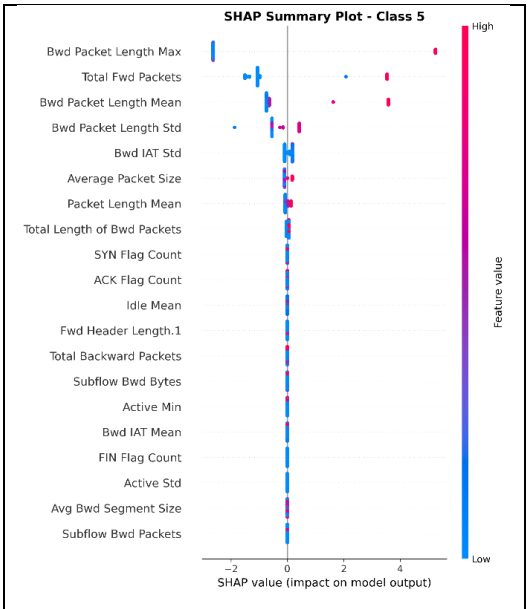
Εικόνα 140 SHAP Summary Plot, κλάση 2 για τον 10ο συνδυασμό.



Εικόνα 141 SHAP Summary Plot, κλάση 3 για τον 10ο συνδυασμό.



Εικόνα 142 SHAP Summary Plot, κλάση 4 για τον 10ο συνδυασμό.



Εικόνα 143 SHAP Summary Plot, κλάση 5 για τον 10ο συνδυασμό.

Στο συνολικό SHAP Summary Plot του ενδέκατου συνδυασμού (Εικόνα 144) παρατηρούμε ότι το Init_Win_bytes_backward κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας έντονο διπλό μοτίβο όπου, κόκκινα σημεία αριστερά (≈ −3.2) και μπλε σημεία δεξιά (έως +0.6), δείχνουν ότι οι υψηλές τιμές TCP backward παραθύρου μειώνουν

ενώ οι χαμηλές αυξάνουν την πιθανότητα ανίχνευσης κακόβουλης κίνησης. Το Fwd IAT Min εμφανίζει πυκνή συγκέντρωση σημείων αριστερά (≈ −0.5) κυρίως μπλε χρώματος, υποδηλώνοντας ότι οι χαμηλές τιμές ελάχιστου χρονικού διαστήματος μεταξύ των προωθούμενων πακέτων μειώνουν συνολικά την πιθανότητα αναγνώρισης της επίθεσης. Το Init_Win_bytes_forward παρουσιάζει κυρίως μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (≈ +0.5), ενώ το Fwd IAT Std εμφανίζει κόκκινα σημεία αριστερά (≈ −0.5) και κοντά στο μηδέν. Το act_data_pkt_fwd παρουσιάζει μπλε και κόκκινα σημεία δεξιά (έως +1), που σημαίνει ότι οι υψηλές τιμές των ενεργών πακέτων δεδομένων σχετίζονται με την αυξημένη πιθανότητα ανίχνευσης. Τα χαρακτηριστικά Total Fwd Packets, Avg Fwd Segment Size και Total Length of Bwd Packets, συμβάλλουν ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 145), το Init_Win_bytes_backward παρουσιάζει ισχυρό διπλό μοτίβο βάση του οποίου, κόκκινα σημεία αριστερά (≈ −4.0) και μπλε σημεία δεξιά (έως +4.0), δείχνουν ότι οι χαμηλές τιμές TCP backward παραθύρου αυξάνουν την πιθανότητα αναγνώρισης της κλάσης ως BENIGN, ενώ υψηλές τιμές τη μειώνουν. Το Fwd IAT Min παρουσιάζει κόκκινα και μπλε σημεία κοντά στο μηδέν αλλά και κόκκινα σημεία δεξιά (έως +8.5), υποδηλώνοντας ότι οι υψηλές τιμές ελάχιστου χρονικού διαστήματος αυξάνουν την πιθανότητα αναγνώρισης. Το act_data_pkt_fwd εμφανίζει μπλε σημεία αριστερά (έως -1.0) και μωβ σημεία κοντά στο μηδέν, δείχνοντας ότι οι χαμηλές τιμές των ενεργών πακέτων δεδομένων δεν σχετίζονται με κανονική κίνηση. Το Flow Bytes/s παρουσιάζει μπλε σημεία δεξιά (≈ +3.5), που σημαίνει ότι οι χαμηλές τιμές του ρυθμού των bytes αυξάνουν την πιθανότητα αναγνώρισης της κλάσης ως BANIGN. Τα χαρακτηριστικά PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets και Total Length of Bwd Packets συμβάλλουν ελάχιστα στη διάκριση.

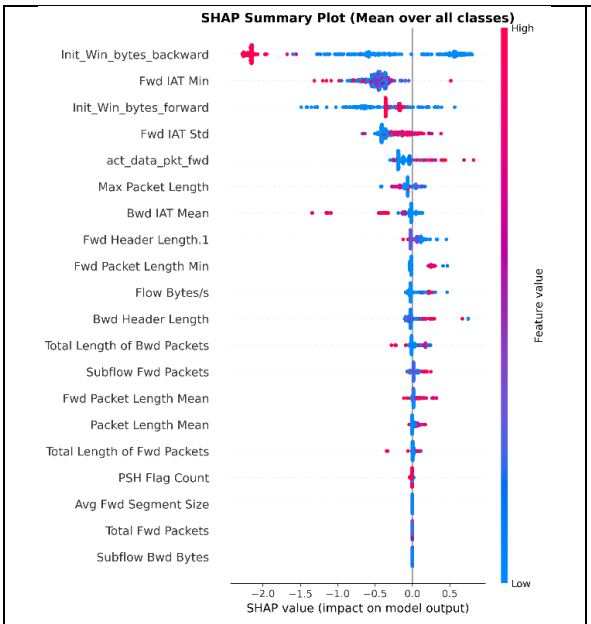
Για την κλάση 1 που αντιστοιχεί στο είδος επίθεσης Web Attack Brute Force (Εικόνα 146), το Fwd IAT Min κυριαρχεί παρουσιάζοντας ένα έντονο διπλό μοτίβο, στο οποίο παρουσιάζονται κόκκινα σημεία αριστερά (≈ −5.0) και μπλε και μωβ σημεία δεξιά (έως +2.5), υποδηλώνοντας ότι οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd IAT Std εμφανίζει διάσπαρτα κόκκινα σημεία αριστερά (≈ −2.5), γύρω από το μηδέν και δεξιά (έως +3.5), που σημαίνει ότι η υψηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης, γεγονός που αντικατοπτρίζει τον επαναληπτικό ρυθμό

αποστολής αιτημάτων της επίθεσης. Το Init_Win_bytes_backward παρουσιάζει μπλε σημεία αριστερά (≈ −2.0) και μεμονωμένα μπλε σημεία δεξιά. Το Fwd Packet Length Mean εμφανίζει κόκκινα σημεία δεξιά (≈ +2.0), υποδεικνύοντας ότι το υψηλό μέσο μέγεθος των προωθούμενων πακέτων αυξάνει την πιθανότητα αναγνώρισης. Τα χαρακτηριστικά PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets και Total Length of Bwd Packets εμφανίζουν πρακτικά μηδενική SHAP τιμή.

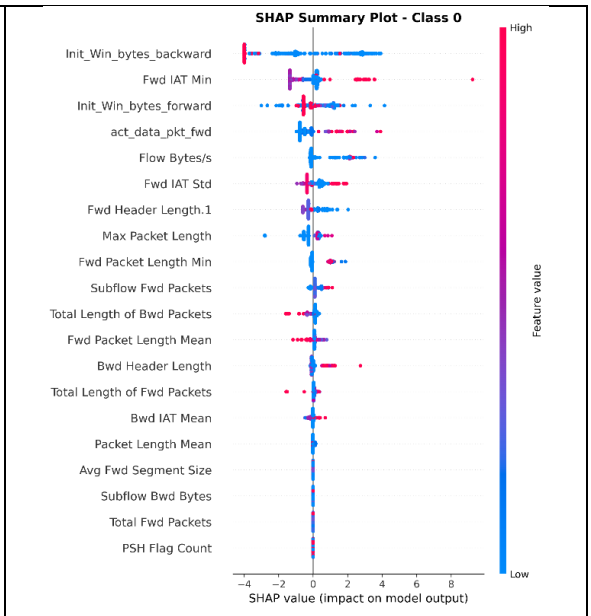
Για την κλάση 2 που αντιστοιχεί στην επίθεση Web Attack SQL Injection (Εικόνα 147), το Init_Win_bytes_backward επικρατεί με ισχυρό διπλό μοτίβο όπου, κόκκινα και μωβ σημεία αριστερά (≈ −6.0) και μπλε σημεία δεξιά (έως +8.0), υποδηλώνοντας ότι οι χαμηλές τιμές του TCP backward παραθύρου αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης, ενώ οι υψηλές τιμές τη μειώνουν. Το Init_Win_bytes_forward παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (≈ +2.5), δείχνοντας ότι οι χαμηλές τιμές του TCP forward παραθύρου αυξάνουν επίσης σημαντικά την πιθανότητα ανίχνευσης. Το Fwd IAT Min παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.2) και μπλε σημεία δεξιά (≈ +0.5), ενώ τα χαρακτηριστικά PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets, Packet Length Mean και Total Length of Bwd Packets εμφανίζουν πολύ μικρό εύρος SHAP τιμών, επομένως η SQL Injection αναγνωρίζεται κυρίως μέσω των χαρακτηριστικών του TCP παραθύρου.

Για την κλάση 3 που αντιστοιχεί στην επίθεση Web Attack XSS (Εικόνα 148), το Fwd IAT Min παρουσιάζει πυκνή συγκέντρωση μπλε και κόκκινων / μοβ σημείων δεξιά (έως +2.5) και κόκκινα σημεία αριστερά (≈ −2.5), υποδηλώνοντας σύνθετο μοτίβο σύμφωνα με το οποίο τόσο οι χαμηλές όσο και οι υψηλές τιμές μπορούν να επηρεάσουν την αναγνώριση της κλάσης. Το Init_Win_bytes_forward εμφανίζει μπλε σημεία αριστερά (≈ −4.0) και κόκκινα σημεία κοντά στο μηδέν, που σημαίνει ότι οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν σημαντικά την πιθανότητα αναγνώρισης. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (≈ −4.5) και κόκκινα σημεία δεξιά (≈ +2.5), υποδεικνύοντας ότι υψηλή μεταβλητότητα των χρονικών διαστημάτων επηρεάζει σημαντικά την πιθανότητα αναγνώρισης της XSS, συμπεριφορά από την οποία αντικατοπτρίζεται ο ασύμμετρος χαρακτήρας των XSS αιτημάτων. Το Bwd IAT Mean εμφανίζει κόκκινα σημεία αριστερά (≈ −0.5), επομένως οι υψηλές τιμές του μέσου χρονικού διαστήματος backward των πακέτων μειώνουν την πιθανότητα ανίχνευσης. Τα χαρακτηριστικά PSH Flag Count, Min Packet Length, Total Fwd Packets, Avg Fwd

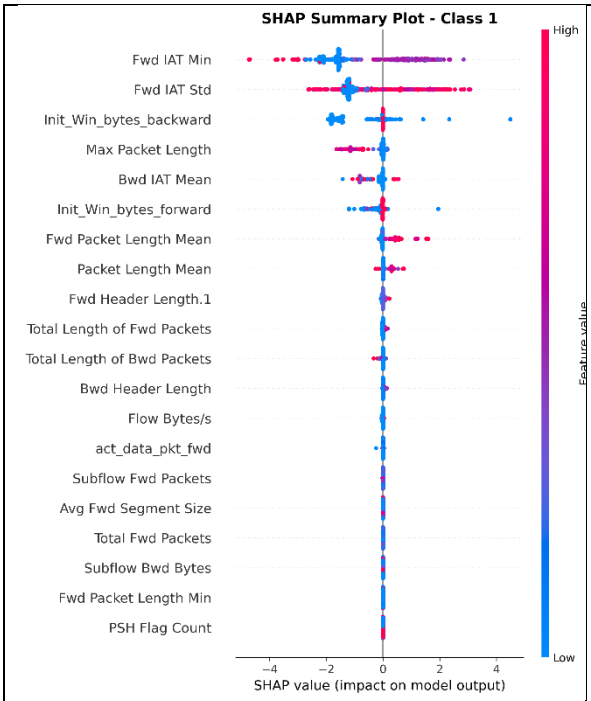
Segment Size και Total Length of Bwd Packets εμφανίζουν μηδενικές τιμές SHAP, συμβάλλοντας ελάχιστα στην αναγνώριση αυτής της κλάσης.



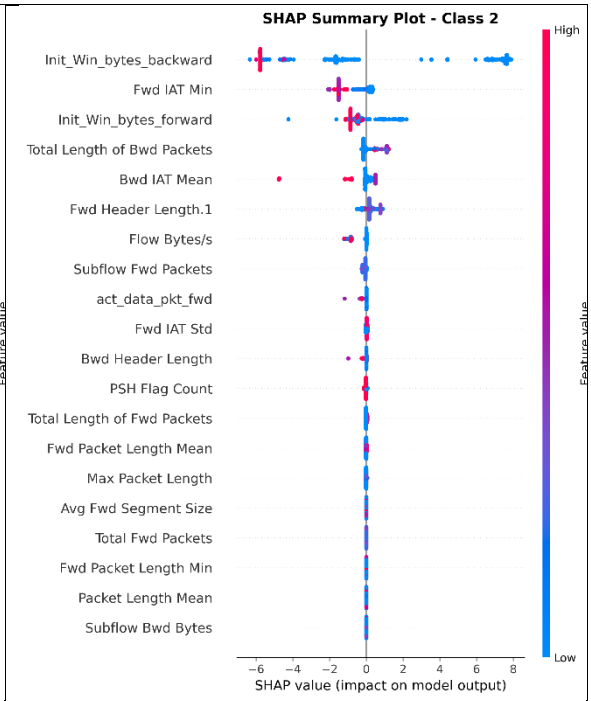
Εικόνα 144 Summary Plot, όλες οι κλάσεις για τον 11ο συνδυασμό.



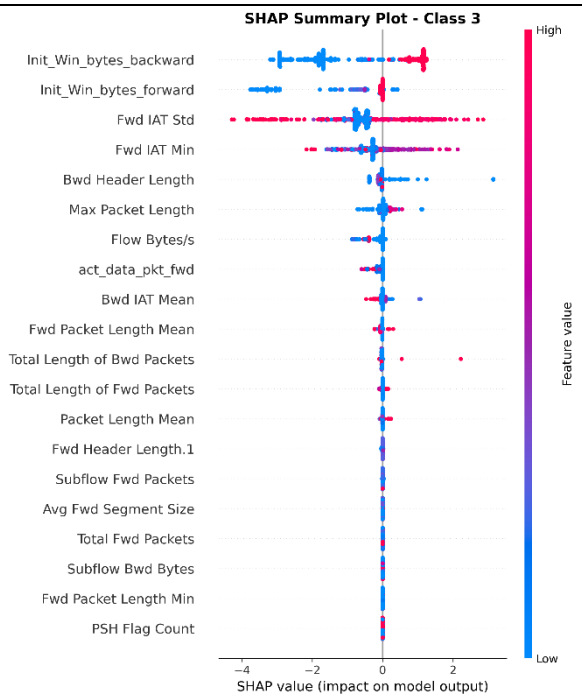
Εικόνα 145 SHAP Summary Plot, κλάση 0 για τον 11ο συνδυασμό.



Εικόνα 146 SHAP Summary Plot, κλάση 1 για τον 11ο συνδυασμό.



Εικόνα 147 SHAP Summary Plot, κλάση 2 για τον 11ο συνδυασμό.



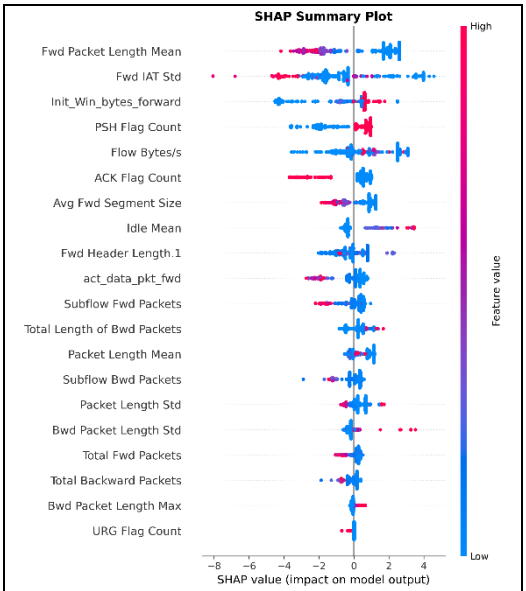
Εικόνα 148 SHAP Summary Plot, κλάση 3 για τον 11ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (φαίνεται στην Εικόνα 149) του δωδέκατου συνδυασμού παρατηρούμε ότι το Fwd Packet Length Mean κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας ένα μοτίβο κατά το οποίο, κόκκινα και μωβ σημεία εμφανίζονται αριστερά (≈ −4.2), μπλε σημεία δεξιά (έως +2.5) και μια μικρή διακοπή (≈ +1.0), υποδεικνύοντας ότι οι υψηλές τιμές του μέσου μεγέθους των προωθούμενων πακέτων μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της εν λόγο κλάσης. Υπάρχουν επίσης μπλε σημεία αριστερά (≈ −1), που σημαίνει ότι ορισμένες χαμηλές τιμές μπορούν επίσης να έχουν και αρνητική επίδραση ανάλογα, με την σύνολικη συμπεριφορά.

Το Flow Bytes/s εμφανίζει επίσης ένα διπλό μοτίβο με μπλε σημεία αριστερά (≈ −3.7) και διάσπαρτα κόκκινα σημεία δεξιά (έως +3.0), άρα ο υψηλός ρυθμός των bytes ανά δευτερόλεπτο αυξάνει την πιθανότητα αναγνώρισης κακόβουλης κίνησης, ενώ οι χαμηλές τιμές τη μειώνουν. Το Init_Win_bytes_forward παρουσιάζει κυρίως μπλε σημεία αριστερά (≈ −5.0) και κόκκινα σημεία δεξιά (≈ +2.0), υποδηλώνοντας έτσι ότι οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν ενώ οι υψηλές αυξάνουν την πιθανότητα ανίχνευσης.

Το ACK Flag Count εμφανίζει αποκλειστικά κόκκινα σημεία αριστερά (≈ από -1.0 έως −4.0) και μπλε σημεία δεξιά (≈ +2.5), από τα οποία προκύπτει ότι οι υψηλές τιμές των ACK σημαιών μειώνουν σημαντικά την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (≈ −4.5) και μπλε σημεία δεξιά (≈ +3.5), υποδηλώνοντας ότι η χαμηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης. Το act_data_pkt_fwd εμφανίζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (≈ +0.5), που σημαίνει ότι οι υψηλές τιμές των ενεργών πακέτων δεδομένων μειώνουν την πιθανότητα ανίχνευσης.

Το Avg Fwd Segment Size παρουσιάζει μοβ σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +1.5), ενώ το Fwd Header Length.1 εμφανίζει ελάχιστα κόκκινα σημεία αριστερά (≈ −0.5) και μπλε σημεία δεξιά (≈ +1.5). Το Subflow Bwd Packets παρουσιάζει μπλε / μωβ σημεία αριστερά και μπλε σημεία δεξιά, υποδηλώνοντας ότι χαμηλές τιμές αριθμού των backward πακέτων subflow μπορούν να έχουν και αρνητική επίδραση. Το Bwd Packet Length Std εμφανίζει μεμονωμένα κόκκινα σημεία δεξιά (≈ +3.8), δείχνοντας ότι οι ακραίες υψηλές τιμές τυπικής απόκλισης των backward πακέτων αποτελούν ισχυρό δείκτη κακόβουλης κίνησης. Τα χαρακτηριστικά URG Flag Count, Total Fwd Packets, Total Backward Packets και Active Std συμβάλλουν ελάχιστα στη διάκριση κανονικής από κακόβουλη κίνηση.



Εικόνα 149 Summary Plot, όλες οι κλάσεις για τον 12ο συνδυασμό.

Στο δέκατο τρίτο συνδυασμό, στο συνολικό SHAP Summary Plot (αποτυπώνει η Εικόνα 150) παρατηρούμε ότι το Bwd Packet Length Max παρουσιάζει ένα ιδιαίτερο διπλό μοτίβο κατά το οποίο εμφανίζονται μπλε και μωβ σημεία αριστερά (≈ −0.4) και κόκκινα σημεία δεξιά (≈ +0.7), υποδηλώνοντας ότι οι χαμηλές τιμές του μέγιστου μεγέθους των backward πακέτων μειώνουν ενώ υψηλές αυξάνουν την πιθανότητα ανίχνευσης. Το Bwd Packet Length Std εμφανίζει κόκκινα και μωβ σημεία αριστερά (≈ −0.8), κατά συνέπεια οι υψηλές τιμές της τυπικής απόκλισης των backward πακέτων μειώνουν συνολικά την πιθανότητα ανίχνευσης. Το Idle Mean παρουσιάζει επίσης ένα διπλό μοτίβο με μπλε σημεία αριστερά (≈ −0.8) και γύρο από το μηδέν και κόκκινα σημεία τόσο ανάμεσά τους όσο και δεξιά (≈ +0.6), γεγονός που υποδεικνύει ότι ο χρόνος αδράνειας επηρεάζει διαφορετικά την ανίχνευση ανάλογα με την κλάση. Το Bwd Packet Length Mean και το Fwd Header Length.1 εμφανίζουν κυρίως αρνητικές τιμές SHAP, ενώ το Bwd IAT Std παρουσιάζει μπλε σημεία δεξιά (έως +0.8), που σημαίνει ότι οι χαμηλές τιμές της τυπικής απόκλισης των backward χρονικών διαστημάτων σχετίζονται με αυξημένη πιθανότητα ανίχνευσης. Τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size παρουσιάζουν πρακτικά μηδενική τιμή SHAP, οπότε και συμβάλλουν ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 151), το Bwd Packet Length Std κυριαρχεί παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −6.0) και μπλε σημεία δεξιά (≈ +1.5), από τα οποία προκύπτει ότι οι υψηλές τιμές της τυπικής απόκλισης των backward πακέτων μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης ως BENIGN. Το Fwd Header Length.1 εμφανίζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία κοντά στο μηδέν και δεξιά (έως +4.5), που σημαίνει ότι οι χαμηλές τιμές του μήκους των επικεφαλίδων μπορούν να επηρεάσουν την αναγνώριση της κλάσης. Το Bwd Packet Length Mean εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +3.5), ενώ το Idle Mean παρουσιάζει κόκκινα σημεία αριστερά (≈ −6.0) και μπλε σημεία κοντά στο μηδέν, δείχνοντας με αυτό το τρόπο ότι οι υψηλές τιμές του χρόνου αδράνειας μειώνουν σημαντικά την πιθανότητα αναγνώρισης. Τα Total Fwd Packets και Subflow Bwd Packets εμφανίζουν μπλε σημεία δεξιά (≈ +2.0), υποδεικνύοντας ότι οι χαμηλές τιμές αριθμού των πακέτων σχετίζονται θετικά με την κανονική κίνηση.

Για την κλάση 1 που αντιστοιχεί στην επίθεση DoS GoldenEye (Εικόνα 152), το Bwd IAT Std παρουσιάζει το διπλό μοτίβο σύμφωνα με το οποίο, μπλε σημεία εμφανίζονται αριστερά

(≈ −2.0) και μπλε σημεία δεξιά (έως +5.5), υποδηλώνοντας σύνθετη επίδραση ανάλογα με την τιμή. Το Idle Mean εμφανίζει κόκκινα σημεία αριστερά (≈ −2.8), κόκκινα (≈ +2.0) και μπλε (≈ +2.0 έως +6.0) σημεία δεξιά, παρουσιάζοντας έντονο διπλό μοτίβο που υποδηλώνει ότι τόσο οι χαμηλές όσο και οι υψηλές τιμές του χρόνου αδράνειας μπορούν να επηρεάσουν σημαντικά την αναγνώριση της κλάσης. Το Subflow Bwd Packets παρουσιάζει κυρίως μπλε αλλά και κάποια κόκκινα σημεία αριστερά και μπλε σημεία κοντά στο μηδέν, γεγονός από το οποίο προκύπτει ότι οι χαμηλές τιμές μειώνουν την πιθανότητα ανίχνευσης. Το Fwd Header Length.1 εμφανίζει κυρίως μπλε σημεία αριστερά (≈ −2.5), με κάποια κόκκινα σημεία (≈ -1.8), ενώ τα χαρακτηριστικά FIN Flag Count, Subflow Bwd Bytes και Avg Bwd Segment Size παρουσιάζουν πρακτικά μηδενική SHAP τιμή.

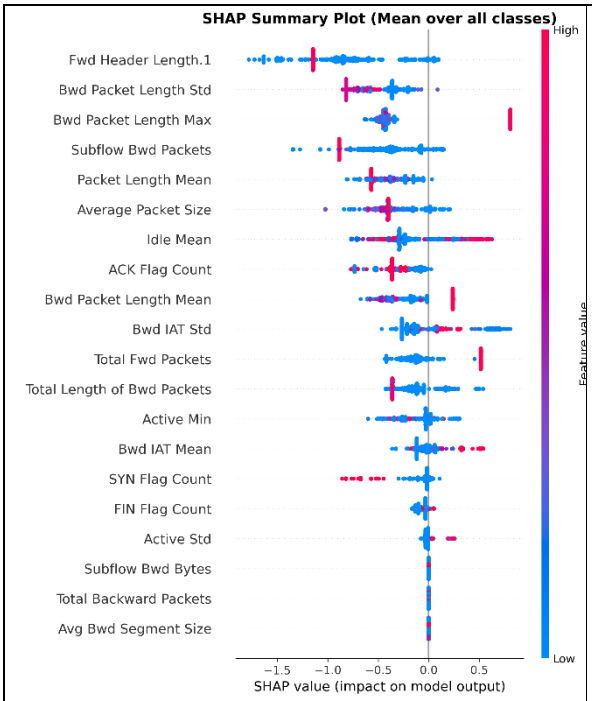
Για την κλάση 2 που αντιστοιχεί στην επίθεση DoS Hulk (φαίνεται στην Εικόνα 153), το Idle Mean κυριαρχεί με ισχυρό διπλό μοτίβο όπου, μπλε σημεία αριστερά (≈ −1.0) και κόκκινα σημεία δεξιά (έως +5.0), υποδηλώνουν ότι οι υψηλές τιμές του μέσου χρόνου αδράνειας αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Packet Length Mean παρουσιάζει μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία κοντά στο μηδέν, από τα οποία προκύπτει ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων μειώνουν την πιθανότητα ανίχνευσης. Το Bwd Packet Length Std εμφανίζει μπλε σημεία αριστερά (≈ −1.0) και κόκκινα σημεία δεξιά (≈ +1.5), ενώ το FIN Flag Count παρουσιάζει κόκκινα σημεία δεξιά (≈ +1.0 και έως +4.5), υποδεικνύοντας ότι ο υψηλός αριθμός των FIN σημαιών αποτελεί χαρακτηριστικό δείκτη αυτής της επίθεσης. Το SYN Flag Count εμφανίζει διάσπαρτα κόκκινα σημεία αριστερά (≈ −1.5), που σημαίνει ότι οι υψηλές τιμές των SYN σημαιών μειώνουν την πιθανότητα αναγνώρισης της συγκεκριμένης κλάσης. Τα χαρακτηριστικά Total Backward Packets, Subflow Bwd Bytes και Avg Bwd Segment Size δεν επηρεάζουν την ανίχνευση.

Για την κλάση 3 που αντιστοιχεί στην επίθεση DoS Slowhttptest (Εικόνα 154), το Subflow Bwd Packets παρουσιάζει μπλε σημεία αριστερά, κάποια λίγα κόκκινα αριστερά (≈ −2.0) και μπλε σημεία δεξιά, δείχνοντας ότι πρόκειται για ακόμα ένα διπλό μοτίβο του οποίου η επιρροή είναι ανάλογα με την τιμή. Το Bwd Packet Length Mean εμφανίζει κυρίως μπλε σημεία αριστερά (≈ −2.0) και δεξιά (≈ +0.5), ενώ το Total Fwd Packets παρουσιάζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (≈ +1), από τα οποία προκύπτει ότι οι υψηλές τιμές των προωθούμενων πακέτων μειώνουν την πιθανότητα ανίχνευσης της κλάσης. Το Active Min εμφανίζει μπλε σημεία κοντά και γύρω από το μηδέν και κόκκινα

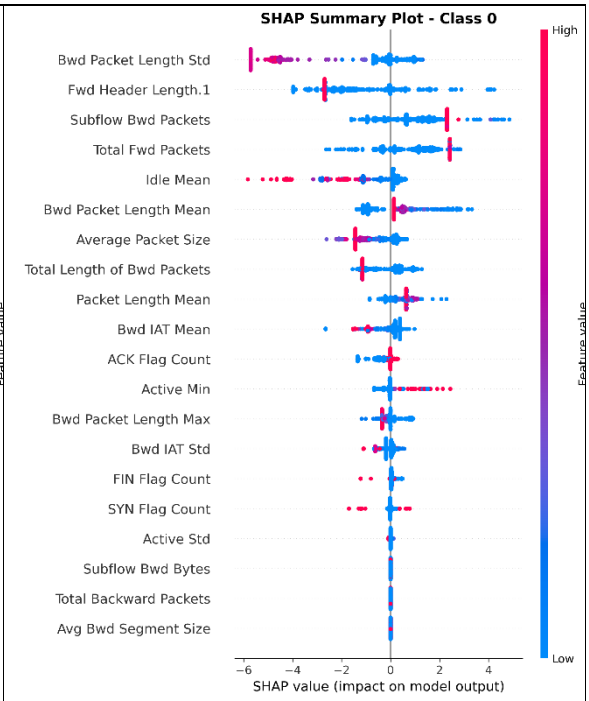
σημεία δεξιά (έως +4.5), υποδεικνύοντας ότι η υψηλή ελάχιστη διάρκεια της ενεργής περιόδου αποτελεί χαρακτηριστικό δείκτη αυτής της κλάσης. Το Bwd IAT Std παρουσιάζει διάσπαρτα κόκκινα σημεία δεξιά (≈ έως +4), οπότε προκύπτει ότι η υψηλή μεταβλητότητα των backward χρονικών διαστημάτων σχετίζεται επίσης με τη κλάση.

Στην κλάση 4 που αντιστοιχεί στην επίθεση DoS Slowloris (Εικόνα 155), το Bwd Packet Length Mean κυριαρχεί παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (≈ +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των backward πακέτων μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης αυτής. Το Average Packet Size εμφανίζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (≈ +0.5), ενώ Το Bwd IAT Mean παρουσιάζει κόκκινα σημεία δεξιά (έως +4), δείχνοντας ότι οι υψηλές τιμές του μέσου backward χρονικού διαστήματος αυξάνουν την πιθανότητα της ανίχνευσης. Το ACK Flag Count εμφανίζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά γύρω από το μηδέν (≈ +0.3), υποδηλώνοντας ότι ο υψηλός αριθμός των ACK σημαιών μειώνει την πιθανότητα αναγνώρισης. Τα χαρακτηριστικά Active Std, Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size ακούν μηδενική επιρροή.

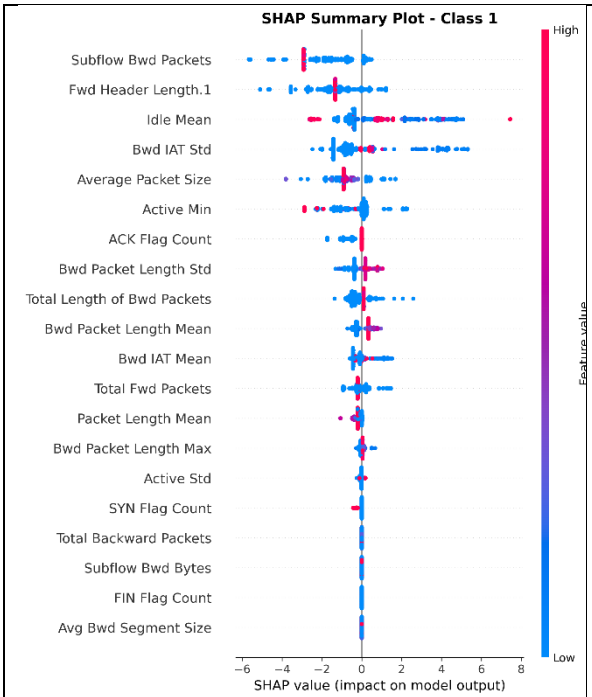
Για την κλάση 5 που αντιστοιχεί στην επίθεση Heartbleed (Εικόνα 156), το Bwd Packet Length Max κυριαρχεί αποκλειστικά με άξιο αναφοράς διπλό μοτίβο στο οποίο εμφανίζονται μπλε σημεία αριστερά (≈ −2.5) και ένα μεμονωμένο κόκκινο σημείο δεξιά (≈ +4.5), υποδηλώνοντας ότι ένα μόνο πακέτο με εξαιρετικά υψηλό μέγιστο μέγεθος προς τα πίσω αποτελεί καθοριστικό δείκτη της κλάσης, ενώ οι χαμηλές τιμές μειώνουν δραστικά την πιθανότητα αναγνώρισης. Το Bwd Packet Length Mean παρουσιάζει μπλε σημεία αριστερά (≈ −0.7) και μεμονωμένα κόκκινα σημεία δεξιά, δείχνοντας με αυτό το τρόπο ότι οι ακραίες τιμές τους αποτελούν δευτερεύοντες δείκτες ανίχνευσης. Το Bwd Packet Length Std εμφανίζει μπλε σημεία αριστερά (≈ −1.0) και κόκκινα σημεία δεξιά (≈ +1.0). Όλα τα υπόλοιπα χαρακτηριστικά εμφανίζουν πρακτικά μηδενική τιμή SHAP, οπότε και προκύπτει ότι η ανίχνευση της κλάσης βασίζεται σχεδόν αποκλειστικά στο χαρακτηριστικό μέγεθος του backward πακέτου, αντικατοπτρίζοντας έτσι τον μηχανισμό της επίθεσης Heartbleed που εκμεταλλεύεται τα υπερμεγέθη TLS response για διαρροή μνήμης.



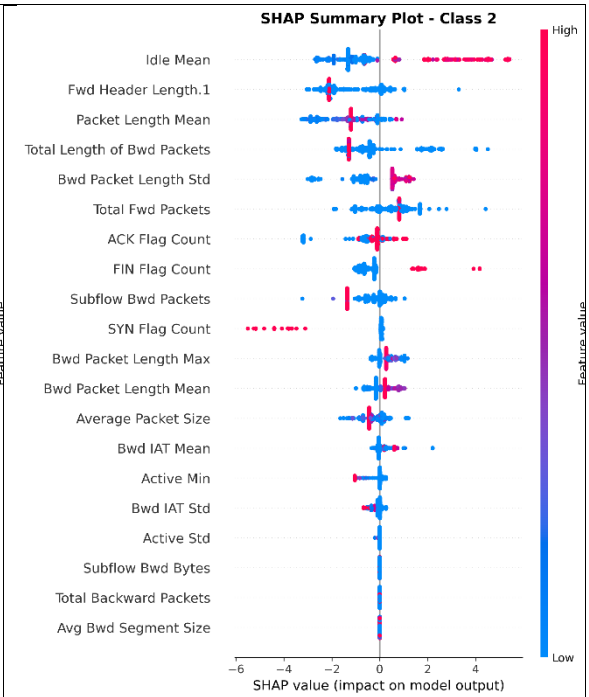
Εικόνα 150 Summary Plot, όλες οι κλάσεις για τον 13ο συνδυασμό.



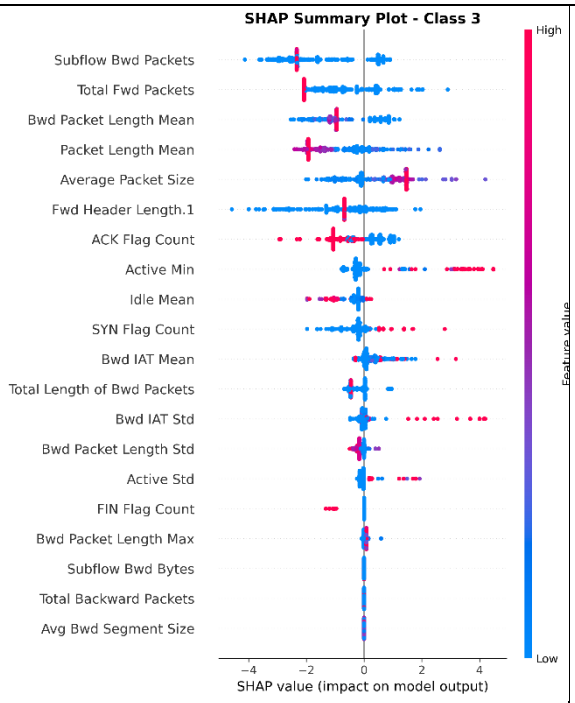
Εικόνα 151 SHAP Summary Plot, κλάση 0 για τον 13ο συνδυασμό.



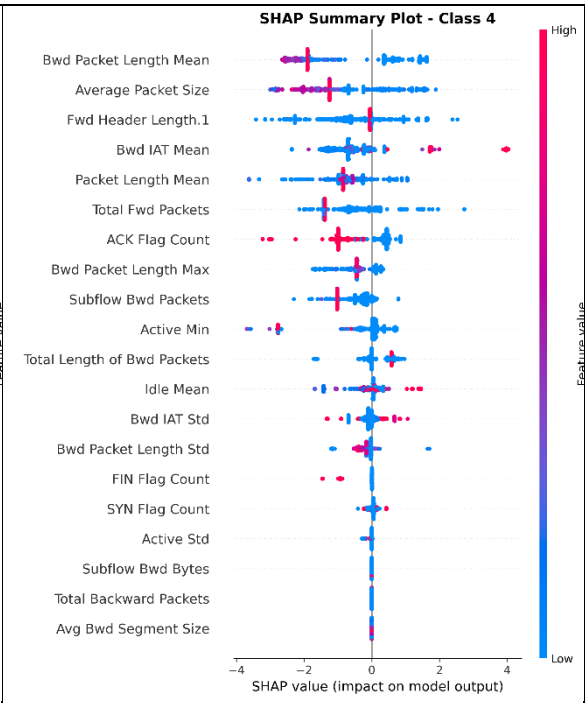
Εικόνα 152 SHAP Summary Plot, κλάση 1 για τον 13ο συνδυασμό.



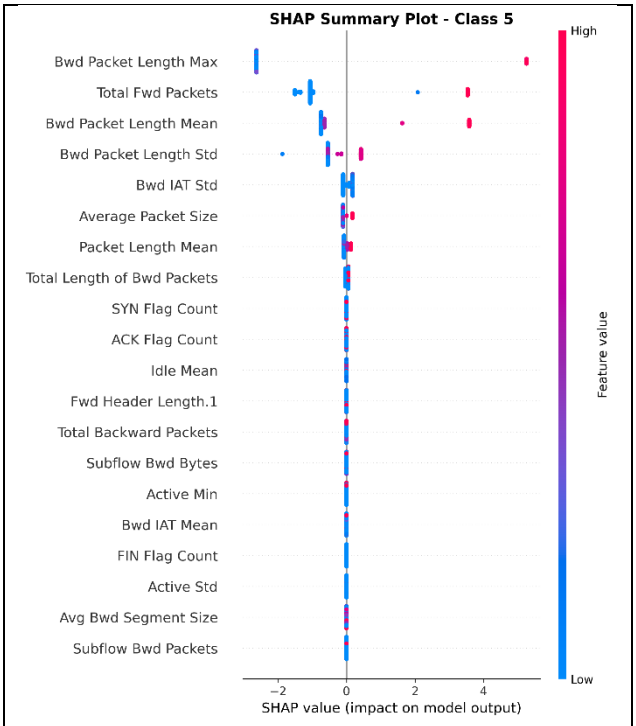
Εικόνα 153 SHAP Summary Plot, κλάση 2 για τον 13ο συνδυασμό.



Εικόνα 154 SHAP Summary Plot, κλάση 3 για τον 13ο συνδυασμό.



Εικόνα 155 SHAP Summary Plot, κλάση 4 για τον 13ο συνδυασμό.



Εικόνα 156 SHAP Summary Plot, κλάση 5 για τον 13ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (αποτυπώνει η Εικόνα 157) του δέκατου τέταρτου συνδυασμού παρατηρούμε ότι το Init_Win_bytes_backward κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −2.8) και μπλε σημεία δεξιά (έως +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του TCP backward παραθύρου μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα ανίχνευσης κακόβουλης κίνησης. Το Fwd IAT Min εμφανίζει πυκνή συγκέντρωση κόκκινων και μωβ σημείων αριστερά (≈ −0.5) και μπλε σημεία κοντά στο μηδέν, δείχνοντας ότι οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος μεταξύ των προωθούμενων πακέτων μειώνουν συνολικά την πιθανότητα αναγνώρισης μιας επίθεσης. Το Init_Win_bytes_forward παρουσιάζει μπλε σημεία αριστερά (≈ −0.8) και κόκκινα σημεία κοντά στο μηδέν, ενώ το Fwd IAT Std εμφανίζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε/κόκκινα σημεία κοντά στο μηδέν. Το act_data_pkt_fwd παρουσιάζει μπλε και μωβ σημεία δεξιά (έως +0.9), από το οποίο προκύπτει ότι οι χαμηλές τιμές των ενεργών πακέτων δεδομένων σχετίζονται με αυξημένη πιθανότητα ανίχνευσης. Τα χαρακτηριστικά Total Fwd Packets, Avg Fwd Segment Size και Total Length of Bwd Packets συμβάλλουν ελάχιστα στη συνολική απόφαση.

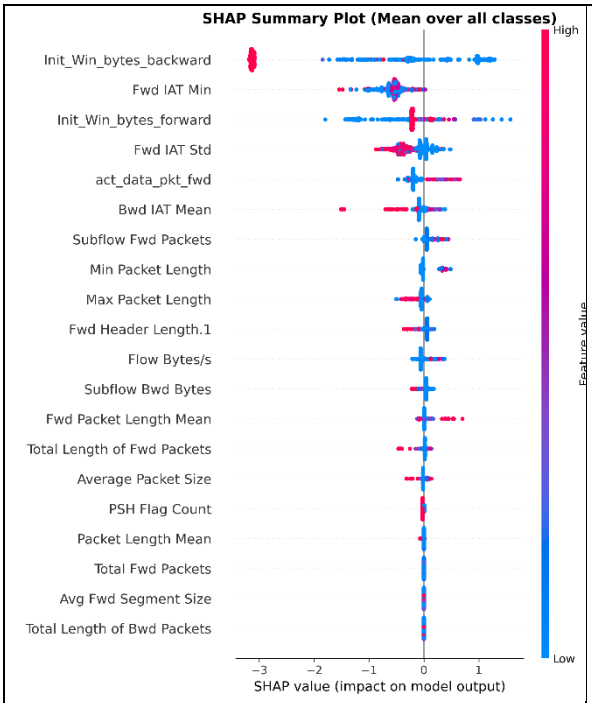
Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 158), το Init_Win_bytes_backward κυριαρχεί παρουσιάζοντας ισχυρό διπλό μοτίβο κατά το οποίο εμφανίζονται κόκκινα σημεία αριστερά (≈ −6.5) και μπλε σημεία δεξιά (έως +6.0), που σημαίνει ότι οι χαμηλές τιμές του TCP backward παραθύρου αυξάνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης ως BENIGN, ενώ οι υψηλές τιμές τη μειώνουν. Το Fwd IAT Min παρουσιάζει επίσης διπλό μοτίβο με κόκκινα σημεία αριστερά (≈ −1.5), πυκνή συγκέντρωση κοντά στο μηδέν και μπλε σημεία δεξιά (έως +6.0), υποδεικνύοντας ότι οι χαμηλές τιμές του ελάχιστου χρονικού διαστήματος αυξάνουν σημαντικά την πιθανότητα αναγνώρισης. Το Init_Win_bytes_forward εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +1.5), ενώ το act_data_pkt_fwd παρουσιάζει μπλε και μωβ σημεία δεξιά (≈ +3.5), που σημαίνει ότι χαμηλές τιμές ενεργών πακέτων δεδομένων αυξάνουν την πιθανότητα κανονικής κίνησης. Το Flow Bytes/s κυρίως εμφανίζει μπλε σημεία δεξιά (≈ +3.0) ενώ τα χαρακτηριστικά PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets και Total Length of Bwd Packets εμφανίζουν μηδενική τιμή SHAP.

Για την κλάση 1 που αντιστοιχεί στην επίθεση Web Attack Brute Force (Εικόνα 159), το Fwd IAT Min κυριαρχεί παρουσιάζοντας έντονο διπλό μοτίβο στο οποίο εμφανίζονται, κόκκινα σημεία αριστερά (≈ −6.0) και μπλε / μωβ σημεία δεξιά (έως +2.0), υποδηλώνοντας

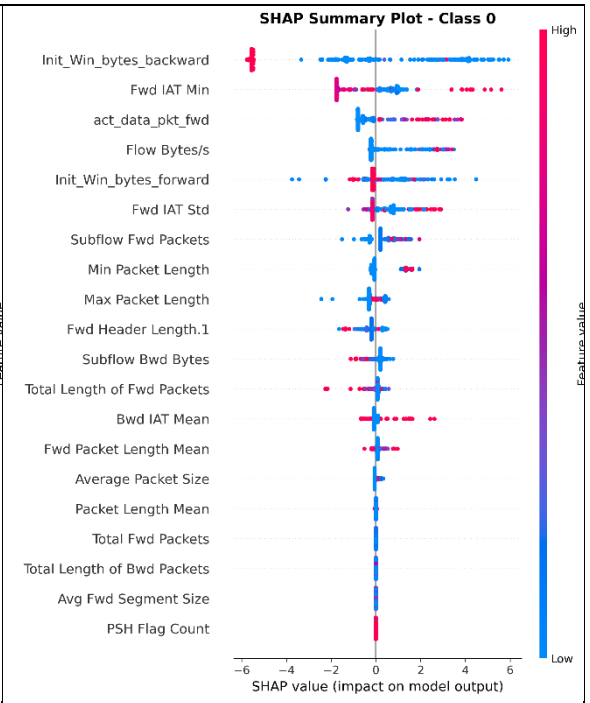
ότι οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Fwd IAT Std εμφανίζει κόκκινα σημεία αριστερά, λίγα μπλε σημεία γύρω από το μηδέν και κόκκινα δεξιά, γεγονός από το οποίο προκύπτει ότι η χαμηλή μεταβλητότητα των χρονικών διαστημάτων επηρεάζει σύνθετα την πιθανότητα αναγνώρισης, ανάλογα με την τιμή. Το Init_Win_bytes_backward παρουσιάζει μπλε σημεία αριστερά (≈ −2.5) και μεμονωμένα μπλε σημεία δεξιά (≈ +1.5 και +4.0), ενώ το Fwd Packet Length Mean εμφανίζει κόκκινα σημεία δεξιά (≈ +3.0), υποδεικνύοντας ότι το υψηλό μέσο μέγεθος των προωθούμενων πακέτων αυξάνει την πιθανότητα αναγνώρισης της εν λόγω κλάσης.

Για την κλάση 2 που αντιστοιχεί στην επίθεση Web Attack SQL Injection (Εικόνα 160), το Init_Win_bytes_backward εμφανίζεται με κόκκινα σημεία αριστερά (≈ −7.5) και μπλε σημεία (έως +7.0), υποδηλώνοντας ότι χαμηλές τιμές TCP backward παραθύρου αυξάνουν την πιθανότητα αναγνώρισης της κλάσης, ενώ υψηλές τιμές τη μειώνουν. Το Fwd IAT Min παρουσιάζει μωβ σημεία αριστερά (≈ −1.0) και μπλε σημεία δεξιά (≈ +2.5), ενώ το Init_Win_bytes_forward εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία κοντά στο μηδέν. Το Bwd IAT Mean εμφανίζει κόκκινα σημεία αριστερά και μπλε σημεία δεξιά ενώ τα χαρακτηριστικά PSH Flag Count, Fwd Packet Length Mean, Max Packet Length, Average Packet Size, Min Packet Length, Total Fwd Packets, Packet Length Mean, Avg Fwd Segment Size και Total Length of Bwd Packets εμφανίζουν πολύ μικρό εύρος τιμών SHAP, υποδηλώνοντας ότι η κλάση αναγνωρίζεται κυρίως μέσω του TCP backward παραθύρου.

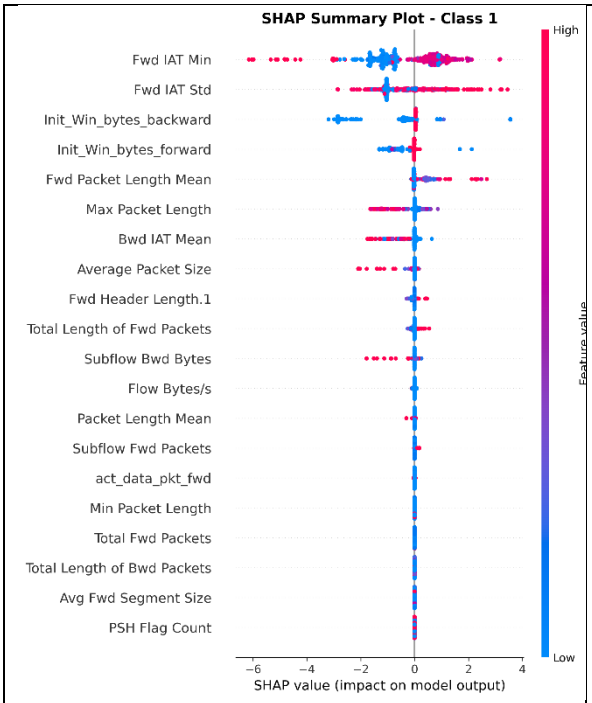
Για την κλάση 3 που αντιστοιχεί στην επίθεση Web Attack XSS (Εικόνα 161), το Fwd IAT Min παρουσιάζει πυκνή συγκέντρωση μπλε σημείων αριστερά (≈ −3.0) και κόκκινα και μωβ σημεία δεξιά (έως +2), υποδηλώνοντας σύνθετο μοτίβο με βάση το οποίο τόσο οι χαμηλές όσο και οι υψηλές τιμές μπορούν να επηρεάσουν την αναγνώριση της κλάσης. Το Init_Win_bytes_forward εμφανίζει μπλε σημεία αριστερά (≈ −5) και κόκκινα σημεία κοντά στο μηδέν, δείχνοντας ότι οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν δραστικά την πιθανότητα αναγνώρισης. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (≈ −6.0) και μπλε και κόκκινα σημεία δεξιά (≈ +2.5), που σημαίνει ότι η υψηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης XSS. Το Subflow Bwd Bytes παρουσιάζει διάσπαρτα κόκκινα σημεία δεξιά (έως +2.5), από τα οποία προκύπτει ότι οι υψηλές τιμές bytes subflow προς τα πίσω αυξάνουν ήπια την πιθανότητα αναγνώρισης.



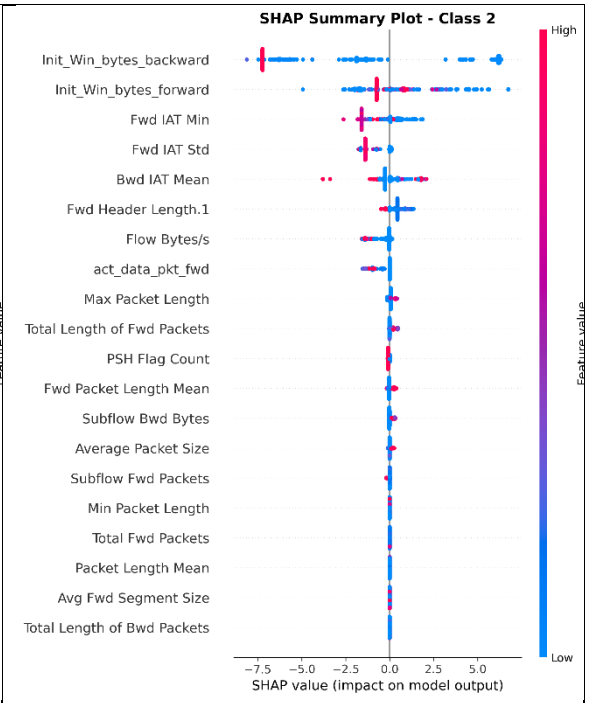
Εικόνα 157 Summary Plot, όλες οι κλάσεις για τον 14ο συνδυασμό.



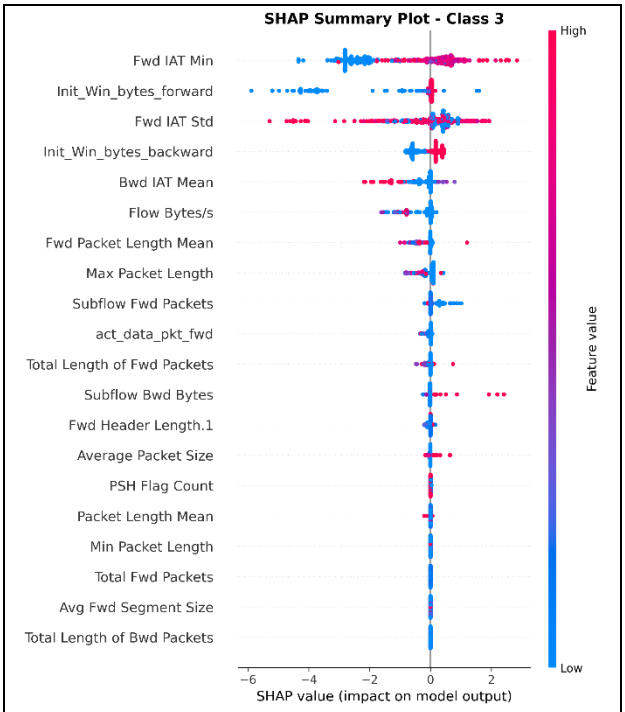
Εικόνα 158 SHAP Summary Plot, κλάση 0 για τον 14ο συνδυασμό.



Εικόνα 159 SHAP Summary Plot, κλάση 1 για τον 14ο συνδυασμό.



Εικόνα 160 SHAP Summary Plot, κλάση 2 για τον 14ο συνδυασμό.



Εικόνα 161 SHAP Summary Plot, κλάση 3 για τον 14ο συνδυασμό.

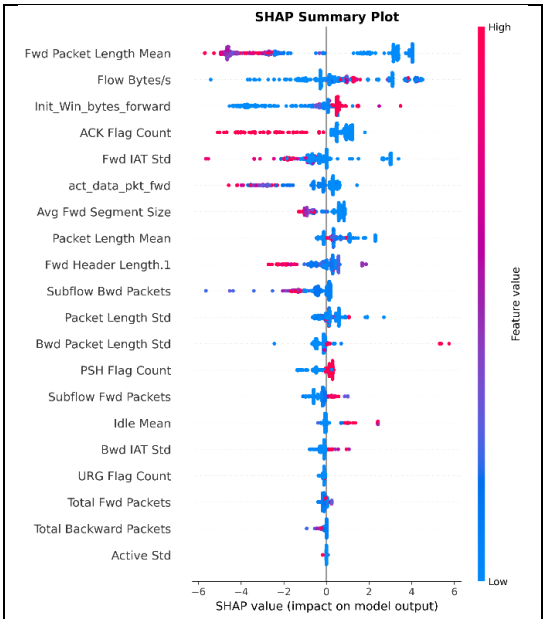
Στο δέκατο πέμπτο συνδυασμό, το συνολικό SHAP Summary Plot (όπως φαίνεται στην Εικόνα 162) για το δυαδικό μοντέλο ταξινόμησης παρατηρούμε ότι το Fwd Packet Length Mean κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας ένα μοτίβο στο οποίο κόκκινα και μωβ σημεία εμφανίζονται αριστερά (≈ −5.5) και πυκνή συγκέντρωση από μπλε σημεία δεξιά (έως +2.5) με μια μικρή διακοπή (≈ +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των προωθούμενων πακέτων μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Υπάρχουν επίσης μπλε σημεία αριστερά (≈ −1.5), από τα οποία προκύπτει ότι ορισμένες χαμηλές τιμές μπορούν επίσης να έχουν αρνητική επίδραση ανάλογα με την συνολική αξιολόγηση.

Το Init_Win_bytes_forward εμφανίζει κυρίως μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία δεξιά (≈ +0.5), υποδεικνύοντας ότι οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν ενώ οι υψηλές αυξάνουν την πιθανότητα ανίχνευσης. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (≈ +3), δείχνοντας με αυτό το τρόπο ότι και η χαμηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Flow Bytes/s εμφανίζει μπλε σημεία

αριστερά και κόκκινα σημεία δεξιά, υποδηλώνοντας ότι ο χαμηλός ρυθμός των bytes ανά δευτερόλεπτο μειώνει σημαντικά την πιθανότητα αναγνώρισης επίθεσης.

Το ACK Flag Count εμφανίζει αποκλειστικά κόκκινα σημεία αριστερά και μπλε σημεία δεξιά (≈ +1.0), υποδηλώνοντας ότι οι υψηλές τιμές των ACK σημαιών μειώνουν δραστικά την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Fwd Header Length.1 εμφανίζει κόκκινα κυρίως σημεία αριστερά (≈ −2.0) και μπλε / μωβ σημεία δεξιά, από το οποίο προκύπτει ότι οι υψηλές τιμές μήκους των επικεφαλίδων μειώνουν ενώ οι χαμηλές αυξάνουν ήπια την πιθανότητα ανίχνευσης.

Το Bwd Packet Length Std έχει μπλε σημεία κοντά στο μηδέν και διάσπαρτα κόκκινα σημεία δεξιά (≈ +0.5 και ≈ +6.0), υποδηλώνοντας ότι οι ακραίες υψηλές τιμές της τυπικής απόκλισης των backward πακέτων αποτελούν ισχυρό δείκτη κακόβουλης κίνησης. Το act_data_pkt_fwd εμφανίζει μωβ σημεία αριστερά και μπλε σημεία κοντά στο μηδέν, ενώ το Idle Mean παρουσιάζει μπλε σημεία κοντά στο μηδέν και κόκκινα σημεία δεξιά, από τα οποία προκύπτει ότι οι υψηλές τιμές του χρόνου αδράνειας αυξάνουν την πιθανότητα αναγνώρισης επίθεσης. Τα χαρακτηριστικά URG Flag Count, Active Std, Total Backward Packets, Total Fwd Packets και Avg Fwd Segment Size συμβάλλουν ελάχιστα στη διάκριση κανονικής από κακόβουλη κίνηση.



Εικόνα 162 Summary Plot, όλες οι κλάσεις για τον 15ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (Εικόνα 163) του δέκατου έκτου συνδυασμού παρατηρούμε ότι το Bwd Packet Length Std κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −0.8) και μπλε σημεία κοντά στο μηδέν, υποδηλώνοντας ότι υψηλές τιμές τυπικής απόκλισης των backward πακέτων μειώνουν συνολικά την πιθανότητα ανίχνευσης. Το Fwd Header Length.1 εμφανίζει μπλε σημεία αριστερά (≈ −1.0) και κάποια κόκκινα σημεία κοντά στο μηδέν, ενώ το Idle Mean παρουσιάζει διπλό μοτίβο με μπλε σημεία αριστερά (≈ −0.8) και κόκκινα σημεία δεξιά (≈ +0.7), δείχνοντας έτσι ότι ο χρόνος αδράνειας επηρεάζει διαφορετικά την ανίχνευση ανάλογα με την κλάση. Το Bwd Packet Length Max παρουσιάζει μπλε σημεία κοντά στο μηδέν και κόκκινα σημεία δεξιά (≈ +0.7), ενώ το Total Fwd Packets εμφανίζει μπλε σημεία αριστερά και κόκκινα σημεία δεξιά (≈ +0.2). Το Bwd IAT Std παρουσιάζει μπλε σημεία αριστερά και δεξιά (έως +0.7), από τα οποία φαίνεται μια πιο σύνθετη επίδραση, ανάλογα με την κλάση. Τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size, συμβάλλουν ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 164), το Bwd Packet Length Std παρουσιάζει κόκκινα και μωβ σημεία αριστερά (≈ −5.0) και μπλε σημεία δεξιά (≈ +1.5), που σημαίνει ότι οι υψηλές τιμές της τυπικής απόκλισης των backward πακέτων μειώνουν την πιθανότητα αναγνώρισης της κλάσης ως BENIGN. Το Idle Mean δείχνει κόκκινα σημεία αριστερά (≈ −4.0) και μπλε σημεία κοντά στο μηδέν, υποδηλώνοντας ότι οι υψηλές τιμές του χρόνου αδράνειας μειώνουν σημαντικά την πιθανότητα κανονικής κίνησης. Το Fwd Header Length.1 εμφανίζει κόκκινα σημεία αριστερά, κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +4.0), που σημαίνει ότι οι χαμηλές τιμές του μήκους της επικεφαλίδας αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Total Fwd Packets εμφανίζει επίσης συσπείρωση μπλε σημείων δεξιά (≈ +1.8), υποδεικνύοντας ότι οι χαμηλές τιμές του αριθμού των πακέτων σχετίζονται θετικά με κανονική κίνηση. Το Bwd Packet Length Mean παρουσιάζει κόκκινα σημεία αριστερά (≈ -1.0) και μπλε σημεία δεξιά (≈ +3), ενώ τα χαρακτηριστικά Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size εμφανίζουν πρακτικά μηδενική SHAP τιμή.

Για την κλάση 1 που αντιστοιχεί στην επίθεση DoS GoldenEye (Εικόνα 165), το Bwd IAT Std επικρατεί παρουσιάζοντας διπλό μοτίβο όπου, μπλε σημεία αριστερά (≈ −1.5) και μπλε και κόκκινα σημεία δεξιά (έως +5.0), υποδηλώνουν σύνθετη επίδραση ανάλογα με την τιμή.

Το Idle Mean εμφανίζει κόκκινα σημεία αριστερά και κόκκινα σημεία δεξιά, δείχνοντας έντονο διπλό μοτίβο όπου τόσο οι χαμηλές όσο και οι υψηλές τιμές του χρόνου αδράνειας επηρεάζουν σημαντικά την αναγνώριση της κλάσης. Το Subflow Bwd Packets έχει κυρίως μπλε σημεία αριστερά, από τα οποία προκύπτει ότι οι χαμηλές τιμές μειώνουν την πιθανότητα ανίχνευσης. Το Fwd Header Length.1 εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία κοντά στο μηδέν, ενώ τα χαρακτηριστικά FIN Flag Count, Subflow Bwd Bytes και Avg Bwd Segment Size παρουσιάζουν πρακτικά μηδενική τιμή SHAP.

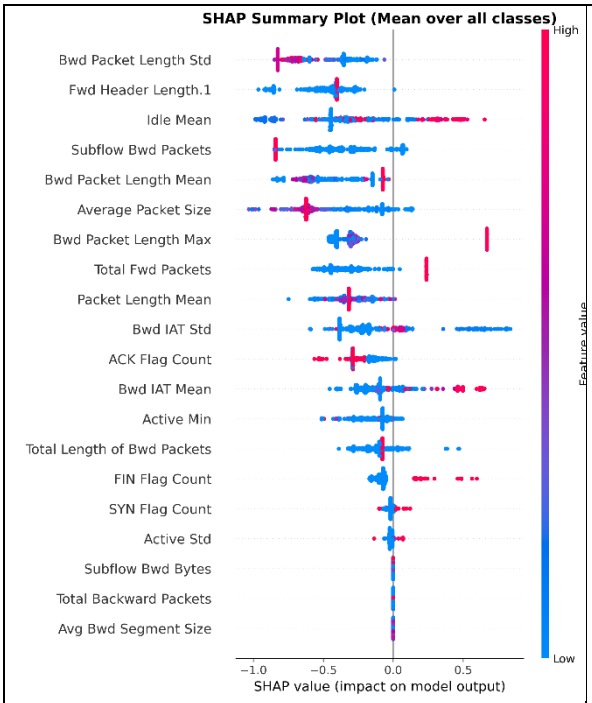
Για την κλάση 2 που αντιστοιχεί στην επίθεση DoS Hulk (Εικόνα 166), το Idle Mean κυριαρχεί με ισχυρό διπλό μοτίβο στο οποίο εμφανίζονται μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (έως +6.5), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου χρόνου αδράνειας αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης αυτής. Το Packet Length Mean παρουσιάζει κυρίως μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία κοντά στο μηδέν, το Bwd Packet Length Std εμφανίζει μπλε σημεία αριστερά και κόκκινα σημεία δεξιά, ενώ το FIN Flag Count παρουσιάζει κόκκινα σημεία δεξιά (≈ +2.5 και έως +6.5), από τα οποία προκύπτει ότι ο υψηλός αριθμός των FIN σημαιών αποτελεί χαρακτηριστικό δείκτη αυτής της επίθεσης. Το ACK Flag Count έχει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία κοντά στο μηδέν και τα χαρακτηστικά SYN Flag Count, Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size εμφανίζουν μηδενική SHAP τιμή.

Για την κλάση 3 που αντιστοιχεί στην επίθεση DoS Slowhttptest (βλέπε Εικόνα 167), το Bwd Packet Length Mean παρουσιάζει κόκκινα / μωβ σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (≈ +1.8), από τα οποία προκύπτει ακόμα ένα διπλό μοτίβο ανάλογα με την τιμή. Το Subflow Bwd Packets εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και κάποια κόκκινα σημεία πιο κοντά στο μηδέν. Το Active Min παρουσιάζει μπλε σημεία γύρω από το μηδέν και κόκκινα σημεία δεξιά (έως +4.5), υποδεικνύοντας ότι η υψηλή ελάχιστη διάρκεια της ενεργής περιόδου αποτελεί χαρακτηριστικό δείκτη αυτής της κλάσης. Το Bwd IAT Mean παρουσιάζει μπλε σημεία κοντά στο μηδέν και κόκκινα σημεία δεξιά (≈ +3.5), ενώ παρόμοια συμπεριφορά έχει και το Bwd IAT Std, υποδηλώνοντας ότι η υψηλή μεταβλητότητα των backward χρονικών διαστημάτων σχετίζεται με αυτή την κλάση.

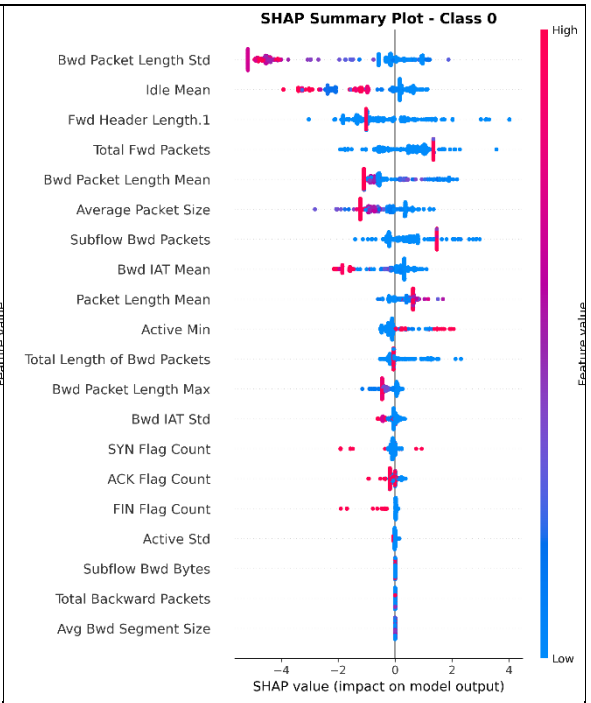
Για την κλάση 4 που αντιστοιχεί στην επίθεση DoS Slowloris (Εικόνα 168), το Average Packet Size κυριαρχεί παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (≈ +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των

πακέτων μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το Bwd IAT Mean εμφανίζει συστάδα μπλε σημείων αριστερά (≈ −1.0) και κόκκινα σημεία δεξιά (έως +6.0), από τα οποία προκύπτει ότι οι υψηλές τιμές του μέσου backward χρονικού διαστήματος αυξάνουν σημαντικά την πιθανότητα ανίχνευσης. Το Packet Length Mean παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (≈ +0.5), ενώ το ACK Flag Count εμφανίζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (≈ +0.3), δείχνοντας ότι ο υψηλός αριθμός των ACK σημαιών μειώνει την πιθανότητα αναγνώρισης. Τα χαρακτηριστικά Active Std, Subflow Bwd Bytes, Total Backward Packets και Avg Bwd Segment Size εμφανίζουν πρακτικά μηδενική τιμή SHAP.

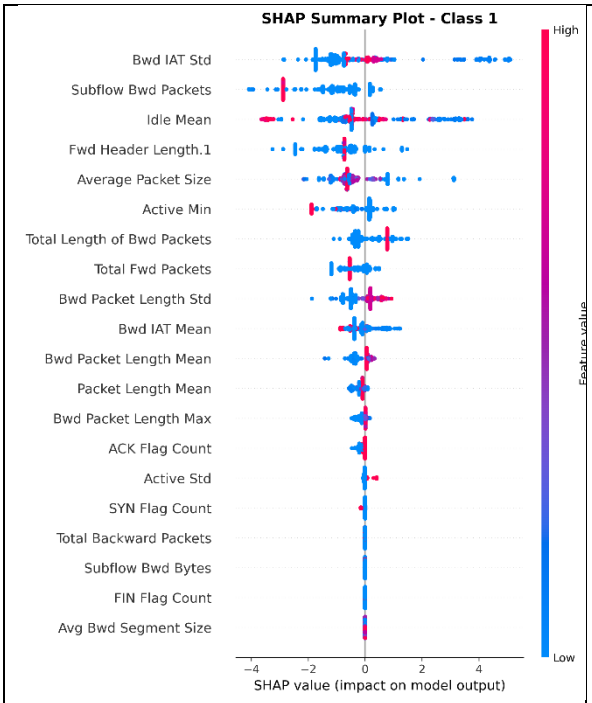
Για την κλάση 5 που αντιστοιχεί στην επίθεση Heartbleed (Εικόνα 169), το Bwd Packet Length Max κυριαρχεί αποκλειστικά με εξαιρετικά ισχυρό μοτίβο το οποίο παρουσιάζει μπλε και μωβ σημεία αριστερά (≈ −2.5) και ένα μεμονωμένο κόκκινο σημείο δεξιά (≈ +3.5), υποδηλώνοντας ότι ένα μόνο πακέτο με εξαιρετικά υψηλό μέγιστο μέγεθος προς τα πίσω αποτελεί καθοριστικό δείκτη αναγνώρισης της κλάσης, ενώ οι χαμηλές τιμές μειώνουν σημαντικά αυτή την πιθανότητα. Τα Total Fwd Packets και Bwd Packet Length Mean παρουσιάζουν επίσης μπλε σημεία αριστερά (≈ −1.5) και μεμονωμένα κόκκινα σημεία δεξιά (≈ +2.5), δείχνοντας ότι οι ακραίες τιμές τους αποτελούν δευτερεύοντες δείκτες ανίχνευσης. Το Fwd Header Length.1 εμφανίζει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (≈ +0.8), ενώ όλα τα υπόλοιπα χαρακτηριστικά εμφανίζουν μηδενική τιμή SHAP, κάνοντας σαφές ότι η ανίχνευση της κλάσης βασίζεται σχεδόν αποκλειστικά στο χαρακτηριστικό μέγεθος του backward πακέτου, αντικατοπτρίζοντας με αυτό το τρόπο τον μηχανισμό της επίθεσης που εκμεταλλεύεται τα υπερμεγέθη TLS response για διαρροή μνήμης.



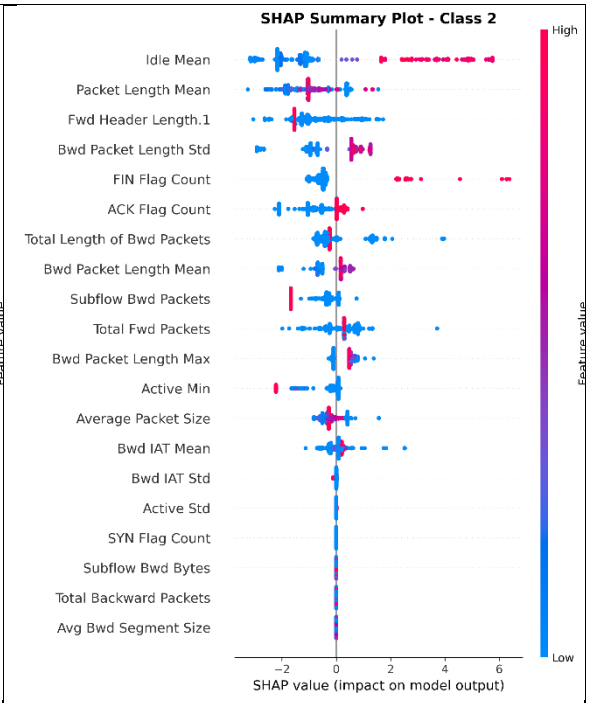
Εικόνα 163 Summary Plot, όλες οι κλάσεις για τον 16ο συνδυασμό.



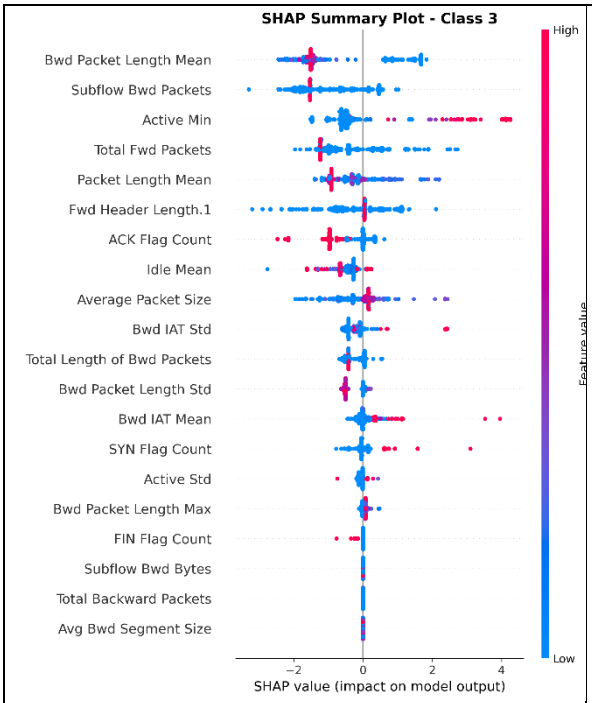
Εικόνα 164 SHAP Summary Plot, κλάση 0 για τον 16ο συνδυασμό.



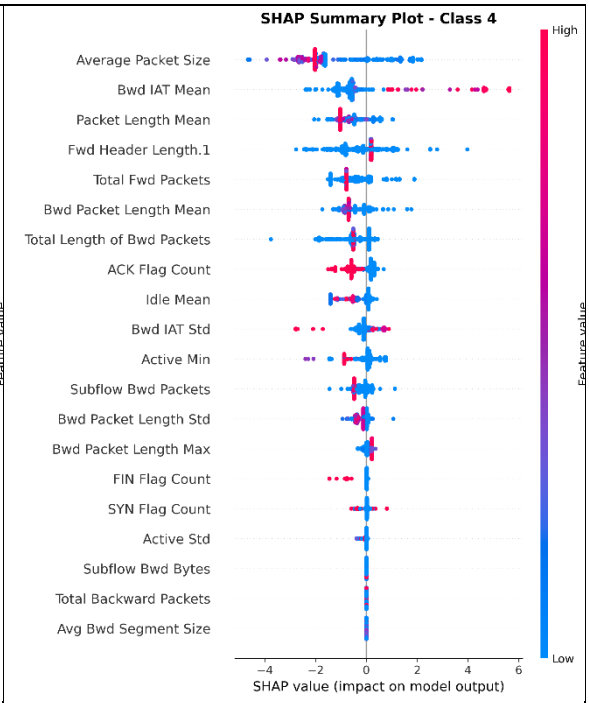
Εικόνα 165 SHAP Summary Plot, κλάση 1 για τον 16ο συνδυασμό.



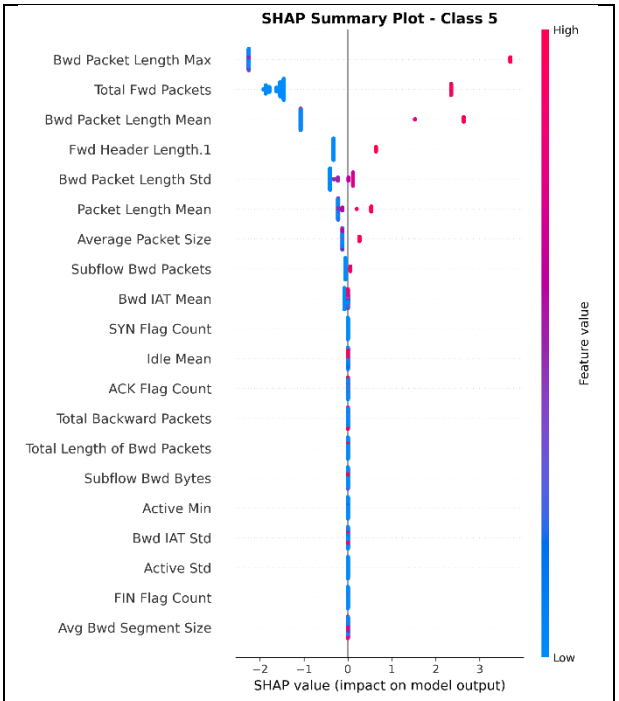
Εικόνα 166 SHAP Summary Plot, κλάση 2 για τον 16ο συνδυασμό.



Εικόνα 167 SHAP Summary Plot, κλάση 3 για τον 16ο συνδυασμό.



Εικόνα 168 SHAP Summary Plot, κλάση 4 για τον 16ο συνδυασμό.



Εικόνα 169 SHAP Summary Plot, κλάση 5 για τον 16ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (φαίνεται στην Εικόνα 170) του δέκατου έβδομου συνδυασμού παρατηρούμε ότι το Init_Win_bytes_backward επικρατεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας έντονο διπλό μοτίβο στο οποίο εμφανίζονται κόκκινα σημεία αριστερά (≈ −2.8) και μπλε σημεία δεξιά (έως +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του TCP backward παραθύρου μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα ανίχνευσης κακόβουλης κίνησης. Το Fwd IAT Min εμφανίζει πυκνή συγκέντρωση μπλε και κόκκινων σημείων αριστερά (≈ −0.7) και κόκκινα σημεία δεξιά (≈ +0.7), δείχνοντας ακόμα ένα σύνθετο μοτίβο επίδρασης, ανάλογα με την τιμή. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (≈ −0.5) και μπλε σημεία δεξιά (≈ +0.5), ενώ το Init_Win_bytes_forward εμφανίζει μπλε σημεία αριστερά (≈ −0.7) και κόκκινα σημεία κοντά στο μηδέν. Το act_data_pkt_fwd παρουσιάζει κόκκινα και μωβ σημεία δεξιά (έως +1.2), υποδηλώνοντας ότι υψηλές τιμές ενεργών πακέτων δεδομένων σχετίζονται με αυξημένη πιθανότητα ανίχνευσης. Τα χαρακτηριστικά Total Fwd Packets, Avg Fwd Segment Size και Total Length of Bwd Packets παρουσιάζουν πρακτικά μηδενική τιμή SHAP, συμβάλλοντας ελάχιστα στη συνολική απόφαση.

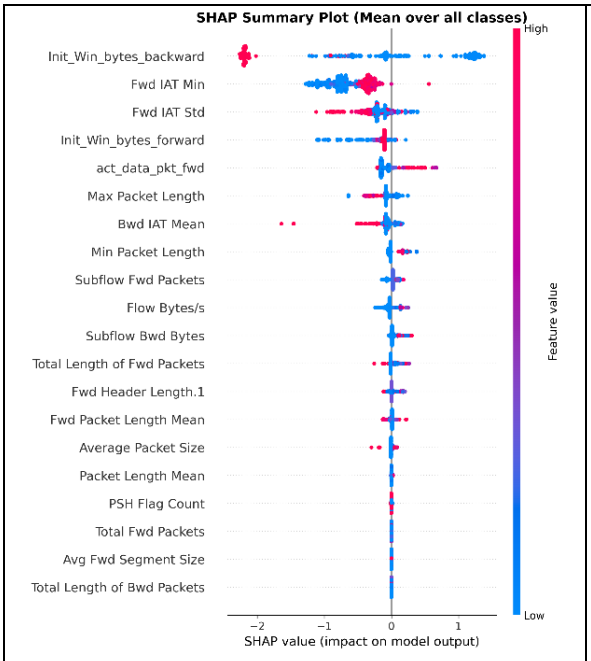
Για την κλάση 0 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 171), το Init_Win_bytes_backward κυριαρχεί παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −5.0) και μπλε σημεία δεξιά (έως +5), υποδηλώνοντας με αυτό το μοτίβο ότι οι χαμηλές τιμές του TCP backward παραθύρου αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης, ενώ υψηλές τιμές τη μειώνουν. Το Fwd IAT Min παρουσιάζεται με κόκκινα και μωβ σημεία κοντά στο μηδέν, μπλε (≈ +1.0) και κόκκινα (έως +6.5) σημεία δεξιά, δείχνοντας ότι οι χαμηλές τιμές του ελάχιστου χρονικού διαστήματος αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το Init_Win_bytes_forward εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά, ενώ το act_data_pkt_fwd παρουσιάζει κόκκινα και μωβ σημεία δεξιά (≈ +3.5), υποδηλώνοντας ότι οι υψηλές τιμές των ενεργών πακέτων δεδομένων αυξάνουν την πιθανότητα κανονικής κίνησης. Το Flow Bytes/s έχει μπλε σημεία δεξιά (≈ +2.5) και τα χαρακτηριστικά PSH Flag Count, Avg Fwd Segment Size, Total Fwd Packets και Total Length of Bwd Packets δεν επηρεάζουν την ανίχνευση.

Για την κλάση 1 που αντιστοιχεί στην επίθεση Web Attack Brute Force (Εικόνα 172), το Fwd IAT Min κυριαρχεί παρουσιάζοντας κυρίως κόκκινα σημεία δεξιά (≈ έως +3.0), υποδηλώνοντας ότι οι υψηλές τιμές του ελάχιστου χρονικού διαστήματος αυξάνουν ενώ χαμηλές μειώνουν την πιθανότητα αναγνώρισης της συγκεκριμένης κλάσης. Παρόμοια

συμπεριφορά παρουσιάζει και το Fwd IAT Std, δείχνοντας με αυτό το τρόπο ότι και η υψηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης. Το Init_Win_bytes_backward παρουσιάζει μπλε σημεία αριστερά (≈ −1.5) και μεμονωμένα μπλε σημεία δεξιά (≈ +1.8), ενώ το Max Packet Length εμφανίζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία κοντά στο μηδέν. Τα χαρακτηριστικά PSH Flag Count, Min Packet Length, Total Fwd Packets, Avg Fwd Segment Size και Total Length of Bwd Packets δεν ασκούν κάποια επιρροή.

Για την κλάση 2 που αντιστοιχεί στην επίθεση Web Attack SQL Injection (Εικόνα 173), το Init_Win_bytes_backward παρουσιάζεται με κόκκινα σημεία αριστερά (≈ −5.5) και μπλε σημεία δεξιά (έως +8.0), υποδηλώνοντας ότι οι χαμηλές τιμές του TCP backward παραθύρου αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης, ενώ οι υψηλές τιμές τη μειώνουν δραστικά. Το Fwd IAT Min παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.0) και μπλε σημεία δεξιά (≈ +0.5), ενώ το Init_Win_bytes_forward εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +2.0). Το Bwd IAT Mean παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.0) και μπλε σημεία δεξιά (≈ +0.5). Τα υπόλοιπα χαρακτηριστικά εμφανίζουν πολύ μικρό εύρος SHAP τιμών, υποδηλώνοντας ότι η SQL Injection αναγνωρίζεται κυρίως μέσω του TCP backward παραθύρου.

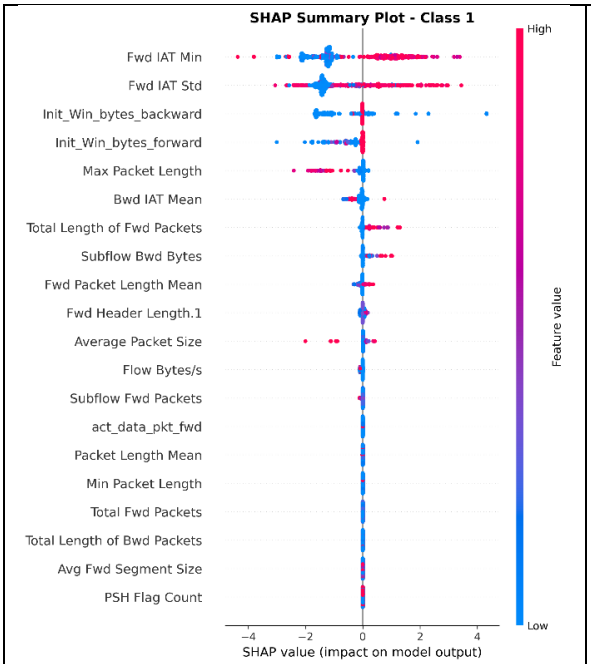
Για την κλάση 3 που αντιστοιχεί στην επίθεση Web Attack XSS (δείχνει η Εικόνα 174), κυριαρχεί το Fwd IAT Min παρουσιάζοντας πυκνή συγκέντρωση μπλε σημείων αριστερά (≈ −3.0) και κόκκινα και μωβ σημεία δεξιά (έως +2.0), υποδηλώνοντας ένα πιο σύνθετο μοτίβο με βάση το οποίο τόσο οι χαμηλές όσο και οι υψηλές τιμές μπορούν να επηρεάσουν την αναγνώριση της κλάσης. Το Init_Win_bytes_forward εμφανίζει μπλε σημεία αριστερά (≈ −3.0) και κόκκινα σημεία κοντά στο μηδέν, υποδεικνύοντας ότι οι χαμηλές τιμές του TCP forward παραθύρου μειώνουν σημαντικά την πιθανότητα αναγνώρισης. Το Fwd IAT Std παρουσιάζει κόκκινα σημεία αριστερά (≈ −5.5) και μπλε / κόκκινα σημεία δεξιά (≈ +2.5), υπονοώντας ότι η υψηλή μεταβλητότητα των χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης. Το Subflow Bwd Bytes παρουσιάζει κόκκινα σημεία δεξιά (≈ +2.0), δείχνοντας ότι και οι υψηλές τιμές των bytes subflow προς τα πίσω αυξάνουν ήπια την πιθανότητα αναγνώρισης. Τα χαρακτηριστικά PSH Flag Count, Min Packet Length, Total Fwd Packets, Avg Fwd Segment Size και Total Length of Bwd Packets συμβάλλουν ελάχιστα στην αναγνώριση αυτής της επίθεσης.



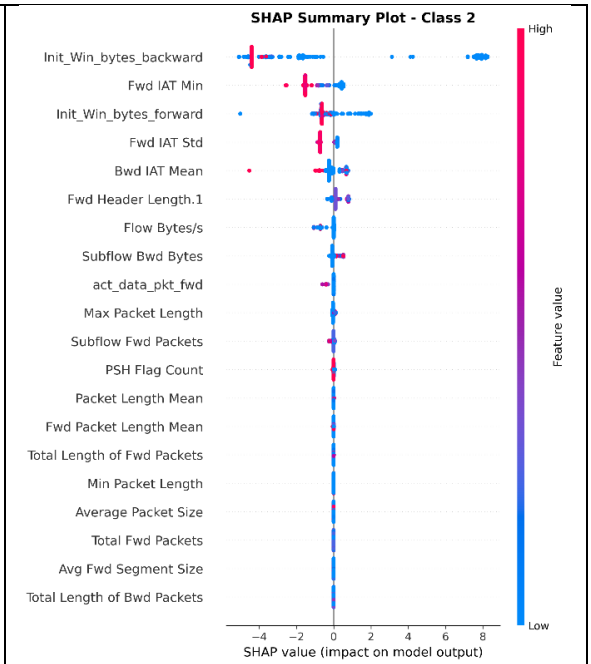
Εικόνα 170 Summary Plot, όλες οι κλάσεις για τον 17ο συνδυασμό.



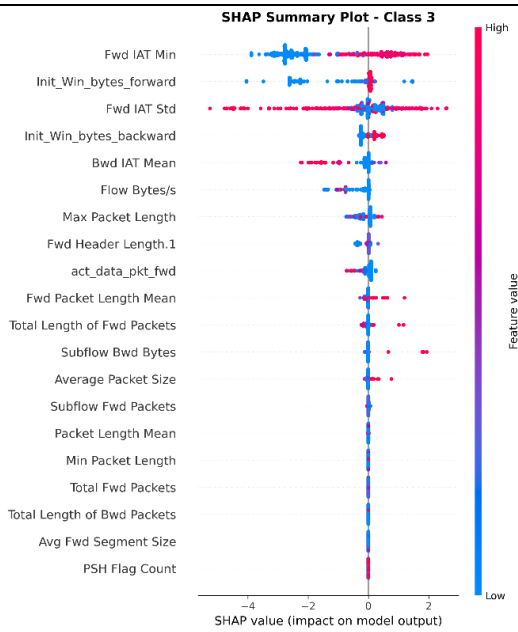
Εικόνα 171 SHAP Summary Plot, κλάση 0 για τον 17ο συνδυασμό.



Εικόνα 172 SHAP Summary Plot, κλάση 1 για τον 17ο συνδυασμό.



Εικόνα 173 SHAP Summary Plot, κλάση 2 για τον 17ο συνδυασμό.



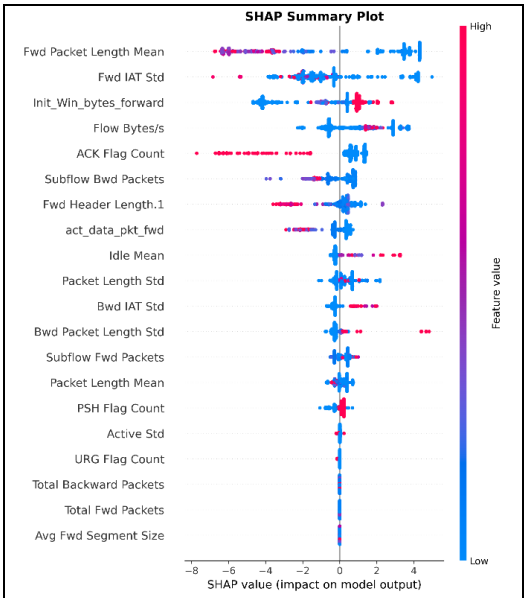
Εικόνα 174 SHAP Summary Plot, κλάση 3 για τον 17ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (όπως παρουσιάζεται στην Εικόνα 175) του δέκατου όγδοου συνδυασμού παρατηρούμε ότι το Fwd Packet Length Mean κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας σύνθετο μοτίβο το οποίο απαρτίζεται από κόκκινα και μωβ σημεία αριστερά (≈ −7) και πυκνή συγκέντρωση μπλε σημείων δεξιά (έως +2.5) με μια μικρή διακοπή (≈ +1.5), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των προωθούμενων πακέτων μειώνουν ενώ οι χαμηλές αυξάνουν την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Υπάρχουν επίσης μπλε σημεία αριστερά (≈ −2.0), υποδηλώνοντας επίσης ότι ορισμένες χαμηλές τιμές μπορούν να έχουν και αρνητική επίδραση ανάλογα με το σύνολο.

Το Fwd IAT Std εμφανίζει διπλό μοτίβο με κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (έως +4), υποδηλώνοντας ότι χαμηλή μεταβλητότητα χρονικών διαστημάτων αυξάνει την πιθανότητα αναγνώρισης κακόβουλης κίνησης, ενώ υψηλή τη μειώνει. Το Init_Win_bytes_forward παρουσιάζει κυρίως μπλε σημεία αριστερά (≈ −2) και κόκκινα σημεία δεξιά (≈ +1.5), υποδηλώνοντας ότι χαμηλές τιμές TCP forward παραθύρου μειώνουν ενώ υψηλές αυξάνουν την πιθανότητα ανίχνευσης επίθεσης. Το Flow Bytes/s εμφανίζει μπλε σημεία δεξιά (έως +4.5) και κόκκινα-μωβ (γύρω από το +2.0), δείχνοντας ότι οι χαμηλές τιμές του ρυθμού των bytes αυξάνουν ήπια την πιθανότητα ανίχνευσης.

Το ACK Flag Count εμφανίζει αποκλειστικά κόκκινα σημεία αριστερά (≈ −7.5) και μπλε σημεία δεξιά (≈ +0.5), κάνοντας ξεκάθαρο ότι οι υψηλές τιμές των ACK σημαιών μειώνουν δραστικά την πιθανότητα αναγνώρισης κακόβουλης κίνησης. Το Subflow Bwd Packets παρουσιάζει μωβ και μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία κοντά στο μηδέν, ενώ το Fwd Header Length.1 εμφανίζει κόκκινα σημεία αριστερά (≈ −3.0) και μπλε σημεία κοντά στο μηδέν. Το act_data_pkt_fwd έχει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (≈ +0.5), κάνοντας σαφές ότι οι υψηλές τιμές των ενεργών πακέτων δεδομένων μειώνουν την πιθανότητα ανίχνευσης.

Το Idle Mean εμφανίζει μπλε / μωβ σημεία κοντά στο μηδέν και κόκκινα σημεία δεξιά (≈ +3.5), γεγονός από το οποίο προκύπτει ότι οι υψηλές τιμές του χρόνου αδράνειας αυξάνουν την πιθανότητα αναγνώρισης επίθεσης. Το Bwd Packet Length Std παρουσιάζει μπλε σημεία αριστερά και γύρω από το μηδέν και κόκκινα σημεία δεξιά (≈ +4.5), υποδηλώνοντας ότι και οι ακραίες υψηλές τιμές της τυπικής απόκλισης των backward πακέτων αποτελούν ισχυρό δείκτη κακόβουλης κίνησης. Το Bwd IAT Std παρουσιάζει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα / μωβ σημεία δεξιά (≈ +1.3), ενώ το Packet Length Std εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά. Τα χαρακτηριστικά URG Flag Count, Total Backward Packets, Total Fwd Packets και Avg Fwd Segment Size, συμβάλλουν ελάχιστα στη διάκριση κανονικής από κακόβουλη κίνηση.



Εικόνα 175 Summary Plot, όλες οι κλάσεις για τον 18ο συνδυασμό.

Στον δέκατο ένατο συνδυασμό, στο συνολικό SHAP Summary Plot (Εικόνα 176) παρατηρούμε ότι το smeansz κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας κυρίως μπλε και κόκκινα σημεία αριστερά και μπλε σημεία κοντά στο μηδέν, υποδηλώνοντας με αυτή τη συμπεριφορά ότι οι υψηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν συνολικά την πιθανότητα ανίχνευσης. Το ct_srv_dst εμφανίζει διπλό μοτίβο με κόκκινα και μωβ σημεία αριστερά (≈ −1.2) και μπλε / μωβ σημεία στη συνέχεια (≈ −0.5), από τα οποία προκύπτει ότι επηρεάζει διαφορετικά ανά κλάση. Το sbytes παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.8) και μπλε / μωβ στη συνέχεια (≈ −0.7), ενώ παρόμοια συμπεριφορά παρουσιάζει και το ct_dst_sport_ltm. Το sttl εμφανίζει μπλε σημεία αριστερά (≈ −1.8) και κόκκινα σημεία κοντά στο μηδέν, ενώ το ct_state_ttl εμφανίζει μπλε σημεία δεξιά (≈ +0.7) και κόκκινα σημεία αριστερά, που σημαίνει ότι οι χαμηλές τιμές αυξάνουν την πιθανότητα ανίχνευσης. Τα χαρακτηριστικά is_sm_ips_ports, dwin, swin και state παρουσιάζουν ουσιαστικά μηδενική τιμή SHAP και άρα συμβάλλουν ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στις επιθέσεις Fuzzers (Εικόνα 177), το sttl κυριαρχεί παρουσιάζοντας μεταξύ των μπλε ορισμένα κόκκινα σημεία αριστερά (≈ −7.5), αλλά και κόκκινα σημεία κοντά στο μηδέν, γεγονός από το οποίο προκύπτει ότι οι υψηλές τιμές TTL πηγής αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το ct_srv_dst εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και μωβ / κόκκινα σημεία δεξιά (≈ +1.5), παρουσιάζοντας ισχυρό διπλό μοτίβο. Το smeansz εμφανίζει επίσης μπλε σημεία στα αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +3.5), υπονοώντας ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν ενώ υψηλές αυξάνουν την πιθανότητα αναγνώρισης. Το ct_srv_src έχει μπλε σημεία αριστερά και κόκκινα σημεία δεξιά, όπως επίσης και το sbytes ενώ τα χαρακτηριστικά is_sm_ips_ports και state παρουσιάζουν μηδενική επιρροή.

Για την κλάση 1 που αντιστοιχεί στις επιθέσεις Analysis (Εικόνα 178), το smeansz παρουσιάζει ισχυρό αρνητικό μοτίβο με διάσπαρτα κόκκινα και μπλε σημεία αριστερά (≈ −8.5) και συστάδα μπλε σημείων δεξιά (≈ +0.8), υποδηλώνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το ct_srv_dst εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και μωβ / κόκκινα σημεία δεξιά (≈ +1.0), ενώ το sbytes παρουσιάζει κόκκινα και μωβ σημεία αριστερά (≈ −5.5) και μπλε σημεία κοντά στο μηδέν. Το ct_dst_sport_ltm έχει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1.0), ενώ το sttl παρουσιάζει κόκκινα σημεία κοντά στο μηδέν

και μπλε σημεία δεξιά (≈ +2.0), δείχνοντας ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν ήπια την πιθανότητα ανίχνευσης. Τα χαρακτηριστικά is_sm_ips_ports, dwin και dloss δεν ασκούν κάποια επιρροή στην αναγνώριση.

Για την κλάση 2 που αντιστοιχεί στις επιθέσεις Backdoors (Εικόνα 179), το smeansz παρουσιάζεται με κόκκινα και μωβ σημεία αριστερά (≈ −9.0) και μπλε σημεία δεξιά (≈ +1.0), υποδεικνύοντας με αυτό το τρόπο ότι οι υψηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το sbytes εμφανίζει μπλε σημεία αριστερά (≈ −1.8) και κόκκινα σημεία δεξιά (≈ +4.0), γεγονός από το οποίο προκύπτει ότι τα υψηλά bytes πηγής αυξάνουν σημαντικά την πιθανότητα ανίχνευσης. Το Dload παρουσιάζει κόκκινα σημεία αριστερά (≈ −4.0) και μπλε σημεία δεξιά (≈ +1.0), ενώ το ct_dst_sport_ltm εμφανίζει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα κυρίως σημεία δεξιά (≈ +0.7).

Για την κλάση 3 που αντιστοιχεί στις επιθέσεις DoS (Εικόνα 180), το Dload επικρατεί παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −4.5) και μπλε σημεία δεξιά (≈ +0.5), γεγονός από το οποίο προκύπτει ότι οι υψηλές τιμές του ρυθμού λήψης δεδομένων μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης ως DoS. Το sbytes εμφανίζει μπλε σημεία αριστερά (≈ −1.0) και κόκκινα σημεία δεξιά (≈ +1.5), ενώ το ct_state_ttl παρουσιάζει αποκλειστικά μπλε σημεία αριστερά (≈ −2.1 έως -3.5), υποδηλώνοντας ότι οι χαμηλές τιμές μειώνουν την πιθανότητα ανίχνευσης. Το sttl εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +0.7), ενώ το dmeansz παρουσιάζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία κοντά στο μηδέν. Τα χαρακτηριστικά is_sm_ips_ports και dwin παρουσιάζουν μηδενική επίδραση.

Για την κλάση 4 που αντιστοιχεί στις επιθέσεις Exploits (Εικόνα 181), το dttl κυριαρχεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία κοντά στο μηδέν, υποδηλώνοντας ότι οι χαμηλές τιμές των TTL προορισμού μειώνουν την πιθανότητα αναγνώρισης της κλάσης αυτής. Το sbytes εμφανίζει (μεταξύ των μπλε) κόκκινα σημεία αριστερά και μπλε σημεία δεξιά, που σημαίνει ότι ορισμένες χαμηλές τιμές μπορούν επίσης να έχουν και αρνητική επίδραση ανάλογα, με την συνολική συμπεριφορά, ενώ το sttl παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.0) και μπλε σημεία δεξιά (≈ +2.5), υποδηλώνοντας ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν ήπια την πιθανότητα ανίχνευσης. Το smeansz παρουσιάζει μπλε σημεία αριστερά (≈ −1) και κόκκινα σημεία δεξιά (≈ +0.5), ενώ το Dload εμφανίζει κόκκινα σημεία αριστερά (≈ −1.0) και μπλε σημεία

δεξιά (≈ +0.5). Τα χαρακτηριστικά is_sm_ips_ports και state παρουσιάζουν πρακτικά μηδενική τιμή SHAP.

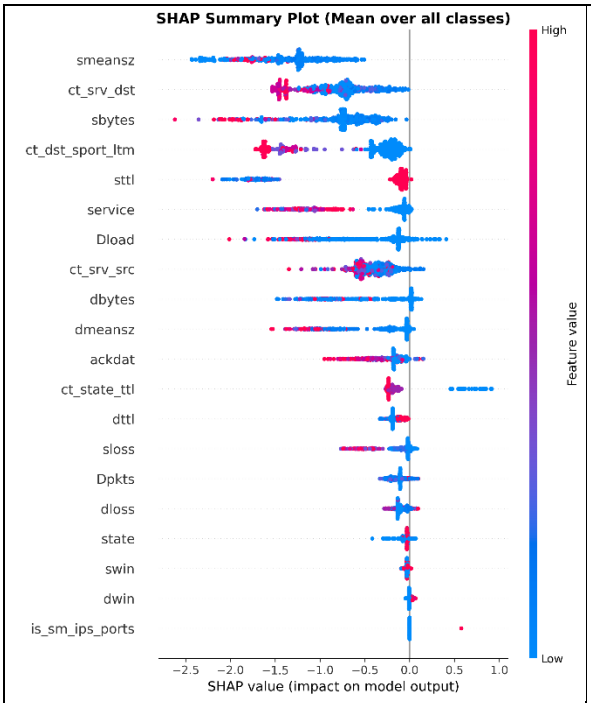
Για την κλάση 5 που αντιστοιχεί στις επιθέσεις Generic (Εικόνα 182), το sbytes παρουσιάζεται με μπλε σημεία αριστερά (≈ −3.5), υποδηλώνοντας ότι οι χαμηλές τιμές των bytes πηγής μειώνουν ενώ υψηλές αυξάνουν την πιθανότητα αναγνώρισης Generic επιθέσεων. Το smeansz εμφανίζει μπλε σημεία αριστερά (≈ −4.5) και κόκκινα σημεία δεξιά (≈ +1.0), ενώ το ct_srv_dst παρουσιάζει μπλε και μωβ σημεία αριστερά (≈ −3.5) και μπλε σημεία κοντά στο μηδέν. Το service έχει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (≈ +0.5), δείχνοντας ότι οι συγκεκριμένες υπηρεσίες σχετίζονται με αυξημένη πιθανότητα ανίχνευσης.

Για την κλάση 6 που αντιστοιχεί στις επιθέσεις Reconnaissance (Εικόνα 183), το smeansz κυριαρχεί παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −7) και μπλε σημεία δεξιά (≈ +2.5), κάνοντας ξεκάθαρο ότι οι υψηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το sbytes εμφανίζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (≈ +2.5), ενώ το ct_dst_sport_ltm παρουσιάζει κόκκινα και μωβ σημεία αριστερά (≈ −4.5) και μπλε σημεία κοντά στο μηδέν, κάνοντας σαφές ότι ο υψηλός αριθμός των συνδέσεων σε διαφορετικές πόρτες μειώνει την πιθανότητα ανίχνευσης. Το sttl εμφανίζει μπλε σημεία αριστερά (≈ −5.0) και κόκκινα σημεία κοντά στο μηδέν, ενώ το dbytes παρουσιάζει κόκκινα σημεία αριστερά (≈ −3.5) και μπλε σημεία κοντά στο μηδέν. Τα χαρακτηριστικά is_sm_ips_ports, ct_state_ttl και swin παρουσιάζουν SHAP δεν επηρεάζουν την αναγνώριση.

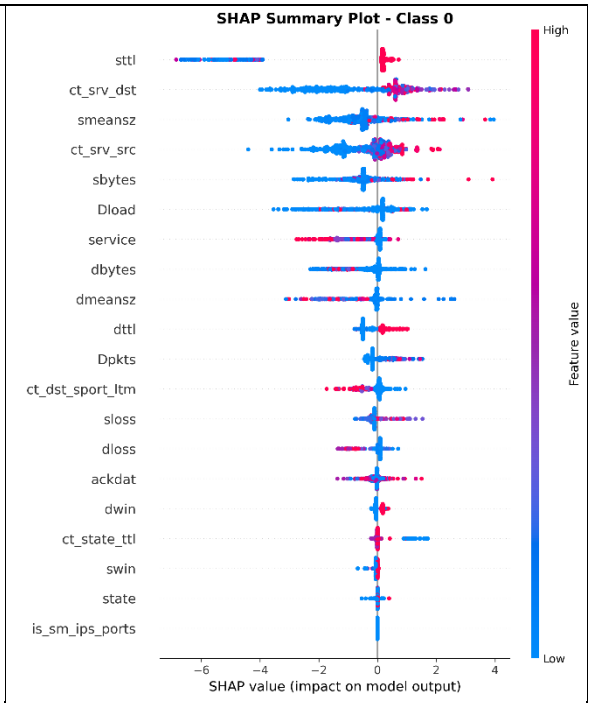
Για την κλάση 7 που αντιστοιχεί στις επιθέσεις Shellcode (Εικόνα 184), το ct_srv_dst παρουσιάζεται με κόκκινα και μωβ σημεία αριστερά (≈ −6.5) και μπλε σημεία δεξιά (≈ +0.5), δείχνοντας έτσι το διπλό μοτίβο επίδρασης. Το service εμφανίζει κόκκινα σημεία αριστερά (≈ −8.5) και μπλε σημεία δεξιά (≈ +1), υποδηλώνοντας ότι συγκεκριμένες υπηρεσίες μειώνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης. Το sbytes παρουσιάζει μπλε σημεία αριστερά (≈ −2.5) και μπλε σημεία δεξιά (≈ +2.0), ενώ το ct_dst_sport_ltm εμφανίζει κόκκινα και μωβ σημεία αριστερά (≈ −3.0) και μπλε σημεία κοντά στο μηδέν. Το sttl δείχνει κόκκινα σημεία αριστερά (≈ −4.0) και κόκκινα σημεία κοντά στο μηδέν, υποδηλώνοντας με αυτό το τρόπο ότι οι υψηλές τιμές των TTL πηγής μειώνουν την πιθανότητα ανίχνευσης, ενώ τα χαρακτηριστικά is_sm_ips_ports, swin και dwin παρουσιάζουν πρακτικά μηδενική τιμή SHAP.

Για την κλάση 8 που αντιστοιχεί στις επιθέσεις Worms (Εικόνα 185), το smeansz κυριαρχεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −10.0) και κόκκινα σημεία δεξιά (≈ +0.5), δείχνοντας έτσι ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης. Το sbytes εμφανίζει κόκκινα σημεία αριστερά (≈ −3.0) και μπλε σημεία δεξιά (≈ +5.0), υποδηλώνοντας ότι ο υψηλός αριθμός των bytes πηγής αυξάνουν σημαντικά την πιθανότητα. Το ct_srv_src παρουσιάζει κόκκινα / μωβ σημεία αριστερά (≈ −2.5) και μπλέ σημεία κοντά στο μηδέν, ενώ το Dload εμφανίζει μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία δεξιά (≈ +2.5), σύμφωνα με τα οποία ο υψηλός ρυθμός της λήψης αυξάνει ήπια την πιθανότητα ανίχνευσης της συγκεκριμένης κλάσης. Το dbytes δείχνει μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία δεξιά (≈ +7.5), που σημαίνει ότι ο εξαιρετικά υψηλός αριθμός των bytes προορισμού αποτελεί ισχυρό δείκτη ανίχνευσης.

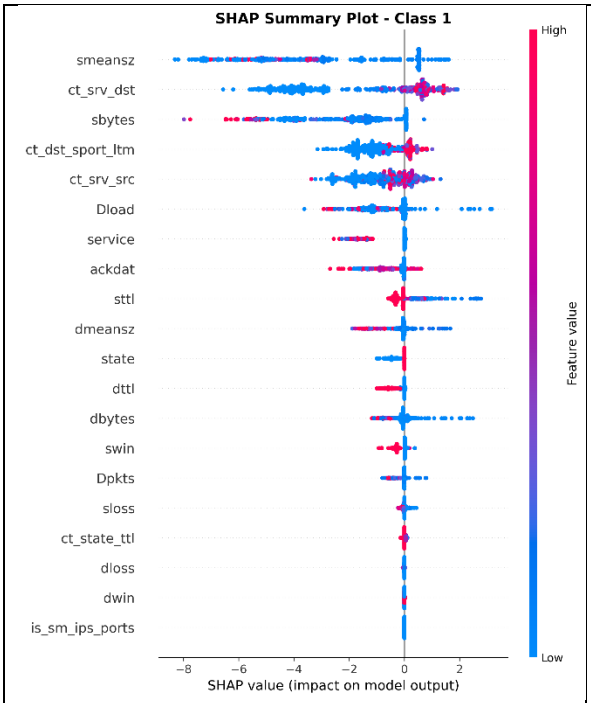
Για την κλάση 9 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 186), το ct_state_ttl παρουσιάζει διπλό μοτίβο με κόκκινα και μωβ σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (έως +11.5), υποδηλώνοντας ότι οι χαμηλές τιμές μειώνουν ενώ οι υψηλές αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης ως κανονική κίνηση. Το ct_srv_dst εμφανίζει μπλε σημεία αριστερά (≈ −2.0) και κόκκινα / μωβ σημεία δεξιά (≈ +1.5), ενώ το ct_dst_sport_ltm παρουσιάζει κόκκινα σημεία αριστερά (≈ −4.0) και μπλε σημεία κοντά στο μηδέν, από τα οποία προκύπτει ότι ο υψηλός αριθμός των συνδέσεων σε διαφορετικές πόρτες μειώνει την πιθανότητα κανονικής κίνησης. Το smeansz εμφανίζει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1.5), ενώ το Dpkts παρουσιάζει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (≈ +2.0), υποδεικνύοντας ότι ο υψηλός αριθμός των πακέτων προορισμού αυξάνει ήπια την πιθανότητα κανονικής κίνησης. Το is_sm_ips_ports παρουσιάζει ένα μεμονωμένο κόκκινο σημείο δεξιά (≈ +5.9), που σημαίνει ότι σε σπάνιες περιπτώσεις η χρήση μικρών ports επηρεάζει σημαντικά την αναγνώριση κανονικής κίνησης, ενώ τα χαρακτηριστικά state, dwin και swin ασκούν ουσιαστικά μηδενική επιρροή.



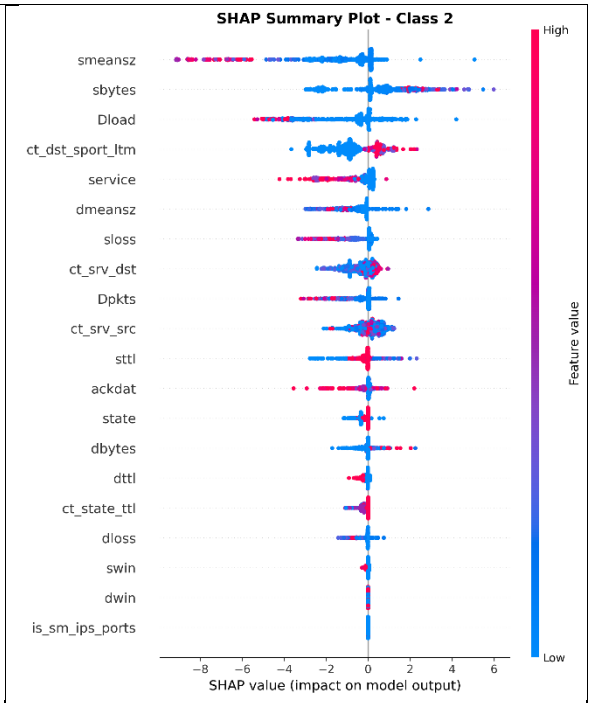
Εικόνα 176 Summary Plot, όλες οι κλάσεις για τον 19ο συνδυασμό.



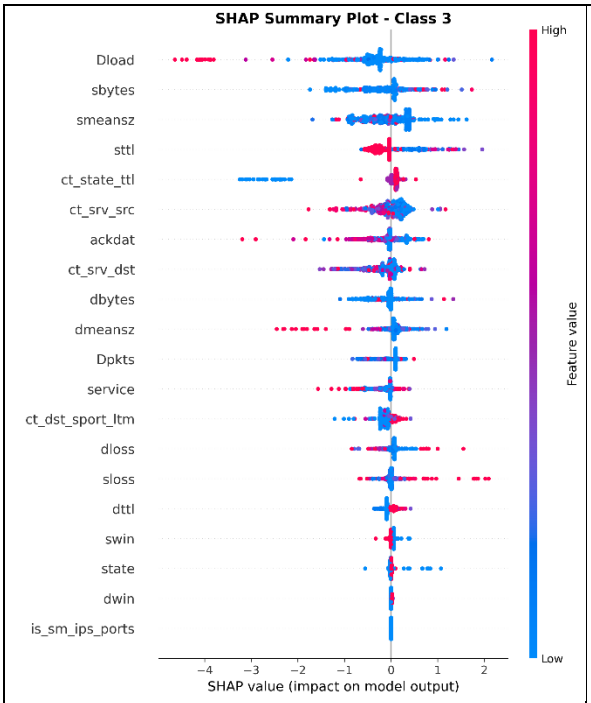
Εικόνα 177 SHAP Summary Plot, κλάση 0 για τον 19ο συνδυασμό.



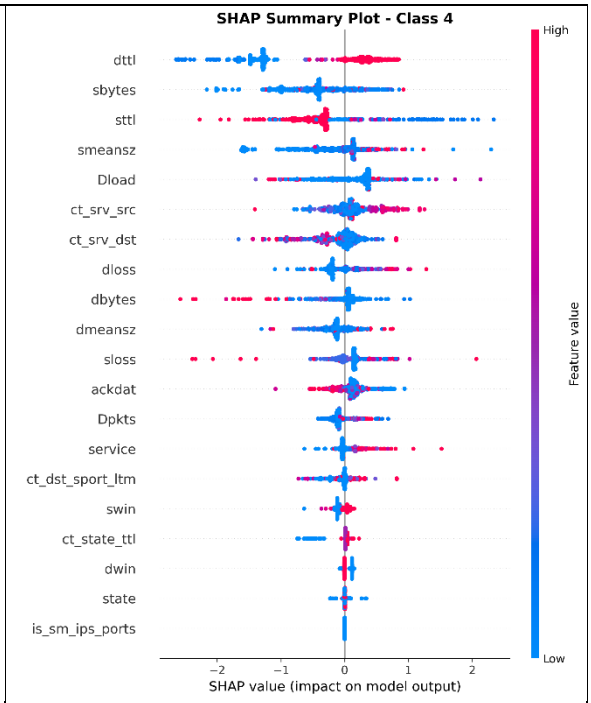
Εικόνα 178 SHAP Summary Plot, κλάση 1 για τον 19ο συνδυασμό.



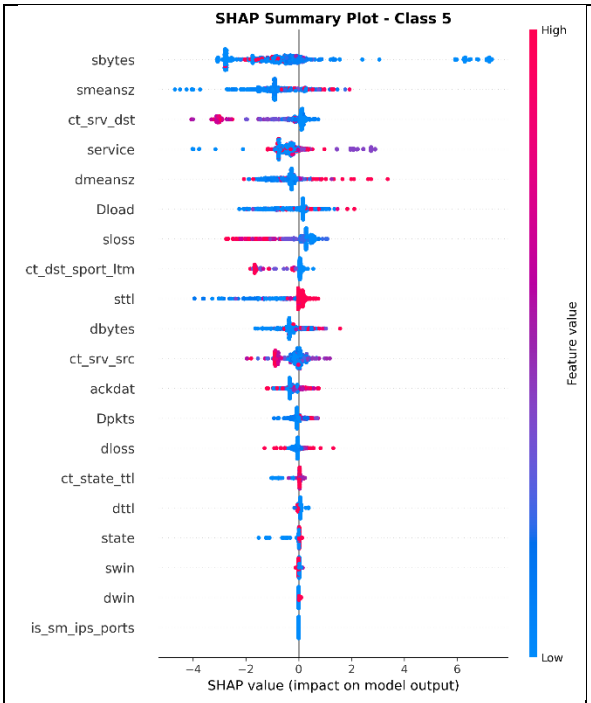
Εικόνα 179 SHAP Summary Plot, κλάση 2 για τον 19ο συνδυασμό.



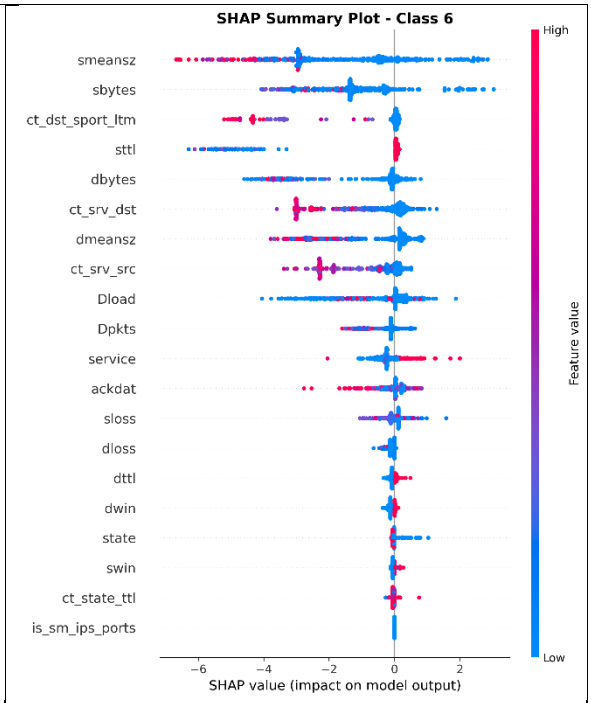
Εικόνα 180 SHAP Summary Plot, κλάση 3 για τον 19ο συνδυασμό.



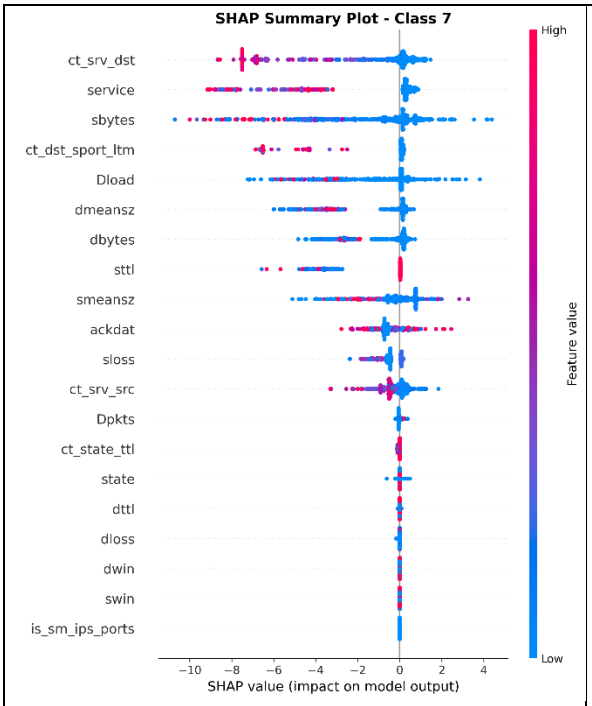
Εικόνα 181 SHAP Summary Plot, κλάση 4 για τον 19ο συνδυασμό.



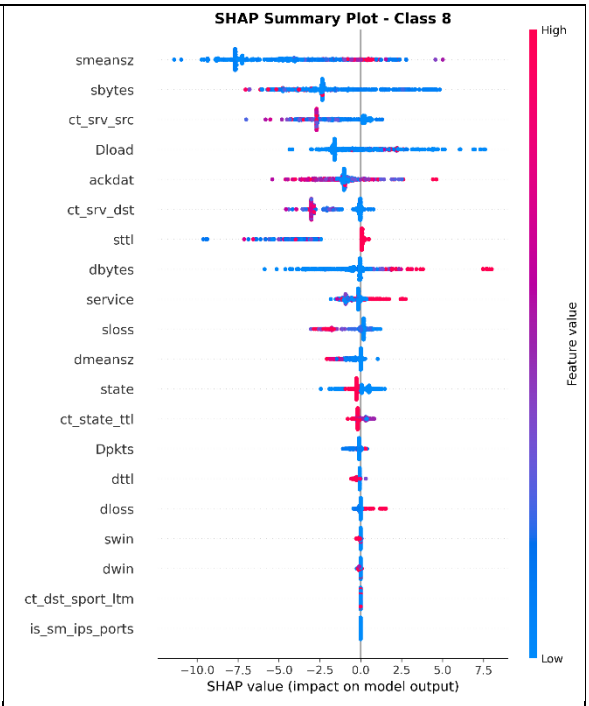
Εικόνα 182 SHAP Summary Plot, κλάση 5 για τον 19ο συνδυασμό.



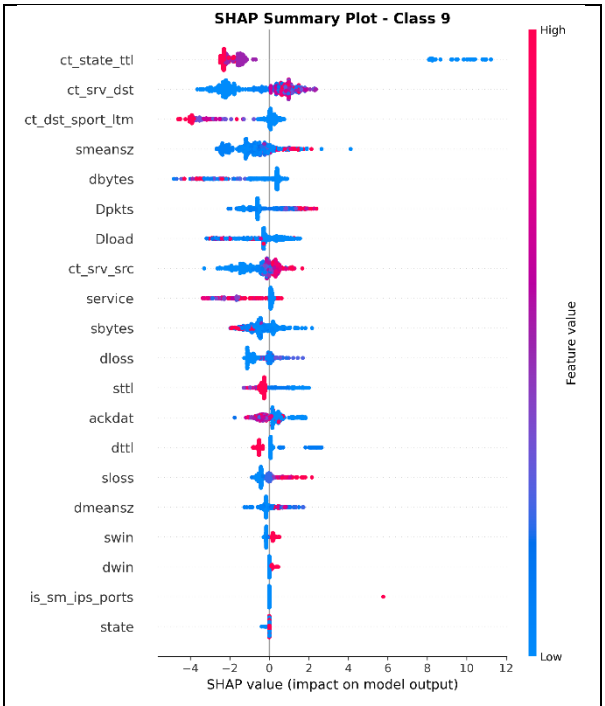
Εικόνα 183 SHAP Summary Plot, κλάση 6 για τον 19ο συνδυασμό.



Εικόνα 184 SHAP Summary Plot, κλάση 7 για τον 19ο συνδυασμό.



Εικόνα 185 SHAP Summary Plot, κλάση 8 για τον 19ο συνδυασμό.



Εικόνα 186 SHAP Summary Plot, κλάση 9 για τον 19ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (Εικόνα 187) του εικοστού συνδυασμού παρατηρούμε ότι το smeansz κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας κυρίως μπλε και κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία κοντά στο μηδέν, δείχνοντας με αυτό το τρόπο ότι το μέσο μέγεθος των πακέτων πηγής επηρεάζει αρνητικά την πιθανότητα ανίχνευσης συνολικά. Το sbytes εμφανίζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία γύρω από το μηδέν και δεξιά, όπως επίσης και το ct_srv_dst. Το sttl εμφανίζει μπλε σημεία αριστερά (≈ −1.7) και κόκκινα σημεία κοντά στο μηδέν (≈ -0.2), που σημαίνει ότι οι χαμηλές τιμές των TTL πηγής μειώνουν συνολικά την πιθανότητα ανίχνευσης. Το ct_state_ttl παρουσιάζει κόκκινα και μωβ σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (έως +1.2), υποδηλώνοντας ότι οι χαμηλές τιμές αυξάνουν ήπια την πιθανότητα ανίχνευσης σε ορισμένες κλάσεις. Τα χαρακτηριστικά is_sm_ips_ports, dwin και swin, συμβάλλουν ελάχιστα στη συνολική απόφαση.

Για την κλάση 0 που αντιστοιχεί στις επιθέσεις Fuzzers (όπως φαίνεται στην Εικόνα 188), το ct_srv_dst επικρατεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία δεξιά (≈ +4.0), δείχνοντας διπλό μοτίβο όπου οι χαμηλές τιμές μειώνουν ενώ υψηλές αυξάνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης ως Fuzzers. Το sttl εμφανίζει μπλε σημεία αριστερά (≈ −7.0) και κόκκινα σημεία κοντά στο μηδέν, υποδηλώνοντας ότι χαμηλές τιμές των TTL πηγής μειώνουν την πιθανότητα ανίχνευσης. Το ct_srv_src έχει μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία δεξιά (≈ +1.5), όμοια με το sbytes το οποίο επίσης εμφανίζει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1). Το Dload παρουσιάζει μπλε σημεία με διάσπαρτα κόκκινα ενώ τα is_sm_ips_ports και state παρουσιάζουν πρακτικά μηδενική τιμή SHAP.

Για την κλάση 1 που αντιστοιχεί στις επιθέσεις Analysis (Εικόνα 189), το smeansz παρουσιάζεται με μπλε σημεία αριστερά (≈ −4.0) και κόκκινα σημεία δεξιά γεγονός που υποδεικνύει ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το ct_srv_dst εμφανίζει μπλε σημεία αριστερά (≈ −3.5) και κόκκινα σημεία δεξιά (≈ +2.0), ενώ το sbytes παρουσιάζει μπλε σημεία αριστερά και κοντά στο μηδέν. Το ct_srv_src φαίνεται με μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1.5), ενώ το service παρουσιάζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία κοντά στο μηδέν, από τα οποία προκύπτει ότι οι συγκεκριμένες υπηρεσίες μειώνουν την πιθανότητα ανίχνευσης.

Για την κλάση 2 που αντιστοιχεί στις επιθέσεις Backdoors (Εικόνα 190), το Dload κυριαρχεί παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −5.5) και μπλε σημεία δεξιά (≈ +1.5), που σημαίνει ότι οι υψηλές τιμές του ρυθμού λήψης δεδομένων μειώνουν την πιθανότητα αναγνώρισης της κλάσης ως Backdoors. Το smeansz παρουσιάζει διάσπαρτα κόκκινα σημεία αριστερά και μπλε σημεία δεξιά, υποδηλώνοντας ένα ισχυρό διπλό μοτίβο. Το sbytes φαίνεται με μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (έως +5.5), από τα οποία προκύπτει ότι υψηλά bytes πηγής αυξάνουν σημαντικά την πιθανότητα ανίχνευσης. Το dmeansz εμφανίζεται με κόκκινα σημεία αριστερά (≈ −2.0) και μπλέ σημεία δεξιά (≈ +1.5), ενώ το sloss έχει κόκκινα και μωβ σημεία αριστερά και μπλε σημεία κοντά στο μηδέν. Τα χαρακτηριστικά is_sm_ips_ports, dwin και swin δεν επηρεάζουν την αναγνώριση.

Για την κλάση 3 που αντιστοιχεί στις επιθέσεις DoS (Εικόνα 191), το Dload παρουσιάζεται με κόκκινα σημεία αριστερά (≈ −5.5) και μπλε σημεία δεξιά (≈ +0.5), υποδηλώνοντας ότι οι υψηλές τιμές του ρυθμού λήψης μειώνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης ως DoS, αντικατοπτρίζοντας ότι η συγκεκριμένη επίθεση χαρακτηρίζεται από ελάχιστη ανταπόκριση από τον προορισμό. Το ct_srv_src εμφανίζει κόκκινα σημεία αριστερά (≈ −2.5) και μπλε σημεία κοντά στο μηδέν, ενώ το sttl παρουσιάζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +2.5), που σημαίνει ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν την πιθανότητα της ανίχνευσης. Το ct_state_ttl εμφανίζει μπλε σημεία αριστερά (≈ −1.5), από τα οποία φαίνεται πως οι χαμηλές τιμές μειώνουν την πιθανότητα αναγνώρισης DoS και το sloss παρουσιάζει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (≈ +2.5).

Για την κλάση 4 που αντιστοιχεί στις επιθέσεις Exploits (Εικόνα 192), το dttl κυριαρχεί παρουσιάζοντας διπλό μοτίβο με μπλε σημεία αριστερά (≈ −3.5) και κόκκινα σημεία κοντά στο μηδέν, κάνοντας σαφές ότι οι χαμηλές τιμές των TTL προορισμού μειώνουν την πιθανότητα αναγνώρισης της κλάσης αυτής. Το sttl εμφανίζει κόκκινα σημεία αριστερά (≈ −1.0) και μπλε σημεία δεξιά (≈ +2.0), υποδηλώνοντας ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν ήπια την πιθανότητα ανίχνευσης. Το sbytes παρουσιάζει κυρίως μπλε σημεία αριστερά, το smeansz μπλε σημεία αριστερά (≈ −1.0) και κόκκινα σημεία δεξιά (≈ +0.5), ενώ το Dload παρουσιάζει διάσπαρτα κόκκινα σημεία κατά μήκος των μπλε. Τα χαρακτηριστικά is_sm_ips_ports και state παρουσιάζουν πρακτικά μηδενική τιμή SHAP και άρα δεν ακούν κάποια επιρροή.

Για την κλάση 5 που αντιστοιχεί στις επιθέσεις Generic (Εικόνα 193), το sbytes παρουσιάζει τα περισσότερα μπλε σημεία αριστερά (≈ −1.0), από τα οποία προκύπτει ότι οι χαμηλές τιμές των bytes πηγής μειώνουν την πιθανότητα αναγνώρισης Generic επιθέσεων. Το smeansz και το dmeansz εμφανίζουν παρόμοια συμπεριφορά με μπλε σημεία αριστερά (≈ −1.7) και κόκκινα σημεία δεξιά (≈ +1.5). Το Dload φαίνεται με μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +0.5), ενώ το service παρουσιάζει μπλε σημεία αριστερά (≈ −1.0) και μωβ σημεία δεξιά (≈ +1.5), υποδεικνύοντας έτσι ότι συγκεκριμένες υπηρεσίες σχετίζονται με αυξημένη πιθανότητα ανίχνευσης. Τα χαρακτηριστικά is_sm_ips_ports, dwin και swin παρουσιάζουν σχεδόν μηδενική τιμή SHAP.

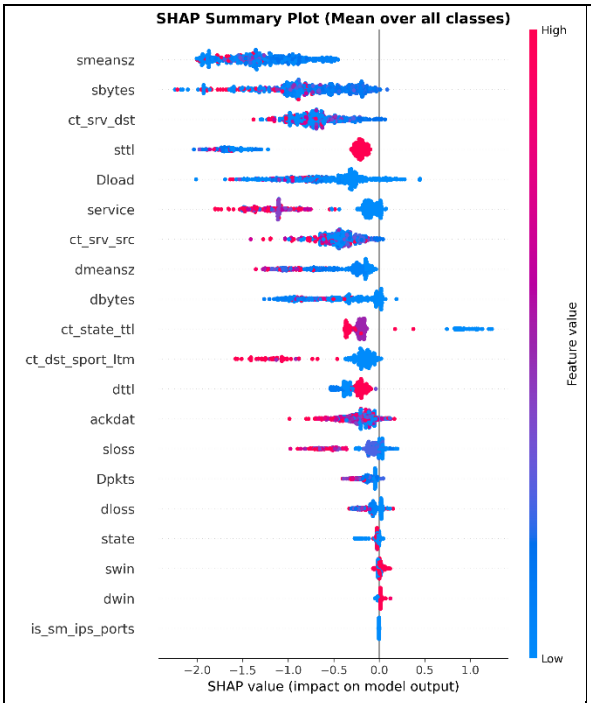
Για την κλάση 6 που αντιστοιχεί στις επιθέσεις Reconnaissance (Εικόνα 194), το smeansz κυριαρχεί παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −4.5) και μπλε σημεία δεξιά (≈ +3.0), δείχνοντας ότι οι υψηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το sbytes εμφανίζει και αυτό κόκκινα σημεία αριστερά (≈ −4.0) και μπλε σημεία δεξιά (≈ +4.0), ενώ το sttl παρουσιάζει μπλε σημεία αριστερά (≈ −6) και κόκκινα κοντά στο μηδέν, από τα οποία προκύπτει ότι οι χαμηλές τιμές των TTL πηγής μειώνουν σημαντικά την πιθανότητα της ανίχνευσης. Το dbytes εμφανίζει κόκκινα σημεία αριστερά (≈ −3.5) και μπλε σημεία κοντά στο μηδέν, το Dload παρουσιάζει μπλε σημεία αριστερά και τα χαρακτηριστικά is_sm_ips_ports, ct_state_ttl και swin δεν ασκούν ουσιαστική επιρροή στην ανίχνευση της κλάσης.

Για την κλάση 7 που αντιστοιχεί στις επιθέσεις Shellcode (αποτυπώνεται στην Εικόνα 195), το service παρουσιάζεται με κόκκινα και μωβ σημεία αριστερά (≈ από −3.5 έως −9.5) και μπλε σημεία δεξιά (≈ +1), υποδηλώνοντας ότι οι συγκεκριμένες υπηρεσίες μειώνουν δραστικά την πιθανότητα αναγνώρισης της κλάσης αυτής. Το sbytes εμφανίζει κόκκινα σημεία αριστερά (≈ −7.5) και μπλε σημεία δεξιά (≈ +1.5), ενώ το dmeansz φαίνεται με κόκκινα σημεία αριστερά (≈ −3.5) και μπλε σημεία κοντά στο μηδέν. Το ct_srv_dst εμφανίζει κόκκινα σημεία αριστερά και μπλε σημεία δεξιά, ενώ το sttl παρουσιάζει κόκκινα σημεία αριστερά (μεταξύ των μπλε) και κόκκινα σημεία κοντά στο μηδέν, υποδεικνύοντας με αυτό το μοτίβο ότι οι υψηλές τιμές των TTL πηγής μειώνουν την πιθανότητα ανίχνευσης.

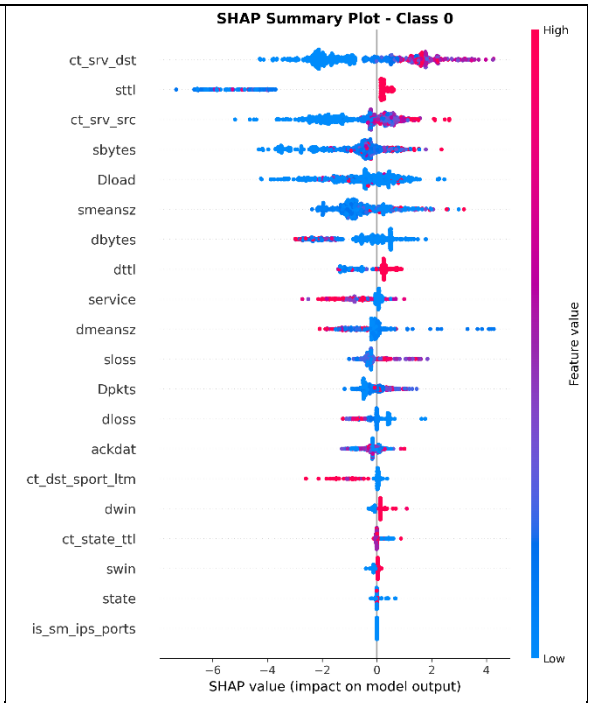
Για την κλάση 8 που αντιστοιχεί στις επιθέσεις Worms (Εικόνα 196), το smeansz κυριαρχεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −10) και κόκκινα σημεία δεξιά (≈ +0.5), που σημαίνει ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το sbytes εμφανίζει μπλε σημεία αριστερά (≈

−3.0), υποδηλώνοντας ότι ο χαμηλός αριθμός των bytes πηγής μειώνει την πιθανότητα ανίχνευσης. Το Dload φαίνεται και αυτό με μπλε σημεία αριστερά (≈ −3.0) αλλά και με κόκκινα σημεία δεξιά (≈ +2.0), από τα οποία πηγάζει ότι ο υψηλός ρυθμός λήψης αποτελεί χαρακτηριστικό δείκτη Worms. Το ackdat εμφανίζει κόκκινα σημεία αριστερά (≈ −3.0) και κόκκινα σημεία δεξιά (≈ +1.5), ενώ το dbytes παρουσιάζει μπλε σημεία αριστερά και κόκκινα σημεία δεξιά, κάνοντας σαφές ότι εξαιρετικά υψηλά bytes του προορισμού αποτελούν επίσης δείκτη Worms.

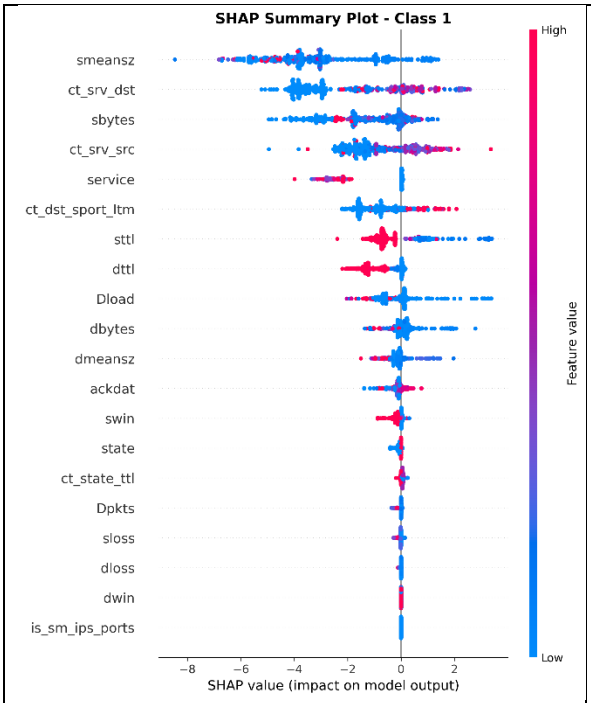
Για την κλάση 9 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 197), το ct_state_ttl παρουσιάζει ένα ισχυρό διπλό μοτίβο με κόκκινα και μωβ σημεία αριστερά (≈ από −2.0 έως −3.0) και μπλε σημεία δεξιά (≈ από +10.0 έως +13.5), υποδηλώνοντας ότι οι χαμηλές τιμές αυξάνουν ενώ οι υψηλές μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το ct_srv_dst εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία δεξιά (≈ +2.5), ενώ το smeansz παρουσιάζει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία κοντά στο μηδέν. Αντίθετα, το sttl εμφανίζει κόκκινα σημεία κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +4.5), δείχνοντας ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν ήπια την πιθανότητα κανονικής κίνησης. Το ct_dst_sport_ltm παρουσιάζει κόκκινα σημεία αριστερά (≈ −3.5) και μπλε σημεία κοντά στο μηδέν, τα οποία σημαίνουν ότι ο υψηλός αριθμός συνδέσεων σε διαφορετικές πόρτες μειώνει την πιθανότητα κανονικής κίνησης. Τα χαρακτηριστικά is_sm_ips_ports, state και dwin ασκούν σχεδόν μηδενική επιρροή στην αναγνώριση της κλάσης.



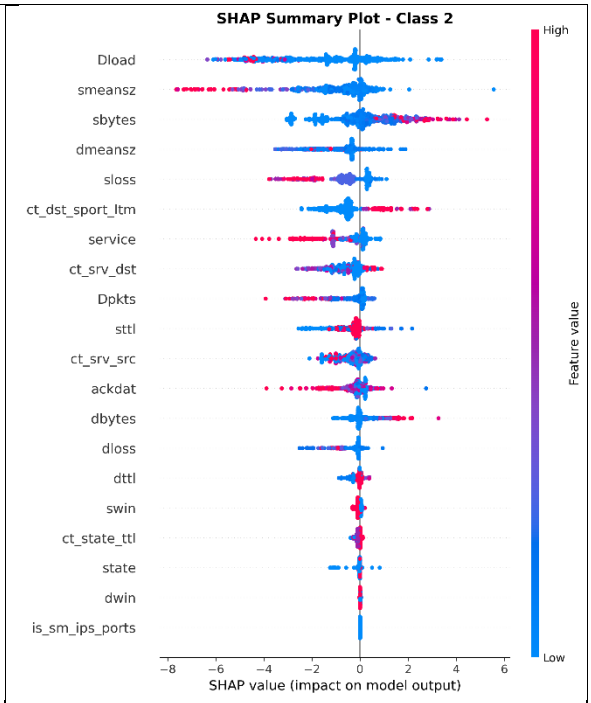
Εικόνα 187 Summary Plot, όλες οι κλάσεις για τον 20ο συνδυασμό.



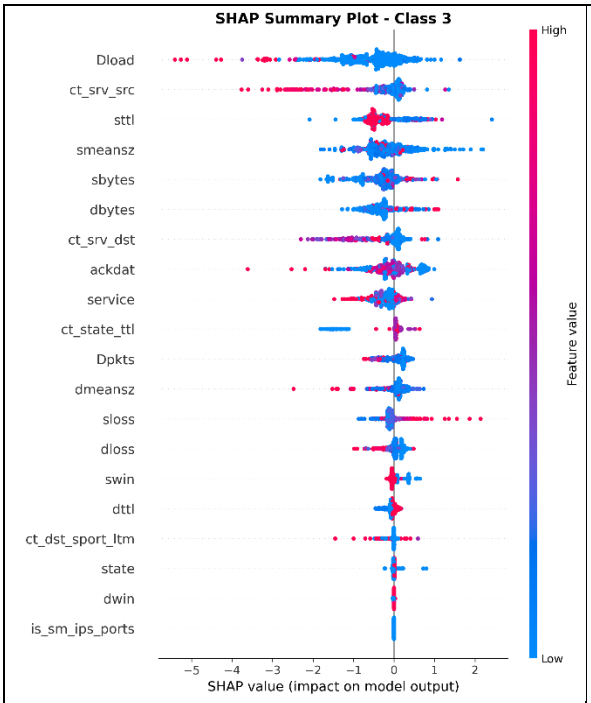
Εικόνα 188 SHAP Summary Plot, κλάση 0 για τον 20ο συνδυασμό.



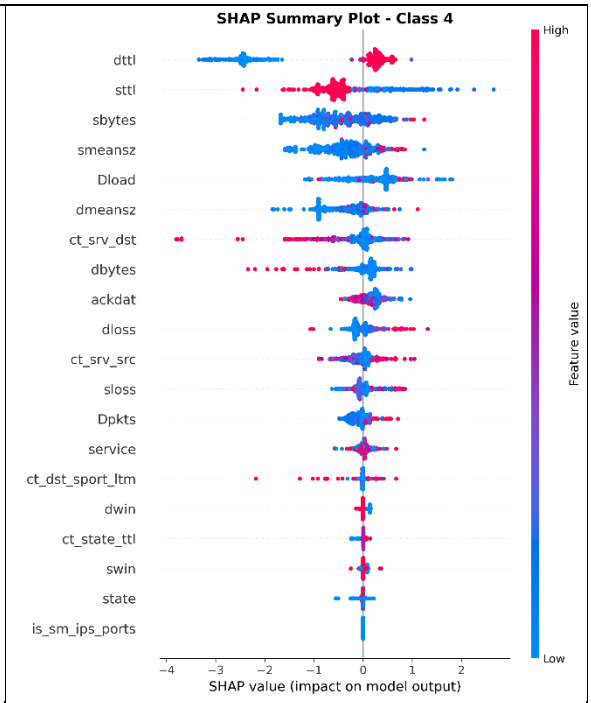
Εικόνα 189 SHAP Summary Plot, κλάση 1 για τον 20ο συνδυασμό.



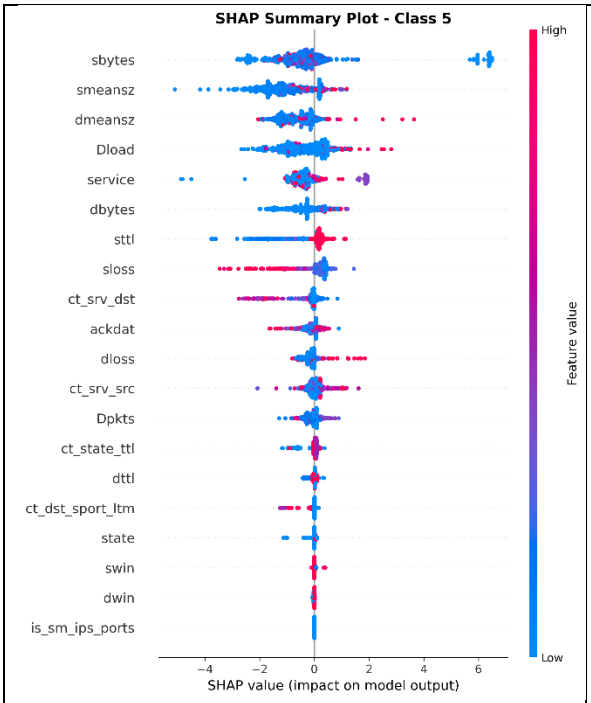
Εικόνα 190 SHAP Summary Plot, κλάση 2 για τον 20ο συνδυασμό.



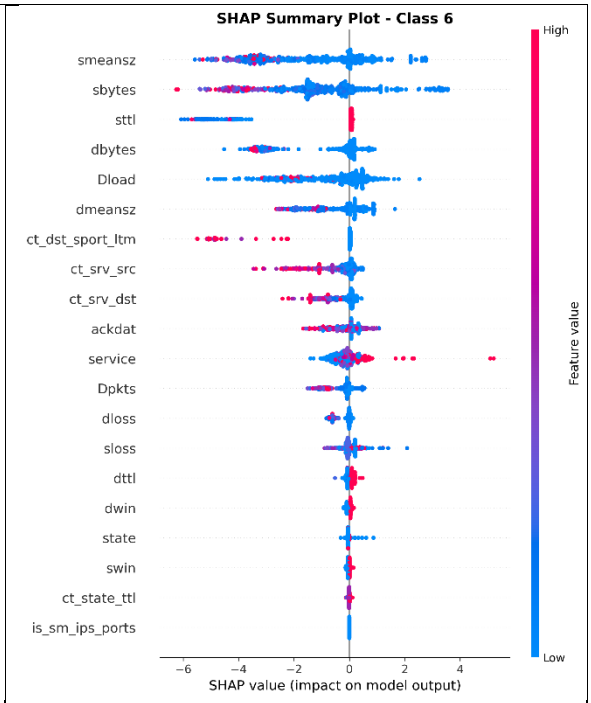
Εικόνα 191 SHAP Summary Plot, κλάση 3 για τον 20ο συνδυασμό.



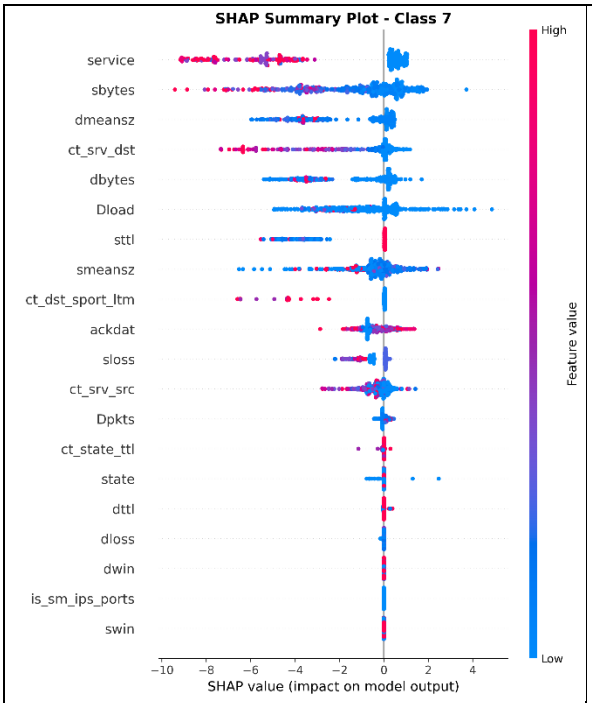
Εικόνα 192 SHAP Summary Plot, κλάση 4 για τον 20ο συνδυασμό.



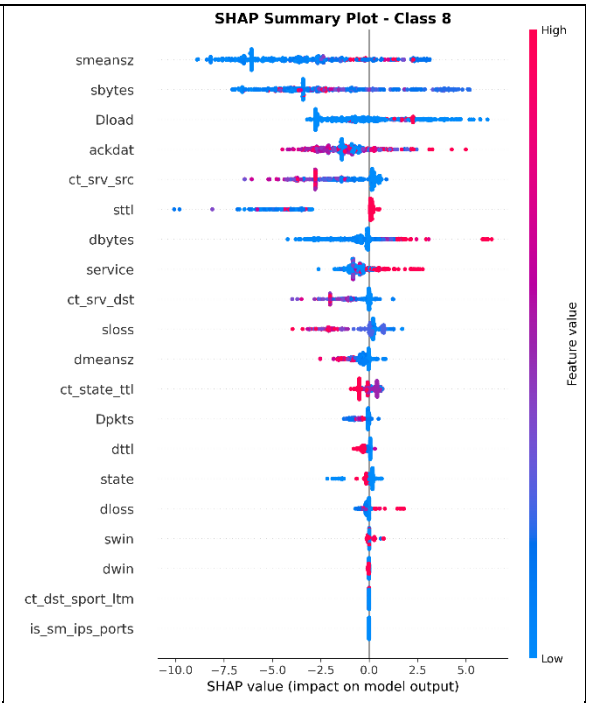
Εικόνα 193 SHAP Summary Plot, κλάση 5 για τον 20ο συνδυασμό.



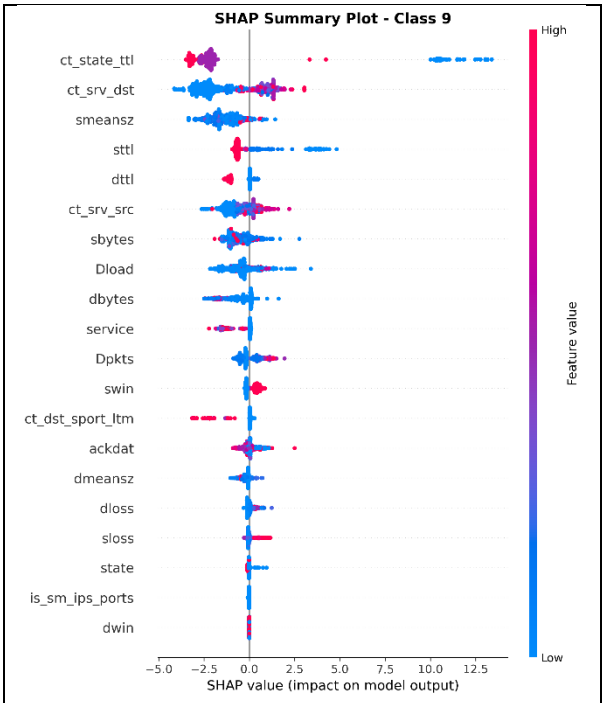
Εικόνα 194 SHAP Summary Plot, κλάση 6 για τον 20ο συνδυασμό.



Εικόνα 195 SHAP Summary Plot, κλάση 7 για τον 20ο συνδυασμό.



Εικόνα 196 SHAP Summary Plot, κλάση 8 για τον 20ο συνδυασμό.



Εικόνα 197 SHAP Summary Plot, κλάση 9 για τον 20ο συνδυασμό.

Στο συνολικό SHAP Summary Plot (όπως δείχνει η Εικόνα 198) του εικοστού πρώτου συνδυασμού παρατηρούμε ότι το smeansz κυριαρχεί ως το πιο σημαντικό χαρακτηριστικό, παρουσιάζοντας κυρίως μπλε και κόκκινα σημεία αριστερά (≈ −1.3) και μπλε σημεία κοντά στο μηδέν, υποδηλώνοντας με αυτό τρόπο ότι το μέσο μέγεθος των πακέτων πηγής επηρεάζει αρνητικά την πιθανότητα ανίχνευσης συνολικά. Το ct_srv_dst εμφανίζει παρόμοια συμπεριφορά, ενώ το sttl παρουσιάζει ένα σύνολο μπλε σημείων αριστερά (≈ −1.5) και κόκκινων σημείων κοντά στο μηδέν (≈ −0.2), που σημαίνει ότι οι χαμηλές τιμές των TTL πηγής μειώνουν συνολικά την πιθανότητα της ανίχνευσης, αλλά παρόμοια επίδρασης ενδέχεται να έχουν και οι υψηλές. Το sbytes εμφανίζει κυρίως μπλε σημεία αριστερά ενώ το ct_state_ttl παρουσιάζει κυρίως κόκκινα, δείχνοντας ότι οι υψηλές τιμές αυξάνουν ήπια την πιθανότητα ανίχνευσης σε ορισμένες κλάσεις. Τα χαρακτηριστικά is_sm_ips_ports, dwin και swin δεν παρουσιάζουν ουσιαστική επιρροή.

Για την κλάση 0 που αντιστοιχεί στις επιθέσεις Fuzzers (Εικόνα 199), το sttl παρουσιάζει μπλε σημεία αριστερά (≈ −6.0) και κόκκινα σημεία κοντά στο μηδέν, από τα οποία προκύπτει ότι οι υψηλές τιμές των TTL πηγής αυξάνουν την πιθανότητα αναγνώρισης της κλάσης. Το ct_srv_dst εμφανίζεται με μπλε σημεία αριστερά και κόκκινα δεξιά, υποδηλώνοντας ισχυρό διπλό μοτίβο κατά το οποίο οι χαμηλές τιμές μειώνουν ενώ οι υψηλές αυξάνουν σημαντικά την πιθανότητα αναγνώρισης. Το sbytes, το smeansz και το ct_srv_src φαίνονται με μπλε σημεία αριστερά και κόκκινα σημεία δεξιά, που σημαίνει ότι οι υψηλές τιμές τους επηρεάζουν θετικά την πιθανότητα αναγνώρισης. Το Dload εμφανίζει επίσης μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +0.5), ενώ τα χαρακτηριστικά is_sm_ips_ports και state παρουσιάζουν πρακτικά μηδενική τιμή SHAP.

Για την κλάση 1 που αντιστοιχεί στις επιθέσεις Analysis (Εικόνα 200), το smeansz επικρατεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −7), τα οποία δείχνουν ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης ως Analysis. Το ct_srv_dst εμφανίζεται με μπλε σημεία αριστερά (≈ −3.0) και κόκκινα / μωβ σημεία δεξιά (≈ +2.5), ενώ το ct_srv_src παρουσιάζει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1.5). Το sbytes φαίνεται με μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία κοντά στο μηδέν, ενώ το service παρουσιάζει κόκκινα / μωβ σημεία αριστερά (≈ −2.0) και μπλε σημεία κοντά στο μηδέν, υποδηλώνοντας ότι συγκεκριμένες υπηρεσίες που αναγνωρίζονται μειώνουν την πιθανότητα ανίχνευσης.

Για την κλάση 2 που αντιστοιχεί στις επιθέσεις Backdoors (Εικόνα 201), το Dload κυριαρχεί παρουσιάζοντας κόκκινα και μπλε σημεία αριστερά (≈ −6.0) και μπλε σημεία δεξιά (≈ +2.0), από τα οποία προκύπτει ότι οι υψηλές τιμές του ρυθμού λήψης δεδομένων μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το smeansz εμφανίζεται με κόκκινα σημεία αριστερά (≈ −7) και μπλε σημεία δεξιά (≈ +5.5), υποδεικνύοντας έτσι ένα ισχυρό διπλό μοτίβο. Το sbytes φαίνεται με μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία δεξιά (≈ +2.5), που σημαίνει ότι ο υψηλός αριθμός bytes της πηγής αυξάνει σημαντικά την πιθανότητα ανίχνευσης. Το dmeansz παρουσιάζει μπλε σημεία δεξιά (≈ +2.0), ενώ το sloss κόκκινα και μωβ σημεία αριστερά (≈ −1.5) και μπλε σημεία κοντά στο μηδέν. Τα χαρακτηριστικά is_sm_ips_ports, dwin και swin παρουσιάζουν πρακτικά μηδενική SHAP τιμή.

Για την κλάση 3 που αντιστοιχεί στις επιθέσεις DoS (Εικόνα 202), επικρατεί το Dload παρουσιάζοντας κόκκινα σημεία αριστερά (≈ −3.0) και μπλε σημεία δεξιά (≈ +1.5), υποδηλώνοντας με αυτό το μοτίβο ότι οι υψηλές τιμές του ρυθμού λήψης μειώνουν σημαντικά την πιθανότητα αναγνώρισης της κλάσης. Το ct_srv_src εμφανίζει κόκκινα σημεία αριστερά (≈ −2.0) και μπλε σημεία κοντά στο μηδέν, ενώ το sttl παρουσιάζει κόκκινα σημεία κοντά στο μηδέν και κυρίως μπλε σημεία δεξιά (≈ +0.5), από τα οποία προκύπτει ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν την πιθανότητα της ανίχνευσης. Το ct_state_ttl εμφανίζει μπλε σημεία αριστερά (≈ −1.0) και κόκκινα κοντα στο μηδέν, επομένως οι χαμηλές τιμές μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το dttl παρουσιάζει μπλε σημεία αριστερά (≈ −0.5) και κόκκινα σημεία δεξιά (≈ +0.5) ενώ τα χαρακτηριστικά is_sm_ips_ports και dwin δεν ασκούν ουσιαστική επιρροή.

Για την κλάση 4 που αντιστοιχεί στις επιθέσεις Exploits (φαίνεται στην Εικόνα 203), το dttl κυριαρχεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −3.5) και κόκκινα σημεία κοντά στο μηδέν, κάνοντας σαφές ότι οι χαμηλές τιμές των TTL προορισμού μειώνουν την πιθανότητα αναγνώρισης της κλάσης ως Exploits. Το sttl εμφανίζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία δεξιά (≈ +2.5), ενώ το sbytes παρουσιάζει κυρίως μπλε σημεία που σημαίνει ότι ορισμένες χαμηλές τιμές μπορούν επίσης να έχουν και αρνητική επίδραση ανάλογα, με την συνολική συμπεριφορά. Το smeansz εμφανίζει μπλε σημεία αριστερά και κόκκινα δεξιά και το Dload παρουσιάζει όμοια συμπεριφορά με μικρές διαφορές στα σημεία.

Για την κλάση 5 που αντιστοιχεί στις επιθέσεις Generic (Εικόνα 204), το sbytes παρουσιάζει μπλε σημεία αριστερά και δεξιά με διάσπαρτα κόκκινα, υποδηλώνοντας ότι οι χαμηλές τιμές των bytes πηγής μειώνουν την πιθανότητα αναγνώρισης Generic επιθέσεων, αλλά μπορούν να λειτουργήσουν και αυξητικά ανάλογα με τη συνολική συμπεριφορά. Το smeansz εμφανίζει μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1.5), ενώ το service παρουσιάζει μπλε σημεία αριστερά (≈ −0.5) και μωβ σημεία δεξιά (≈ +2.5), από τα οποία προκύπτει ότι συγκεκριμένες υπηρεσίες σχετίζονται με αυξημένη πιθανότητα ανίχνευσης. Το Dload φαίνεται με μπλε σημεία αριστερά (≈ −1.5) και κόκκινα σημεία δεξιά (≈ +1.0), το dmeansz παρουσιάζει επίσης μπλε σημεία αριστερά (≈ −1) και κόκκινα σημεία δεξιά (≈ +0.5), ενώ τα is_sm_ips_ports, dwin και swin παρουσιάζουν πρακτικά μηδενική τιμή SHAP.

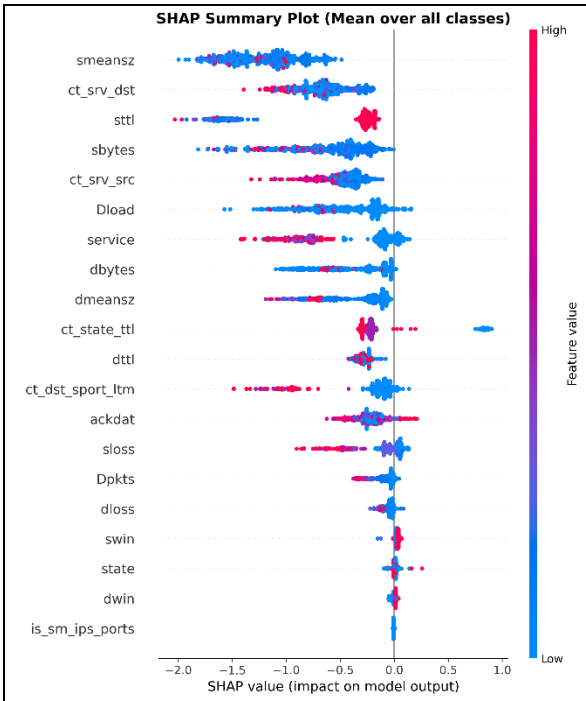
Για την κλάση 6 που αντιστοιχεί στις επιθέσεις Reconnaissance (βλέπε Εικόνα 205), το smeansz παρουσιάζει κόκκινα σημεία αριστερά (≈ −5.5) και μπλε σημεία δεξιά (≈ +3.0), που σημαίνει ότι οι υψηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν την πιθανότητα αναγνώρισης της κλάσης. Το sbytes εμφανίζεται με παρόμοια συμπεριφορά, με κόκκινα σημεία αριστερά (≈ −4.5) και μπλε σημεία δεξιά (≈ +4), ενώ το sttl παρουσιάζει μπλε σημεία αριστερά (≈ −6.0) και κόκκινα σημεία κοντά στο μηδέν, από τα οποία προκύπτει ότι οι χαμηλές τιμές των TTL πηγής μειώνουν σημαντικά την πιθανότητα ανίχνευσης. Το dbytes φαίνεται με ορισμένα κόκκινα σημεία αριστερά (≈ −2.8) και μπλε σημεία κοντά στο μηδέν, ενώ το Dload παρουσιάζει κυρίως μπλε σημεία αριστερά που επεκτείνονται γύρω από το μηδέν. Τα χαρακτηριστικά is_sm_ips_ports, ct_state_ttl και swin δεν ασκούν σημαντική επιρροή στην αναγνώριση.

Για την κλάση 7 που αντιστοιχεί στις επιθέσεις Shellcode (Εικόνα 206), το service κυριαρχεί παρουσιάζοντας κόκκινα και μωβ σημεία αριστερά (≈ −9.5) και μπλε σημεία δεξιά (≈ +1), υποδηλώνοντας ότι συγκεκριμένες υπηρεσίες μειώνουν δραστικά την πιθανότητα αναγνώρισης Shellcode επίθεσης. Το sbytes εμφανίζει κόκκινα σημεία αριστερά (≈ −6.5) και μπλε σημεία δεξιά (≈ +1.5), το ct_srv_dst με παρόμοια συμπεριφορά, κόκκινα σημεία αριστερά (≈ −5.5) και μπλε σημεία δεξιά (≈ +0.5), ενώ το Dload εμφανίζει μπλε σημεία αριστερά και δεξιά με διάσπαρτα κόκκινα, μοτίβο που δείχνει ότι ο χαμηλός ρυθμός μετάδοσης του προορισμού σε bits ανά δευτερόλεπτο αυξάνει την πιθανότητα αναγνώρισης. Το dmeansz παρουσιάζει μπλε σημεία αριστερά και κοντά στο μηδέν και τα

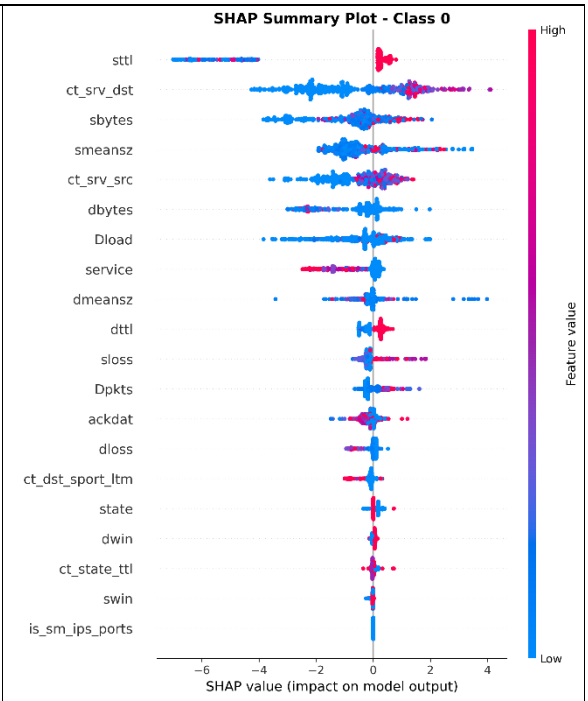
χαρακτηριστικά is_sm_ips_ports, swin και dwin παρουσιάζουν ουσιαστικά μηδενική τιμή SHAP.

Για την κλάση 8 που αντιστοιχεί στις επιθέσεις Worms (Εικόνα 207), το smeansz κυριαρχεί παρουσιάζοντας μπλε σημεία αριστερά (≈ −5.0) και κόκκινα σημεία δεξιά (≈ +0.5), από τα οποία προκύπτει ότι οι χαμηλές τιμές του μέσου μεγέθους των πακέτων πηγής μειώνουν σημαντικά την πιθανότητα αναγνώρισης της συγκεκριμένης κλάσης. Το sbytes (πλειοψηφικά) εμφανίζει μπλε σημεία αριστερά (≈ −2.5) και κόκκινα σημεία δεξιά (≈ +2.0), υποδηλώνοντας ότι ο υψηλός αριθμός bytes που στάλθηκαν από την πηγή προς τον προορισμό αυξάνει την πιθανότητα ανίχνευσης. Το Dload παρουσιάζει μπλε σημεία αριστερά και κόκκινα σημεία δεξιά (≈ +1.5), που σημαίνει ότι ο υψηλός ρυθμός λήψης αποτελεί χαρακτηριστικό δείκτη της κλάσης. Το dbytes φαίνεται με μπλε σημεία αριστερά (≈ −2.0) και κόκκινα σημεία δεξιά (≈ +5.5), δείχνοντας με αυτό το τρόπο ότι εξαιρετικά υψηλός αριθμός bytes προορισμού αποτελεί επίσης ισχυρό δείκτη της κλάσης, ενώ το ackdat παρουσιάζει μωβ σημεία αριστερά (≈ −2.5) και κόκκινα σημεία δεξιά (≈ +3.5).

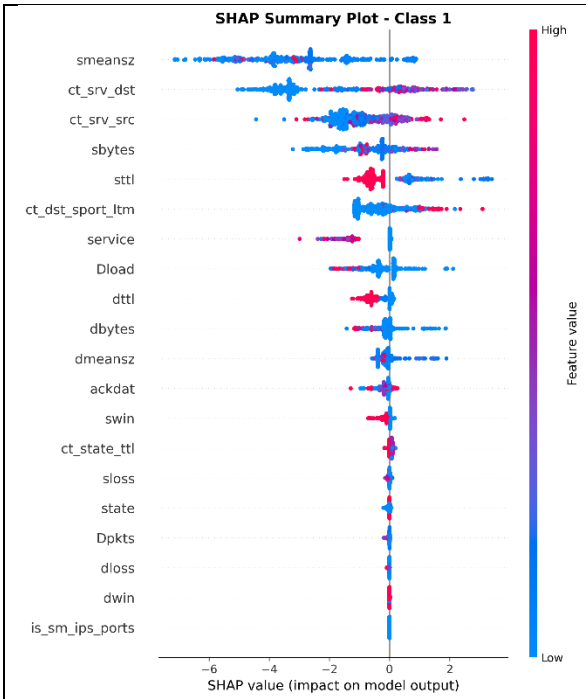
Για την κλάση 9 που αντιστοιχεί στην κανονική κίνηση (Εικόνα 208), το ct_state_ttl παρουσιάζεται με διπλό μοτίβο με κόκκινα και μωβ σημεία αριστερά (≈ −2.0) και μπλε σημεία δεξιά (έως +11.5), τα οποία υποδεικνύουν ότι οι χαμηλές τιμές μειώνουν ενώ οι υψηλές αυξάνουν δραστικά την πιθανότητα αναγνώρισης της εν λόγω κλάσης. Το dttl εμφανίζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία κοντά στο μηδέν, ενώ το sttl παρουσιάζει κόκκινα σημεία πιο κοντά στο μηδέν και μπλε σημεία δεξιά (≈ +4.5), από τα οποία προκύπτει ότι οι χαμηλές τιμές των TTL πηγής αυξάνουν ήπια την πιθανότητα κανονικής κίνησης. Το smeansz εμφανίζει μπλε σημεία αριστερά (≈ −1), ενώ το ct_dst_sport_ltm παρουσιάζει κόκκινα σημεία αριστερά (≈ −1.5) και μπλε σημεία κοντά στο μηδέν, υποδηλώνοντας ότι ο υψηλός αριθμός των συνδέσεων σε διαφορετικές πόρτες μειώνει την πιθανότητα κανονικής κίνησης. Τα χαρακτηριστικά is_sm_ips_ports, state και dwin παρουσιάζουν πρακτικά μηδενική τιμή SHAP.



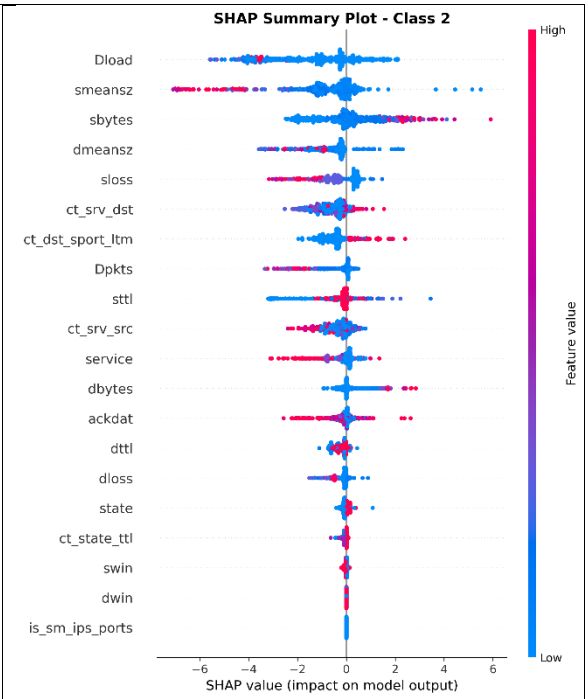
Εικόνα 198 Summary Plot, όλες οι κλάσεις για τον 21ο συνδυασμό.



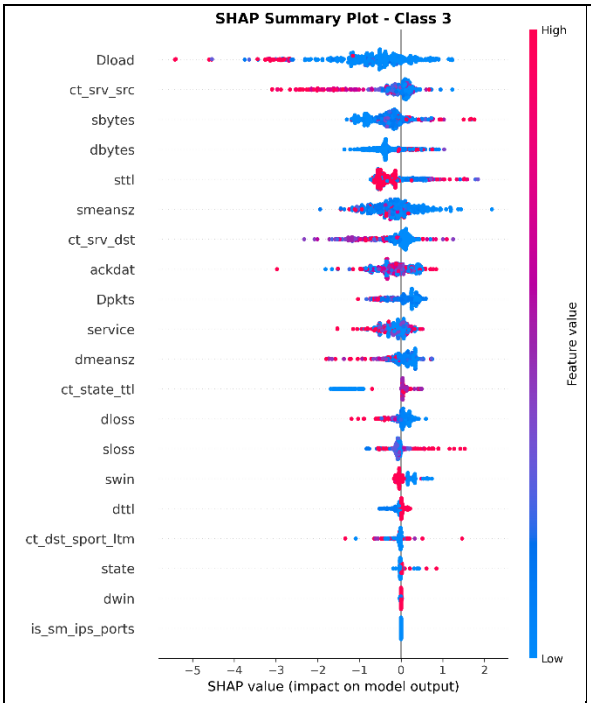
Εικόνα 199 SHAP Summary Plot, κλάση 0 για τον 21ο συνδυασμό.



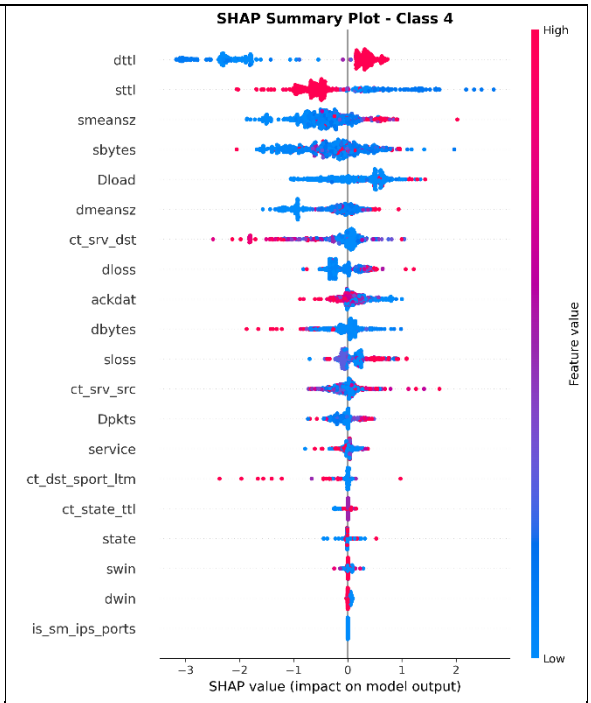
Εικόνα 200 SHAP Summary Plot, κλάση 1 για τον 21ο συνδυασμό.



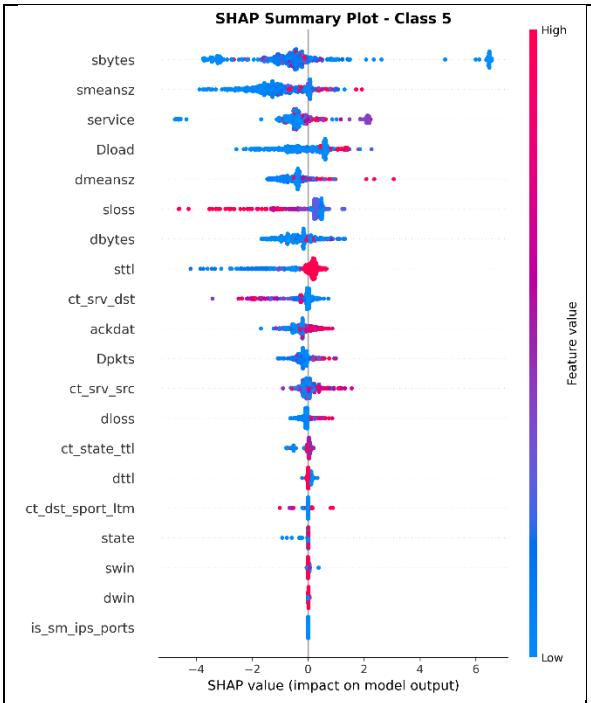
Εικόνα 201 SHAP Summary Plot, κλάση 2 για τον 21ο συνδυασμό.



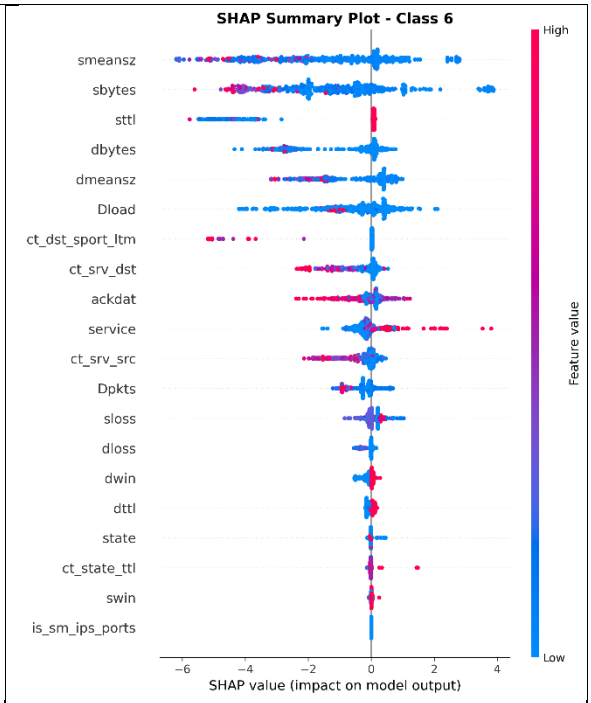
Εικόνα 202 SHAP Summary Plot, κλάση 3 για τον 21ο συνδυασμό.



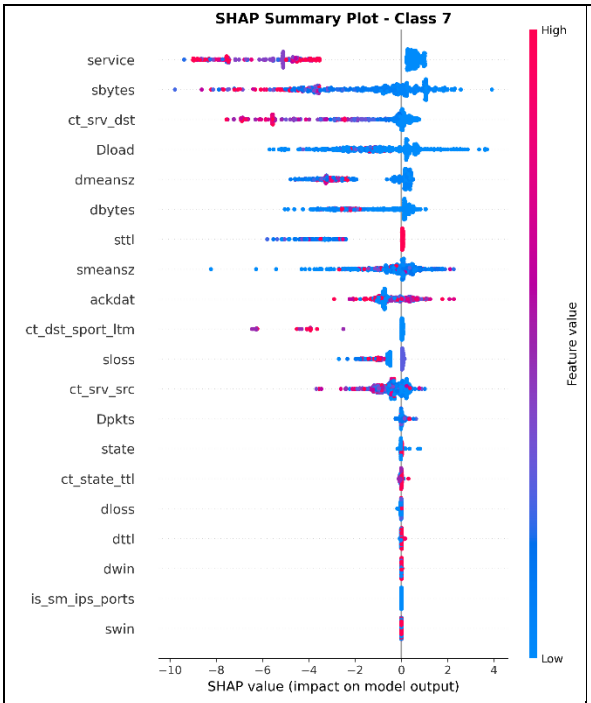
Εικόνα 203 SHAP Summary Plot, κλάση 4 για τον 21ο συνδυασμό.



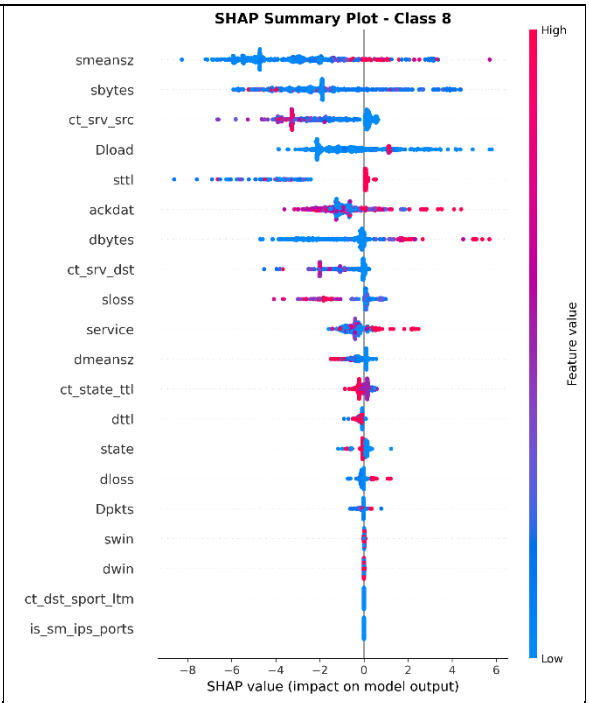
Εικόνα 204 SHAP Summary Plot, κλάση 5 για τον 21ο συνδυασμό.



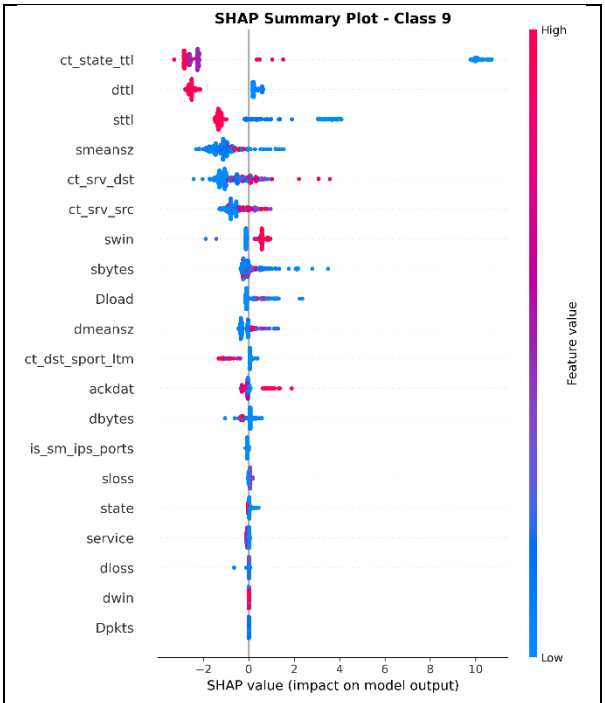
Εικόνα 205 SHAP Summary Plot, κλάση 6 για τον 21ο συνδυασμό.



Εικόνα 206 SHAP Summary Plot, κλάση 7 για τον 21ο συνδυασμό.



Εικόνα 207 SHAP Summary Plot, κλάση 8 για τον 21ο συνδυασμό.



Εικόνα 208 SHAP Summary Plot, κλάση 9 για τον 21ο συνδυασμό.

6.4. Συγκριτική ανάλυση αποτελεσμάτων εκπαίδευσης

Στο παρόν κεφάλαιο πραγματοποιείται συγκριτική ανάλυση των αποτελεσμάτων εκπαίδευσης που παρουσιάστηκαν αναλυτικά στο προηγούμενο κεφάλαιο. Η ανάλυση οργανώνεται σε δύο επίπεδα: αρχικά, συγκρίνονται οι βέλτιστοι συνδυασμοί κάθε συνόλου δεδομένων με αντίστοιχες μεθοδολογίες της σύγχρονης βιβλιογραφίας, ώστε να αξιολογηθεί η ανταγωνιστικότητα της προτεινόμενης προσέγγισης σε σχέση με την υπάρχουσα έρευνα, ενώ στη συνέχεια εξετάζονται συγκριτικά τα πρότυπα σφαλμάτων που αναδεικνύουν οι πίνακες σύγχυσης των 21 συνδυασμών, με στόχο την κατανόηση της ταξινομητικής συμπεριφοράς του μοντέλου ανά τύπο επίθεσης και μέθοδο εξισορρόπησης.

6.4.1. Σύγκριση μετρικών με τη βιβλιογραφία

Για τη σύγκριση με τη βιβλιογραφία επιλέχθηκαν οι βέλτιστοι συνδυασμοί για καθένα από τα τέσσερα σύνολα δεδομένων που χρησιμοποιήθηκαν στην παρούσα εργασία, δηλαδή ο συνδυασμός 4 από το Wednesday-workingHours (DoS επιθέσεις), ο συνδυασμός 11 από το Thursday-WorkingHours-Morning-WebAttacks (Web επιθέσεις), ο συνδυασμός 9 από το Friday-WorkingHours-Afternoon-PortScan (PortScan επιθέσεις) και ο συνδυασμός 19 από το UNSW-NB15. Για κάθε σύνολο δεδομένων επιλέχθηκε μία ή δύο αντιπροσωπευτικές εργασίες από τη βιβλιογραφία που χρησιμοποιούν το ίδιο σύνολο δεδομένων, ώστε να υπάρξει άμεση και δίκαιη σύγκριση υπό τις ίδιες πειραματικές συνθήκες.

Για το σύνολο δεδομένων Friday-WorkingHours-Afternoon-PortScan του CICIDS2017, η σύγκριση πραγματοποιείται με τους συγγραφείς [14] να προτείνουν ένα σύστημα IDS βασισμένο στον συνδυασμό Conditional Generative Adversarial Network (CGAN) και XGBoost ταξινομητή. Στη μεθοδολογία που αναφέρεται, το CGAN χρησιμοποιείται για την παραγωγή συνθετικών δεδομένων κίνησης δικτύου με σκοπό να παραπλανήσει πιθανούς εισβολείς και να μειώσει τον αριθμό των φυσικών αισθητήρων που απαιτούνται στο δίκτυο, ενώ ο XGBoost αναλαμβάνει την ταξινόμηση της κίνησης σε κανονική και PortScan επίθεση. Το σύνολο δεδομένων που χρησιμοποιήθηκε είναι ακριβώς το ίδιο με τον συνδυασμό 9 της παρούσας εργασίας, δηλαδή το Friday-WorkingHours-Afternoon-PortScan του CICIDS2017, με 127.537 BENIGN ροές και 158.930 PortScan ροές.

Τα αποτελέσματα που αναφέρονται [14] για το συγκεκριμένο σύνολο δεδομένων φτάνουν σε ορθότητα 99.95% και ακρίβεια 99.91%, ενώ δεν αναφέρεται ξεχωριστή τιμή βαθμολογίας F1 για το PortScan σύνολο δεδομένων. Ο βέλτιστος συνδυασμός 9 της

παρούσας εργασίας, τον οποίο αποτυπώνει επίσης ο Πίνακας 7 και ο οποίος χρησιμοποιεί XGBoost με υβριδική μέθοδο εξισορρόπησης (Hybrid SMOTE+ENN) και αντικατάσταση των NaN τιμών, επιτυγχάνει ορθότητα 99.97%, ακρίβεια 99.97%, ανάκληση 99.97% και βαθμολογία F1 99.97%, υπερτερώντας ελαφρώς σε όλες τις μετρικές.

Η ελαφρά υπεροχή του συνδυασμού 9 είναι αξιοσημείωτη, ειδικότερα αν ληφθεί υπόψη η πολυπλοκότητα των δύο προσεγγίσεων. Στην εργασία [14] εκπαιδεύεται ένα πολύπλοκο μοντέλο CGAN βαθυάς μάθησης με Generator και Discriminator δίκτυα τεσσάρων πλήρως συνδεδεμένων στρωμάτων, 5.000 βήματα εκπαίδευσης και Adam optimizer, προτού παραδώσουν τα δεδομένα στον XGBoost ταξινομητή. Η δική μας προσέγγιση επιτυγχάνει ανώτερα αποτελέσματα χρησιμοποιώντας αποκλειστικά τον XGBoost με βελτιστοποίηση υπερπαραμέτρων μέσω RandomizedSearchCV και υβριδική εξισορρόπηση, χωρίς να απαιτείται εκπαίδευση πολύπλοκων γεννητικών μοντέλων. Αυτό υποδηλώνει ότι για το συγκεκριμένο πρόβλημα δυαδικής ταξινόμησης PortScan, η αποτελεσματική βελτιστοποίηση ενός ισχυρού αλγορίθμου μπορεί να αποδώσει εξίσου καλά ή και καλύτερα από πολύπλοκα υβριδικά βαθυάς μάθησης συστήματα.

Για το σύνολο δεδομένων Thursday-WorkingHours-Morning-WebAttacks του CICIDS2017, η σύγκριση πραγματοποιείται με την εργασία [16], στην οποία προτείνεται μια ανθρωποκεντρική μεθοδολογία IDS που συνδυάζει XGBoost ως προγνωστικό μοντέλο με δέντρα αποφάσεων ως μοντέλο επεξήγησης, εμπλουτισμένη με τιμές SHAP και Feature Outlier Score (FOS) για την ανίχνευση ύποπτων προβλέψεων. Το σύνολο δεδομένων που χρησιμοποιείται είναι ακριβώς το ίδιο με τους συνδυασμούς 3, 5, 8, 11, 14 και 17 της παρούσας εργασίας, με 168.186 κανονικές ροές, 1.507 Brute Force, 652 XSS και 21 SQL Injection. Η προεπεξεργασία περιλαμβάνει αφαίρεση 18 χαρακτηριστικών, Min-Max scaling, random under-sampling για την κανονική κίνηση και SMOTE για τις μειονοτικές κλάσεις, με αναλογία 70/30 για εκπαίδευση και αξιολόγηση.

Μέθοδος	**Ορθότητα**	**Ακρίβεια**	**Ανάκληση**	**Βαθμολογία F1**
CGAN + XGBoost	99.95%	99.91%	-	-
Συνδυασμός 9	99.97%	99.97%	99.97%	99.97%

Πίνακας 7 Σύγκριση Friday-WorkingHours-Afternoon

Μέθοδος	Ορθότητα	Ακρίβεια	Ανάκληση	Βαθμολογία F1
XGBoost + DT + SHAP	90.60%	89.70%	90.60%	90.09%
Συνδυασμός 11	99.48%	99.52%	99.48%	99.50%

Πίνακας 8 Σύγκριση Thursday-WorkingHours-Morning-WebAttacks

Τα αποτελέσματα που επιτυγχάνονται [16] με τον XGBoost ταξινομητή στο συγκεκριμένο σύνολο δεδομένων φτάνουν σε ορθότητα 90.60%, ακρίβεια 89.70%, ανάκληση 90.60% και βαθμολογία F1 90.09%. Ο βέλτιστος συνδυασμός 11 της παρούσας εργασίας (Πίνακας 8), ο οποίος χρησιμοποιεί XGBoost με SMOTE εξισορρόπηση και διαγραφή των NaN τιμών, επιτυγχάνει ορθότητα 99.48%, ακρίβεια 99.52%, ανάκληση 99.48% και βαθμολογία F1 99.50%, υπερτερώντας σημαντικά σε όλες τις μετρικές.

Η σημαντική διαφορά απόδοσης μεταξύ των δύο προσεγγίσεων, η οποία αγγίζει σχεδόν 9 ποσοστιαίες μονάδες σε βαθμολογία F1, αποδίδεται σε διαφορετικούς λόγους. Πρώτον, στην εργασία [16] δεν εστιάζουν στη μεγιστοποίηση της ταξινομητικής απόδοσης αλλά στην ερμηνευσιμότητα του συστήματος, το μοντέλο δέντρων απόφασης εκπαιδεύεται μόνο στα 10 κορυφαία χαρακτηριστικά βάσει information gain, και ο XGBoost χρησιμοποιείται με παραμέτρους που δεν έχουν υποστεί εκτεταμένη βελτιστοποίηση. Δεύτερον, η παρούσα εργασία αξιοποιεί RandomizedSearchCV για συστηματική βελτιστοποίηση υπερπαραμέτρων, η οποία επιτρέπει στο μοντέλο να προσαρμοστεί βέλτιστα στα χαρακτηριστικά του συγκεκριμένου συνόλου δεδομένων. Τρίτον, η χρήση SMOTE εξισορρόπησης στην παρούσα εργασία αντιμετωπίζει αποτελεσματικότερα τη σοβαρή ανισορροπία κλάσεων που χαρακτηρίζει το Thursday dataset, ιδιαίτερα για την κλάση SQL Injection με μόλις 21 δείγματα. Τα αποτελέσματα αυτά επιβεβαιώνουν ότι η συστηματική βελτιστοποίηση υπερπαραμέτρων και η κατάλληλη εξισορρόπηση κλάσεων αποτελούν ιδιαίτερα σημαντικούς παράγοντες για την επίτευξη υψηλής ταξινομητικής απόδοσης, ακόμα και όταν χρησιμοποιείται ο ίδιος βασικός αλγόριθμος.

Για το σύνολο δεδομένων Wednesday-workingHours του CICIDS2017, η σύγκριση πραγματοποιείται με την εργασία [43] στην οποία προτείνεται το ROSE-BOX (τυχαίο δάσος, SMOTE, BOX-gboost), μια σύνθετη μεθοδολογία τριών σταδίων για την ανίχνευση πολλαπλών κυβερνοεπιθέσεων. Συγκεκριμένα, το ROSE-BOX χρησιμοποιεί τυχαίο δάσος για την κατάταξη της σημαντικότητας χαρακτηριστικών με EarlyStopping strategy, SMOTE για την εξισορρόπηση του συνόλου δεδομένων, και XGBoost με Bayesian Optimization (BO-XGBoost) για τη βελτιστοποίηση των υπερπαραμέτρων. Το σύνολο δεδομένων Wednesday-workingHours του CICIDS2017 περιλαμβάνει BENIGN κίνηση και πέντε κατηγορίες DoS επιθέσεων (DoS Hulk, DoS GoldenEye, DoS Slowhttptest, DoS slowloris, Heartbleed), αποτελώντας ένα από τα πιο απαιτητικά σενάρια πολυκλασικής ταξινόμησης λόγω της σοβαρής ανισορροπίας των κλάσεων, ιδίως για την κλάση Heartbleed με μόλις 11 δείγματα.

Τα αποτελέσματα που επιτυγχάνει το ROSE-BOX στο σύνολο δεδομένων Wednesday, όπως δείχνει και ο Πίνακας 9, φτάνουν σε ορθότητα 99.991%, ακρίβεια 99.993%, ανάκληση 99.987% και βαθμολογία F1 99.989%, με AUROC 1.0. Ο βέλτιστος συνδυασμός 4 της παρούσας εργασίας, ο οποίος χρησιμοποιεί XGBoost με SMOTE-ENN εξισορρόπηση και αντικατάσταση των NaN τιμών, επιτυγχάνει ορθότητα 99.36%, ακρίβεια 99.36%, ανάκληση 99.36% και βαθμολογία F1 99.36%.

Το ROSE-BOX επιτυγχάνει ελαφρώς ανώτερη απόδοση, με διαφορά περίπου 0.63 ποσοστιαίων μονάδων σε βαθμολογία F1. Ωστόσο, η διαφορά αυτή πρέπει να ερμηνευθεί υπό το πρίσμα της σημαντικά διαφορετικής πολυπλοκότητας των δύο μεθοδολογιών. Το ROSE-BOX υλοποιεί τρία διακριτά στάδια βελτιστοποίησης: επιλογή χαρακτηριστικών μέσω τυχαίου δάσους με EarlyStopping, εξισορρόπηση δεδομένων μέσω SMOTE και βελτιστοποίηση υπερπαραμέτρων XGBoost μέσω Bayesian Optimization, η οποία είναι γενικά αποδοτικότερη αλλά και υπολογιστικά πιο απαιτητική από τη RandomizedSearchCV. Η παρούσα εργασία επιτυγχάνει συγκρίσιμη απόδοση στο ίδιο εξαιρετικά απαιτητικό πολυκλασικό σενάριο χρησιμοποιώντας μια απλούστερη και υπολογιστικά αποδοτικότερη μεθοδολογία, γεγονός που αναδεικνύει την αποτελεσματικότητα της προτεινόμενης προσέγγισης. Επιπλέον, αξίζει να σημειωθεί ότι και οι δύο μεθοδολογίες επιτυγχάνουν εξαιρετικά υψηλές τιμές σε όλες τις μετρικές, οι οποίες ξεπερνούν κατά πολύ τις αντίστοιχες τιμές παλαιότερων προσεγγίσεων όπως KNN

Μέθοδος	Ορθότητα	Ακρίβεια	Ανάκληση	Βαθμολογία F1
ROSE-BOX	99.991%	99.993%	99.987%	99.989%
Συνδυασμός 4	99.36%	99.36%	99.36%	99.36%

Πίνακας 9 Σύγκριση Wednesday-workingHours

(F1: 98.867%), CNN (F1: 95.732%) και LSTM (F1: 31.859%) όπως αναφέρεται στην ίδια εργασία [43], επιβεβαιώνοντας την ανωτερότητα των XGBoost-βασισμένων προσεγγίσεων για την ανίχνευση DoS επιθέσεων.

Για το σύνολο δεδομένων UNSW-NB15, η σύγκριση με τη βιβλιογραφία παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς οι περισσότερες εργασίες προσεγγίζουν το σύνολο δεδομένων διαφορετικά από την παρούσα εργασία. Ο συνδυασμός 19, ο βέλτιστος για το UNSW-NB15, εφαρμόζει πολυκλασική ταξινόμηση 10 κλάσεων (Normal και 9 κατηγορίες επιθέσεων: Analysis, Backdoors, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode και Worms) επιτυγχάνοντας ορθότητα 99.15%, ακρίβεια 99.36%, ανάκληση 99.15% και βαθμολογία F1 99.23%. Για τη σύγκριση επιλέχθηκαν δύο αντιπροσωπευτικές εργασίες από τη βιβλιογραφία που χρησιμοποιούν το ίδιο σύνολο δεδομένων, αναδεικνύοντας τόσο ομοιότητες όσο και διαφορές ως προς το πρόβλημα ταξινόμησης που αντιμετωπίζουν.

Στην πρώτη εργασία [26], η οποία αποτελεί το πιο άμεσα συγκρίσιμο σημείο αναφοράς, οι συγγραφείς προτείνουν ένα σύνθετο ensemble μοντέλο που συνδυάζει Convolutional Neural Network (CNN) για εξαγωγή χαρακτηριστικών, Bidirectional LSTM (BiLSTM) για ανάλυση χρονικών δεδομένων και τυχαίου δάσους για την τελική ταξινόμηση, ενσωματώνοντας τεχνικές XAI, καθώς επίσης SHAP και LIME για ερμηνευσιμότητα. Τα αποτελέσματα που παρουσιάζονται για το UNSW-NB15 φτάνουν σε ορθότητα 97.71%, ακρίβεια 98.40%, ανάκληση 97.41% και βαθμολογία F1 97.90%, με FPR μόλις 0.01. Ο βέλτιστος συνδυασμός 19 της παρούσας εργασίας υπερτερεί σε όλες τις μετρικές, με τη διαφορά στην ορθότητα να ανέρχεται σε 1.44 ποσοστιαίες μονάδες (99.15% έναντι 97.71%) και στο βαθμολογία F1 σε 1.33 ποσοστιαίες μονάδες (99.23% έναντι 97.90%). Αυτό καθίσταται ακόμα πιο αξιοσημείωτο αν ληφθεί υπόψη ότι η παρούσα εργασία λύνει ένα σαφώς δυσκολότερο πρόβλημα 10 κλάσεων, ενώ η εργασία με την οποία γίνετε η σύγκριση

εφαρμόζει δυαδική ταξινόμηση στο UNSW-NB15, καθιστώντας τη σύγκριση ακόμα πιο ευνοϊκή για την προτεινόμενη μεθοδολογία.

Στην δεύτερη εργασία [24] στην οποία εφαρμόζεται δυαδική ταξινόμηση (Normal έναντι Attack) στο UNSW-NB15 χρησιμοποιώντας τους τρεις ταξινομητές δέντρα απόφασης, MLP και XGBoost, με SHAP και LIME για ερμηνευσιμότητα. Στο test set των 82.332 εγγραφών, ο XGBoost με feature importance επιτυγχάνει ορθότητα 98.13%, αποτελώντας τον κορυφαίο ταξινομητή μεταξύ των τριών μοντέλων που δοκιμάστηκαν. Η σύγκριση αυτή δεν είναι απόλυτα ισότιμη, καθώς το πρόβλημα δυαδικής ταξινόμησης (Normal έναντι Attack) είναι εκ φύσεως απλούστερο από το πολυκλασικό σενάριο 10 κλάσεων της παρούσας εργασίας. Παρ' όλα αυτά, ο συνδυασμός 19 επιτυγχάνει ορθότητα 99.15% στο σαφώς πιο απαιτητικό πολυκλασικό πρόβλημα, υπερτερώντας κατά 1.02 ποσοστιαίες μονάδες ακόμα και σε σχέση με αυτό το απλούστερο σενάριο. Ο Πίνακας 10 αποτυπώνει συγκεντρωτικά την εν λόγω σύγκριση

Μέθοδος	**Ορθότητα**	**Ακρίβεια**	**Ανάκληση**	**Βαθμολογία F1**
XGBoost + SHAP/LIME	98.13%	-	-	-
CNN + BiLSTM + RF	97.71%	98.40%	97.41%	97.90%
Συνδυασμός 19	99.15%	99.36%	99.15%	99.23%

Πίνακας 10 Σύγκριση UNSW-NB15

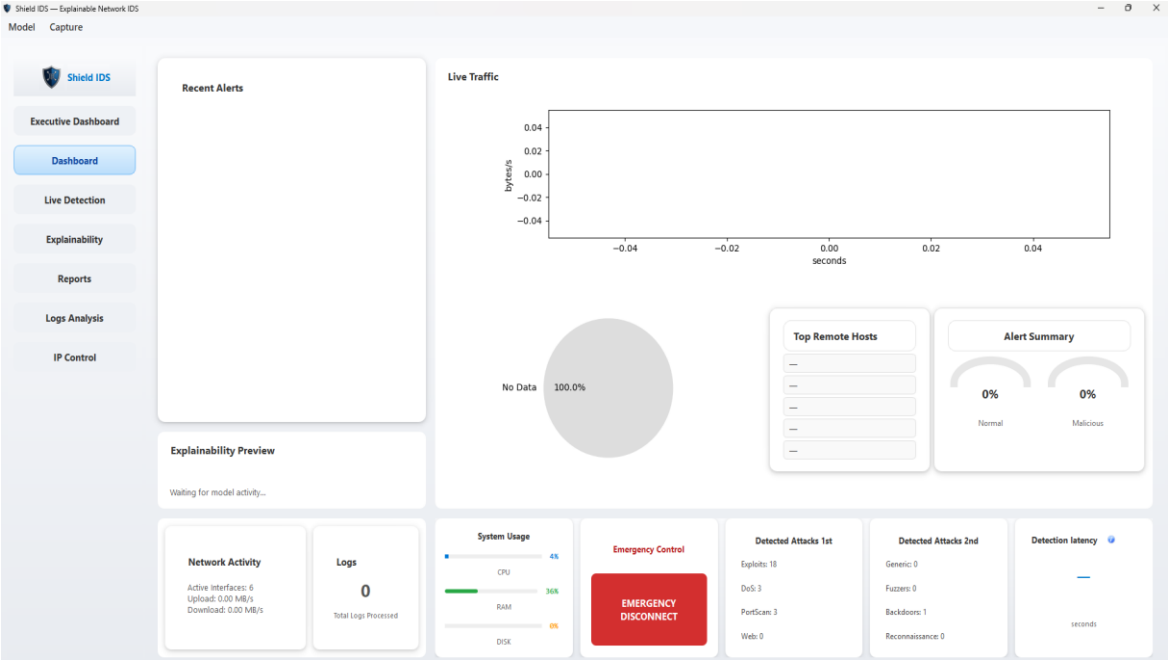
7. Κύρια Εφαρμογή και Λειτουργικές Ενότητες

Στην παρούσα διπλωματική εργασία αναπτύχθηκε η εφαρμογή Shield IDS (αρχική οθόνη της οποίας δείχνει η Εικόνα 209) με σκοπό την παροχή ενός ολοκληρωμένου συστήματος ανίχνευσης εισβολών με γραφικό περιβάλλον χρήστη. Επιτρέπει στους χρήστες να παρακολουθούν την δικτυακή κίνηση σε πραγματικό χρόνο έτσι ώστε να εντοπίζουν τυχόν κακόβουλη δραστηριότητα στο δίκτυο.

Η εφαρμογή έχει σχεδιαστεί και υλοποιηθεί με τέτοιο τρόπο έτσι ώστε να παρέχει στον χρήστη μια εύκολη πλοήγηση. Για την επίτευξη αυτού υλοποιήθηκε ένα εύχρηστο μενού για την φόρτωση των μοντέλων αλλά και για την εκκίνηση και διακοπή της λειτουργίας της εφαρμογής. Επίσης δημιουργήθηκε μια πλαϊνή γραμμή πλοήγησης με καρτέλες που καθεμία από αυτές εξυπηρετεί διαφορετικό σκοπό. Η βασική καρτέλα η οποία εμφανίζεται όταν εκτελείται η εφαρμογή είναι το Dashboard που παρέχει μια συνολική εικόνα στους τεχνικούς ασφαλείας. Η πρώτη καρτέλα της πλαϊνής γραμμής πλοήγησης αφορά τον πίνακα ελέγχου στελεχών (Executive Dashboard) ο οποίος παρέχει συγκεντρωτικά και κατανοητά στοιχεία για τα στελέχη της διοίκησης που δεν έχουν τεχνικές γνώσεις. Η επόμενη καρτέλα η οποία είναι και η βασική, είναι το Dashboard ενώ στην συνέχεια συναντάμε την ζωντανή ανίχνευση - Live Detection που επιτρέπει την παρακολούθηση της δικτυακής κίνησης σε πραγματικό χρόνο και εμφανίζει τα χαρακτηριστικά της κάθε ροής. Επίσης οπτικοποιεί την κανονική κίνηση με πράσινο χρώμα και την κακόβουλη κίνηση με κόκκινο.

Η καρτέλα Explainability παρέχει επεξηγήσεις για τις αποφάσεις των μοντέλων μηχανικής μάθησης που χρησιμοποιούνται για την ανίχνευση επιθέσεων ενισχύοντας με αυτό τον τρόπο την διαφάνεια του μοντέλου. Η λειτουργία αυτή βασίζεται στην μέθοδο ερμηνείας SHAP, η οποία επιτρέπει την ανάδειξη της συμβολής των χαρακτηριστικών εισόδου στην τελική πρόβλεψη. Η επόμενη καρτέλα αποδίδει τις αναφορές, παρουσιάζει δηλαδή συγκεντρωτικά τις αναφορές σχετικά με τις επιθέσεις που έχουν εντοπιστεί από το σύστημα. Η καρτέλα Logs Analysis επιτρέπει την προβολή και ανάλυση των καταγεγραμμένων δεδομένων δικτυακής δραστηριότητας ενώ η καρτέλα IP Control δίνει τη δυνατότητα διαχείρισης λιστών επιτρεπόμενων και αποκλεισμένων διευθύνσεων IP.

Στόχος της εφαρμογής είναι η εποπτεία και η ανάλυση της ασφάλειας του δικτύου σε πραγματικό χρόνο, προσφέροντας την απαραίτητη πληροφόρηση τόσο στους τεχνικούς ασφαλείας όσο και στα στελέχη της διοίκησης.



Εικόνα 209 Εφαρμογή Shield IDS

Προκειμένου να επαληθευτεί η αποτελεσματικότητα του Shield IDS σε συνθήκες πραγματικής κυβερνοεπίθεσης, διεξήχθη μια σειρά από δοκιμές ελεγχόμενης επίθεσης (controlled attack testing) σε τοπικό δικτυακό περιβάλλον. Για τον σκοπό αυτό αξιοποιήθηκε συνδυασμός καθιερωμένων εργαλείων δοκιμών διείσδυσης και κυβερνοεπιθέσεων, καθώς και τέσσερα εργαλεία που αναπτύχθηκαν εξ ολοκλήρου από τους συγγραφείς σε Python.

Το Apache JMeter χρησιμοποιήθηκε ως εργαλείο δοκιμής φορτίου (load testing), επιτρέποντας τη δημιουργία μαζικής κίνησης HTTP/HTTPS προς έναν διακομιστή-στόχο. Μέσω της παραμετροποίησης του αριθμού ταυτόχρονων εικονικών χρηστών και του ρυθμού αποστολής αιτημάτων, αναπαράχθηκαν σενάρια κορεσμού πόρων που αντιστοιχούν στις κατηγορίες DoS/DDoS (HTTP Flood) του CIC-IDS2017, αξιολογώντας την ικανότητα του μοντέλου να διακρίνει υψηλόσυχνη κακόβουλη HTTP κίνηση από κανονική χρήση. Το LOIC (Low Orbit Ion Cannon) χρησιμοποιήθηκε για την παραγωγή επιθέσεων τύπου DoS σε επίπεδο TCP, UDP και HTTP, στοχεύοντας στην αξιολόγηση της ανίχνευσης επιθέσεων DoS Hulk, DoS Slowloris και DoS GoldenEye, κατηγοριών που περιλαμβάνονται στο υποσύνολο Wednesday του CIC-IDS2017, καθώς και επιθέσεων τύπου DoS του UNSW-NB15, οι οποίες χαρακτηρίζονται από κορεσμό εύρους ζώνης και εξάντληση πόρων. Το OWASP, μέσω των εργαλείων και μεθοδολογιών που παρέχει, χρησιμοποιήθηκε για τη δοκιμή επιθέσεων σε εφαρμογές ιστού, καλύπτοντας κατηγορίες

όπως SQL Injection, Cross-Site Scripting (XSS) και brute-force authentication attacks, που αντιστοιχούν άμεσα στις κατηγορίες Web Attacks (Brute Force, XSS, SQL Injection) του τμήματος Thursday του CIC-IDS2017, καθώς και στις κατηγορίες Exploits και Generic του UNSW-NB15. Τέλος, το Nmap χρησιμοποιήθηκε για την εκτέλεση σαρώσεων θυρών διαφόρων τύπων, όπως SYN scan, TCP connect scan και UDP scan, προσομοιώνοντας τις κατηγορίες PortScan του υποσυνόλου δεδομένων Friday του CIC-IDS2017 και Reconnaissance του UNSW-NB15, αξιολογώντας κατά πόσον το μοντέλο εντοπίζει συστηματικές σαρώσεις τόσο σε πιο επιθετικό όσο και σε χαμηλότερης έντασης ρυθμό.

Επιπλέον, αναπτύχθηκαν τέσσερα εργαλεία προσομοίωσης επιθέσεων σε Python (και σε περιβάλλον RPi 5, Linux kali), σχεδιασμένα ώστε τα εξαγόμενα χαρακτηριστικά ροής να είναι πλήρως συμβατά με τα χαρακτηριστικά του UNSW-NB15. Το backdoors.py προσομοιώνει μυστική επικοινωνία μεταξύ παραβιασμένου κόμβου και διακομιστή ελέγχου (C&C), αποστέλλοντας κίνηση με χαμηλή συχνότητα και ασύμμετρο ρυθμό πακέτων, στοχεύοντας στην κατηγορία Backdoors του UNSW-NB15. Το exploits.py κατασκευάζει και αποστέλλει μια σειρά από εντολές (No Operation - NOP sled) που εκτελούνται διαδοχικά από τον επεξεργαστή, αναγκάζοντας το πρόγραμμα να «γλιστρήσει» προς τα κάτω στη μνήμη μέχρι να φτάσει στον πραγματικό κώδικα - στόχο (ψευδο-shellcode), αναπαράγοντας έτσι την υπογραφή επίθεσης buffer overflow που αντιστοιχεί στις κατηγορίες Exploits του UNSW-NB15. Το generic.py υλοποιεί επίθεση τύπου generic αποστέλλοντας τυχαία malformed δεδομένα αυξανόμενου μεγέθους (από 1KB έως αρκετά MB) μέσω του TCP socket, αντιστοιχώντας στην κατηγορία Generic του UNSW-NB15. Το fuzzers_backdoors_recon.py αξιοποιεί τη βιβλιοθήκη scapy για την κατασκευή raw IP πακέτων πρωτοκόλλου OSPF (proto=89) με σταθερό TTL=254 και υψηλό jitter χρονισμού, αναπαράγοντας με ακρίβεια τις στατιστικές υπογραφές ροής (smeansz, Sjit, Spkts) που χαρακτηρίζουν ταυτόχρονα τις κατηγορίες Fuzzers, Backdoors και Reconnaissance του UNSW-NB15. Το εργαλείο αυτό υποστηρίζει επίσης IP spoofing για την προσομοίωση κατανεμημένης επίθεσης από πολλαπλές πηγές.

Όλες οι δοκιμές διεξήχθησαν σε απομονωμένο τοπικό δίκτυο με αρχιτεκτονική επιτιθέμενου – στόχου (attacker–victim), ώστε να αποφευχθεί οποιαδήποτε επίπτωση σε παραγωγικά συστήματα. Το Shield IDS εκτελούνταν σε πραγματικό χρόνο (live mode), λαμβάνοντας και ταξινομώντας τα χαρακτηριστικά ροής μέσω του module packet capturing, με τα αποτελέσματα να απεικονίζονται άμεσα στο κεντρικό dashboard και στο

περιβάλλον explainability, επιβεβαιώνοντας τη λειτουργική ολοκλήρωση του συστήματος από άκρο σε άκρο.

7.1. Λειτουργίες μενού: Φόρτωση μοντέλων και έλεγχος σύλληψης

Η εφαρμογή διαθέτει στο επάνω μέρος της διεπαφής χρήστη ένα βασικό μενού λειτουργιών το οποίο επιτρέπει τη φόρτωση των μοντέλων μηχανικής μάθησης και τη διαχείριση της διαδικασίας σύλληψης της δικτυακής κίνησης. Οι παραπάνω λειτουργίες αποτελούν τον βασικό μηχανισμό παραμετροποίησης και ενεργοποίησης της εφαρμογής, καθώς μέσω αυτών ο χρήστης επιλέγει τα μοντέλα ανίχνευσης και ελέγχει την έναρξη ή διακοπή της παρακολούθησης του δικτύου.

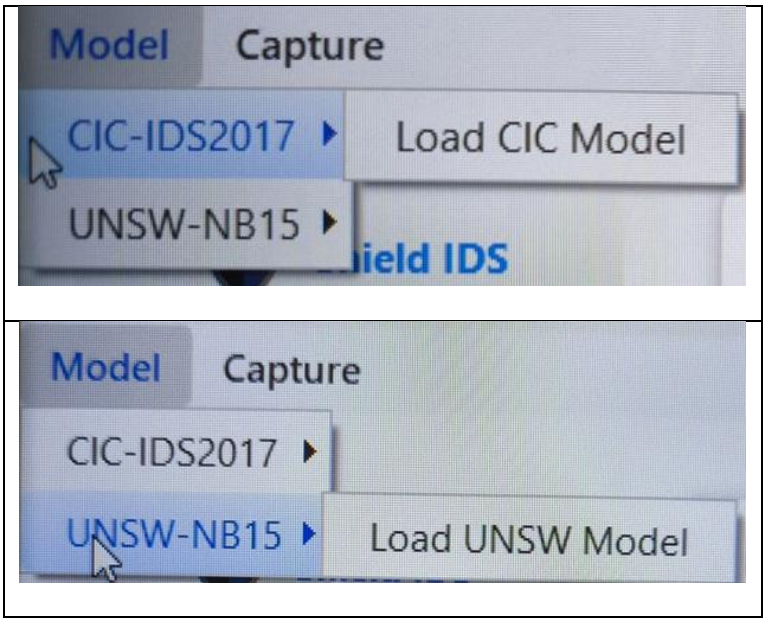
Για την φόρτωση των μοντέλων έχει υλοποιηθεί ένα αναδυόμενο μενού (όπως φαίνεται στην Εικόνα 210) το οποίο είναι οργανωμένο σε δύο κατηγορίες οι οποίες αντιστοιχούν στα σύνολα δεδομένων CIC-IDS2017 και UNSW-NB15. Για το CIC-IDS2017, η εφαρμογή υποστηρίζει φόρτωση περισσότερων επιμέρους μοντέλων τα οποία έχουν εκπαιδευτεί για διαφορετικές οικογένειες επιθέσεων, όπως επιθέσεις άρνησης υπηρεσίας, επιθέσεις σε διαδικτυακές εφαρμογές και σάρωση θυρών. Μαζί με κάθε μοντέλο είναι δυνατό να φορτωθούν και οι αντίστοιχοι κωδικοποιητές ετικετών (label encoders), ώστε οι αριθμητικές έξοδοι των ταξινομητών να μετατρέπονται σε κατανοητές ονομασίες επιθέσεων. Για το UNSW-NB15, η εφαρμογή υποστηρίζει φόρτωση ενός ενιαίου πολυταξικού μοντέλου, καθώς και του αντίστοιχου label encoder. Επιπλέον, προβλέπεται η φόρτωση εξειδικευμένων encoders για τα κατηγορικά χαρακτηριστικά proto, service και state, ώστε τα δεδομένα εισόδου να μετατρέπονται σωστά πριν από την πρόβλεψη.

Η διαδικασία φόρτωσης των μοντέλων πραγματοποιείται μέσω παραθύρων επιλογής αρχείων επιτρέποντας στον χρήστη να επιλέξει τα κατάλληλα pkl αρχεία τα οποία έχουν εξαχθεί από την διαδικασία της εκπαίδευσης. Μετά τη φόρτωση, τα μοντέλα αποθηκεύονται στις αντίστοιχες μεταβλητές της εφαρμογής και είναι διαθέσιμα για χρήση στη διαδικασία πρόβλεψης.

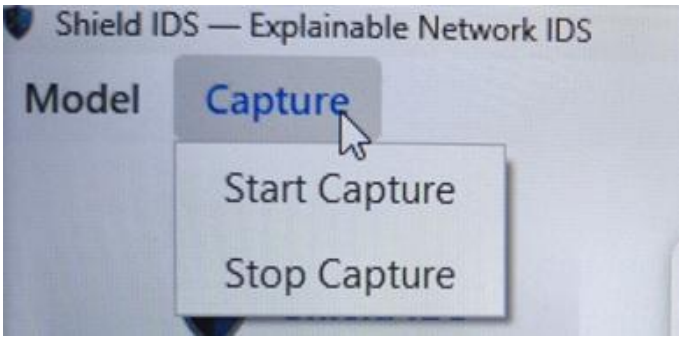
Το δεύτερο τμήμα του μενού αφορά τον έλεγχο της σύλληψης της δικτυακής κίνησης. Μέσα από το μενού Capture (φαίνεται στην Εικόνα 211), ο χρήστης μπορεί να ξεκινήσει ή να σταματήσει τη διαδικασία παρακολούθησης των ενεργών διεπαφών δικτύου. Κατά την έναρξη της σύλληψης, η εφαρμογή εντοπίζει τις διαθέσιμες δικτυακές διεπαφές και δημιουργεί νήμα σύλληψης για καθεμία από αυτές. Η υλοποίηση είναι πολυνηματική, ώστε

η εφαρμογή να μπορεί να παρακολουθεί ταυτόχρονα περισσότερες από μία διεπαφές και να συγκεντρώνει τα πακέτα σε κοινό μηχανισμό ομαδοποίησης ροών. Όταν ο χρήστης επιλέξει τη διακοπή της σύλληψης, η εφαρμογή τερματίζει τους ενεργούς μηχανισμούς καταγραφής και καθαρίζει τα προσωρινά δεδομένα των ροών ώστε να αποτραπεί η συνέχιση της επεξεργασίας πακέτων μετά το stop.

Τέλος, το μενού διαχειρίζεται δυναμικά την κατάσταση των εντολών του, ενεργοποιώντας ή απενεργοποιώντας τις αντίστοιχες επιλογές ώστε να αποφεύγεται η ταυτόχρονη ή λανθασμένη εκτέλεση αντικρουόμενων ενεργειών, όπως η πολλαπλή έναρξη της διαδικασίας capture όταν αυτή ήδη εκτελείται.



Εικόνα 210 Μενού φόρτωσης των μοντέλων



Εικόνα 211 Μενού εκκίνησης και παύσης της σύλληψης της δικτυακής κίνησης

7.2. Executive Dashboard

Ο πίνακας ελέγχου στελεχών για μια εφαρμογή IDS είναι ένα σημαντικό εργαλείο που μεταφράζει την τεχνική κατάσταση για την ασφάλεια σε χρήσιμες πληροφορίες. Αυτές οι πληροφορίες αξιοποιούνται από τη διοίκηση, τα επιχειρησιακά στελέχη και τους υπεύθυνους ασφαλείας σχετικά με την λήψη σημαντικών αποφάσεων. Καθώς ο όγκος δεδομένων είναι τεράστιος και οι επιθέσεις εξελίσσονται διαρκώς, μια IDS εφαρμογή είναι σημαντική όχι μόνο για την ανίχνευση εισβολών, αλλά και για την ικανότητά της να εξάγει γνώση που θα είναι ερμηνεύσιμη από τους χρήστες καθώς και να παρέχει άμεση αξιολόγηση κινδύνου.

Σύμφωνα με τη βιβλιογραφία για τους πίνακες ελέγχου κυβερνοασφάλειας (cybersecurity dashboards), τα συστήματα αυτά μετατρέπουν πολύπλοκα δεδομένα ασφάλειας σε κατανοητές οπτικοποιήσεις παρέχοντας συνολική εικόνα της κατάστασης και υποστήριξη στη λήψη αποφάσεων. Η προσέγγιση αυτή μπορεί να εφαρμοστεί και σε μια εφαρμογή IDS όπου το επίπεδο Executive Dashboard αποτελεί τη συνοπτική παρουσίαση υψηλού επιπέδου των πολύπλοκων δεδομένων που συγκεντρώνονται παρουσιάζοντας κρίσιμους δείκτες κατάστασης, εκτιμήσεις κινδύνου και συνολική εικόνα της ασφάλειας αντί για λεπτομερή τεχνικά συμβάντα. Έτσι επιτυγχάνεται συνεχής επίγνωση της κατάστασης του συστήματος και υποστήριξη έγκαιρης λήψης αποφάσεων από τα αρμόδια στελέχη συμβάλλοντας στην προληπτική αντιμετώπιση περιστατικών πριν αυτά εξελιχθούν σε σοβαρές επιθέσεις [44].

Επίσης, η αξιολόγηση της ασφάλειας δεν θα πρέπει να βασίζεται αποκλειστικά σε τεχνικούς και δυσνόητους δείκτες για τους απλούς χρήστες, αλλά απαιτεί συνδυασμό μετρικών κυβερνοασφάλειας και φυσικού αντίκτυπου (cyber-physical resilience), δηλαδή εκτίμηση του πραγματικού επιχειρησιακού αντίκτυπου μιας επίθεσης. Η ορατότητα στο δίκτυο αλλά και στις τερματικές συσκευές (endpoints) είναι κρίσιμη για την επίγνωση της κατάστασης και τη διατήρηση της λειτουργίας του συστήματος ακόμη και υπό συνθήκες επίθεσης [45].

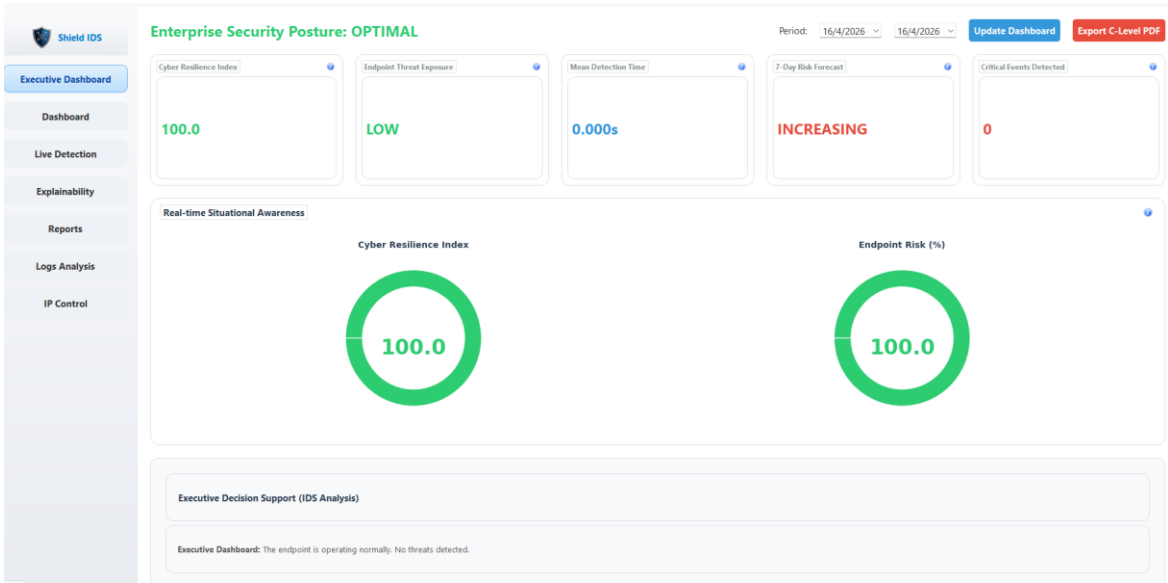
Το Executive Dashboard μιας εφαρμογής IDS παρουσιάζει τη συνολική κατάσταση ασφάλειας μέσω δεικτών κατανοητών από μη τεχνικούς χρήστες, αποτυπώνει τη σοβαρότητα και εξέλιξη των απειλών και υποστηρίζει τη λήψη αποφάσεων. Χωρίς αυτά, ένα IDS παραμένει απλώς ένα εργαλείο καταγραφής και όχι ένας μηχανισμός υποστήριξης αποφάσεων.

Σε πραγματικά περιβάλλοντα, μπορεί να παράγονται καθημερινά εκατοντάδες ή χιλιάδες συναγερμοί (alerts), όμως μόνο ένα μικρό ποσοστό είναι κρίσιμο για τη λειτουργία του οργανισμού. Η μετατροπή αυτών των δεδομένων σε συνοπτικούς δείκτες επιτρέπει στη διοίκηση να κατανοήσει άμεσα αν η κατάσταση είναι φυσιολογική, αν υπάρχει αυξημένος κίνδυνος ή αν απαιτείται άμεση παρέμβαση. Αυτή η πρακτική συνδέεται με τη λογική των δεικτών ανθεκτικότητας (resilience scores), όπου η ασφάλεια αποτυπώνεται ως αριθμητικός δείκτης που μειώνεται όσο αυξάνεται η επιτυχία των επιθέσεων [45].

Το Executive Dashboard μιας εφαρμογής IDS πρέπει να ευθυγραμμίζεται με τους επιχειρησιακούς στόχους του οργανισμού. Για τη διοίκηση, η κυβερνοασφάλεια δεν αντιμετωπίζεται ως ανεξάρτητος στόχος αλλά ως παράγοντας επιχειρησιακού ρίσκου και πιθανής οικονομικής ζημίας. Κατά συνέπεια, η αναφορά πρέπει να απαντά σε ερωτήματα όπως πόσο ασφαλής είναι η υποδομή, ή πόσο εκτεθειμένη είναι σε νέες απειλές. Έτσι το IDS μετατρέπεται από τεχνικό σύστημα σε σημαντικό εργαλείο διαχείρισης επιχειρησιακού κινδύνου.

7.2.1. Λειτουργίες φιλτραρίσματος και ενημέρωσης του Executive Dashboard

Στο Executive Dashboard (αρχική οθόνη του οποίου δείχνει η Εικόνα 212) υλοποιήθηκε και παρέχεται η δυνατότητα επιλογής χρονικού διαστήματος μέσω φίλτρου ημερομηνίας (από / έως), έτσι ώστε όλοι οι στρατηγικοί δείκτες και τα γραφήματα να υπολογίζονται με βάση την επιλεγμένη περίοδο αναφοράς.



Εικόνα 212 Οπτικοποίηση του Executive Dashboard

Εξαιτίας αυτής της λειτουργίας η απεικόνιση της κατάστασης ασφαλείας δεν περιορίζεται μόνο στην τρέχουσα χρονική στιγμή, αλλά μπορεί να προσαρμόζεται στις ανάγκες ανάλυσης του χρήστη. Το ίδιο χρονικό φίλτρο χρησιμοποιείται και κατά την εξαγωγή της εκτελεστικής αναφοράς σε μορφή PDF, ώστε το περιεχόμενο της αναφοράς να αντιστοιχεί πλήρως στην περίοδο που έχει επιλεγεί. Επίσης κατά την ενεργοποίηση της διαδικασίας ενημέρωσης, το κουμπί Update Dashboard αλλάζει προσωρινά ένδειξη σε κατάσταση ενημέρωσης και απενεργοποιείται για μικρό χρονικό διάστημα, ώστε να αποφεύγονται πολλαπλά πατήματα.

7.2.2. Στρατηγικοί δείκτες (KPIs)

Η σύγχρονη προσέγγιση στην ασφάλεια πληροφοριών επιβάλλει τη μετάβαση από την απλή ανίχνευση συμβάντων στη μέτρηση της ανθεκτικότητας του συστήματος. Στο Executive Dashboard η συνολική εικόνα ασφάλειας αποτυπώνεται μέσω στρατηγικών δεικτών απόδοσης (KPIs). Οι δείκτες αυτοί μετατρέπουν τα τεχνικά δεδομένα του IDS σε μετρήσιμες και συγκρίσιμες ποσότητες, επιτρέποντας την ποσοτική αξιολόγηση της ανθεκτικότητας και της έκθεσης της υποδομής.

Για τον σκοπό αυτό σχεδιάστηκε το Executive Dashboard το οποίο οπτικοποιεί κρίσιμους δείκτες (KPIs) όπως ο CRI, ο Endpoint Threat Exposure και ο MTTD, προσφέροντας μια άμεσα ερμηνεύσιμη εικόνα στη διοίκηση.

Ο δείκτης CRI εμφανίζει τη συνολική ανθεκτικότητα του συστήματος απέναντι σε κυβερνοαπειλές. Τιμές κοντά στο 100% δείχνουν ότι το περιβάλλον λειτουργεί δίχως κανένα πρόβλημα και με υψηλό επίπεδο ασφάλειας, ενώ χαμηλότερες τιμές υποδηλώνουν αυξημένη πίεση από κακόβουλη δραστηριότητα. Όσο λιγότερες απειλές ανιχνεύονται σε σχέση με το σύνολο των γεγονότων, τόσο πιο υγιές και σταθερό θεωρείται το σύστημα.

Ο δείκτης ορίζεται με τον μαθηματικό τύπο:

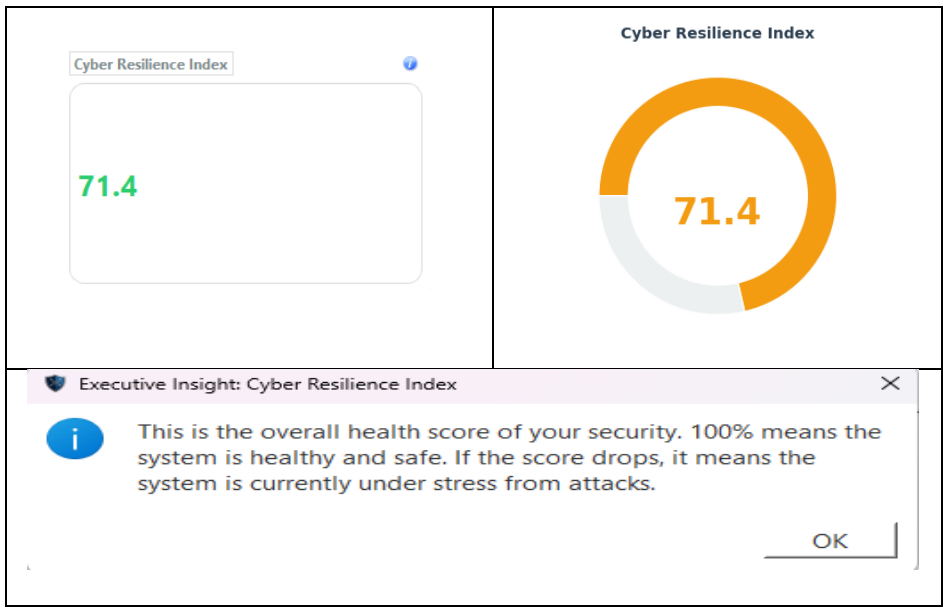
$$CRI = 100 \cdot (1 - \mathrm{N_m} / \mathrm{N}) \qquad (12)$$

όπου N_m είναι το πλήθος των κακόβουλων ροών που ανιχνεύονται σε συγκεκριμένο χρονικό διάστημα και N είναι το συνολικό πλήθος των ροών στο ίδιο χρονικό παράθυρο.

Αυτή η διατύπωση ακολουθεί τη γενική λογική της βιβλιογραφίας [45] περί cyber resilience, όπου η ανθεκτικότητα ορίζεται ως το συμπλήρωμα του ποσοστού επιτυχούς κακόβουλης δραστηριότητας σε σχέση με τη συνολική δραστηριότητα του συστήματος.

Στην παρούσα εργασία η έννοια αυτή αποτυπώνεται μέσω της αναλογίας κακόβουλων ροών προς το συνολικό πλήθος ροών.

Ο τύπος αυτός μετράει την κακόβουλη δραστηριότητα σε σχέση με τη συνολική λειτουργία του συστήματος. Όταν ο αριθμός των κακόβουλων ροών είναι μικρός σε σχέση με το συνολικό φορτίο κίνησης, ο λόγος $\frac{N_m}{N}$ προσεγγίζει το μηδέν και ο δείκτης ανθεκτικότητας προσεγγίζει το 100. Αντίθετα, όσο αυξάνεται το ποσοστό κακόβουλης δραστηριότητας, η τιμή του δείκτη υποχωρεί αναλογικά υποδηλώνοντας αυξημένη επιχειρησιακή πίεση και χαμηλότερο επίπεδο ανθεκτικότητας. Η τελική τιμή του δείκτη προβάλλεται με διπλή οπτικοποίηση (όπως δείχνει η Εικόνα 213) για μέγιστη σαφήνεια. Η πρώτη αφορά την ψηφιακή ένδειξη που επιτρέπει την ακριβή ανάγνωση του ποσοστού και η δεύτερη ένα κυκλικό όργανο μέτρησης. Όσον αφορά το δεύτερο γραφικό στοιχείο, παρέχεται χρωματική διαβάθμιση ανάλογα με το επίπεδο ανθεκτικότητας, διευκολύνοντας την άμεση κατανόηση της κατάστασης ασφαλείας από τη διοίκηση χωρίς την ανάγκη ανάλυσης των αριθμητικών δεδομένων. Επίσης ενσωματώνει ένα εικονίδιο πληροφοριών το οποίο λειτουργεί ως εργαλείο επεξήγησης (Εικόνα 213). Πατώντας το ο χρήστης έχει τη δυνατότητα να ενεργοποιήσει ένα αναδυόμενο παράθυρο το οποίο παρέχει πληροφορίες σχετικά με το συγκεκριμένο δείκτη.



Εικόνα 213 Οπτικοποίηση του δείκτη CRI στο Executive Dashboard με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου

Ο Endpoint Threat Exposure είναι ένας δείκτης που αποτυπώνει το επίπεδο κινδύνου του συστήματος με βάση τις κακόβουλες επιθέσεις που έχουν ανιχνευθεί. Ο δείκτης είναι μια απλοποιημένη παραλλαγή της προσέγγισης heatmap [44], όπου η «θερμότητα» εκφράζεται μέσω ενός αριθμητικού δείκτη που υπολογίζεται εσωτερικά αντί να κατανέμεται σε περιοχές. Η αριθμητική τιμή του δείκτη εμφανίζεται στον χρήστη μέσω του γραφήματος Gauge Chart με ονομασία Endpoint Risk και ανάλογα με την τιμή του, εμφανίζεται και η αντίστοιχη κατάσταση κινδύνου, η οποία αποδίδεται με συγκεκριμένο χρώμα (Εικόνα 214). Παράλληλα, η διεπαφή ενσωματώνει ένα εικονίδιο πληροφοριών που λειτουργεί ως εργαλείο επεξήγησης. Πατώντας το ο χρήστης έχει τη δυνατότητα να ενεργοποιήσει ένα αναδυόμενο παράθυρο με πληροφορίες σχετικά με το συγκεκριμένο δείκτη (Εικόνα 214).

Για τον υπολογισμό του δείκτη αρχικά καταμετρώνται οι κακόβουλες επιθέσεις ανά κατηγορία επίθεσης ενώ στην συνέχεια ομαδοποιούνται σε δύο επίπεδα. Σύμφωνα με την βιβλιογραφία όλες οι επιθέσεις δεν έχουν την ίδια βαρύτητα. Οι κρίσιμες επιθέσεις έχουν υψηλή βαρύτητα ενώ οι μη κρίσιμες επιθέσεις έχουν χαμηλή βαρύτητα [46], [47]. Η προσέγγιση αυτή ακολουθεί τη λογική της μεθόδου Entropy Weight, που επιτρέπει την ιεράρχηση των απειλών βάσει του αντικτύπου τους στο σύστημα [46]. Έτσι οι επιθέσεις όπως DoS, Exploits, Bot και Infiltration θεωρούνται υψηλής βαρύτητας ενώ οι επιθέσεις όπως PortScan, Fuzzers και Reconnaissance αντιμετωπίζονται ως ενέργειες αναγνώρισης, με μικρότερη βαρύτητα. Πιο συγκεκριμένα, οι επιθέσεις DoS, Exploits, Botnets, Infiltration, Backdoors, Worms, Shellcode και Web Attacks έχουν υψηλό συντελεστή βαρύτητας (w=2.5) γιατί απειλούν άμεσα την ακεραιότητα του συστήματος. Αντίθετα, οι επιθέσεις PortScan, Fuzzers, Reconnaissance, Analysis και Generic έχουν χαμηλότερο συντελεστή (w=0.2).

Ο μαθηματικός τύπος υπολογισμού είναι:

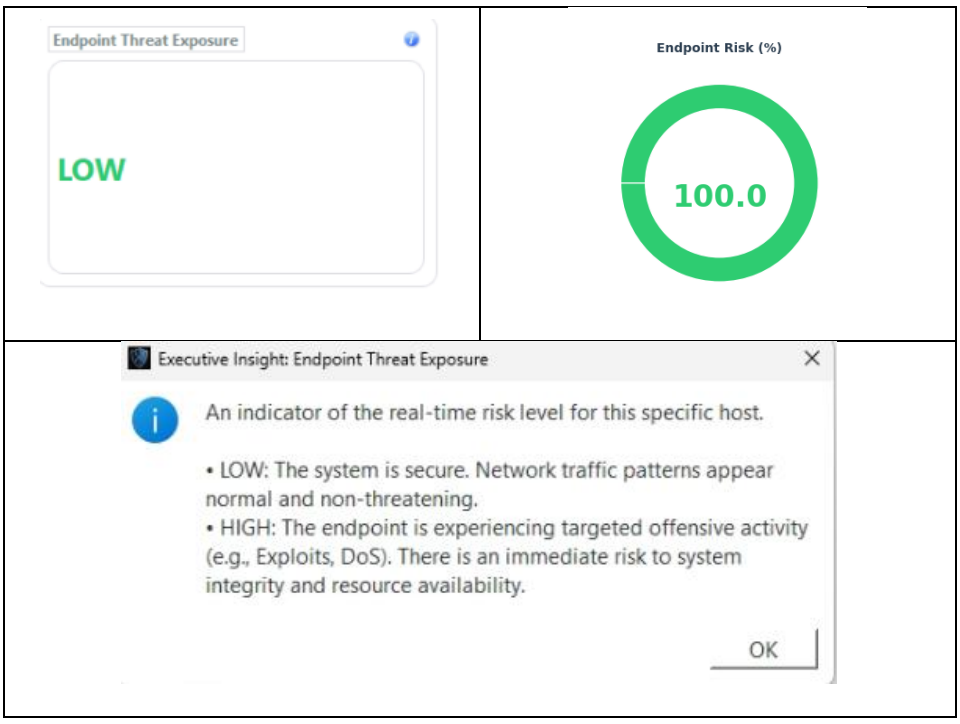
$$Score = 100 - \min(95, (2.5 * \mathrm{N1} + 0.2 * \mathrm{N2})) \quad (13)$$

όπου N1 είναι οι υψηλής βαρύτητας επιθέσεις ενώ με N2 είναι οι χαμηλής βαρύτητας επιθέσεις. Η χρήση της συνάρτησης min με τιμή ορίου 95 έχει τεθεί έτσι ώστε να μην υπάρχει αρνητική τιμή στον υπολογισμό που θα είχε ως αποτέλεσμα ο δείκτης να εμφανίσει error. Έτσι ο δείκτης διατηρείται εντός του εύρους 0–100.

Ο Πίνακας 11 παρουσιάζει τις καταστάσεις ανά τιμή score.

Σκορ	Κατάσταση	Περιγραφή
>40 & <100	ELEVATED HEAT	Το σύστημα βρίσκεται σε κίνδυνο λόγω μεγάλης κακόβουλης δραστηριότητας
<40	CRITICAL HEAT	Κρίσιμη κατάσταση, απαιτείται άμεση παρέμβαση.
100	LOW / OPTIMAL	Φυσιολογική λειτουργία, ελάχιστη ή μηδενική κακόβουλη δραστηριότητα

Πίνακας 11 Καταστάσεις ανά τιμή score



Εικόνα 214 Οπτικοποίηση του δείκτη Endpoint Threat Exposure με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου

Ο δείκτης MTTD αποτυπώνει τον μέσο χρόνο που μεσολαβεί από τη στιγμή που εμφανίζεται για πρώτη φορά μια κακόβουλη ροή στο δίκτυο μέχρι την στιγμή που αυτή αναγνωρίζεται από το IDS ως απειλή. Σε αυτόν τον δείκτη μικρότερες τιμές σημαίνουν

ταχύτερη ανίχνευση. Για τον τρόπο υπολογισμού του καταγράφεται για κάθε ροή η χρονική στιγμή εμφάνισης του πρώτου πακέτου (first_ts). Στη συνέχεια, όταν το σύστημα χαρακτηρίσει την συγκεκριμένη ροή ως κακόβουλη δραστηριότητα, τότε καταγράφεται η χρονική στιγμή της πρώτης ανίχνευσης (first_detect_ts). Η καθυστέρηση ανίχνευσης για τη ροή ορίζεται με τον μαθηματικό τύπο:

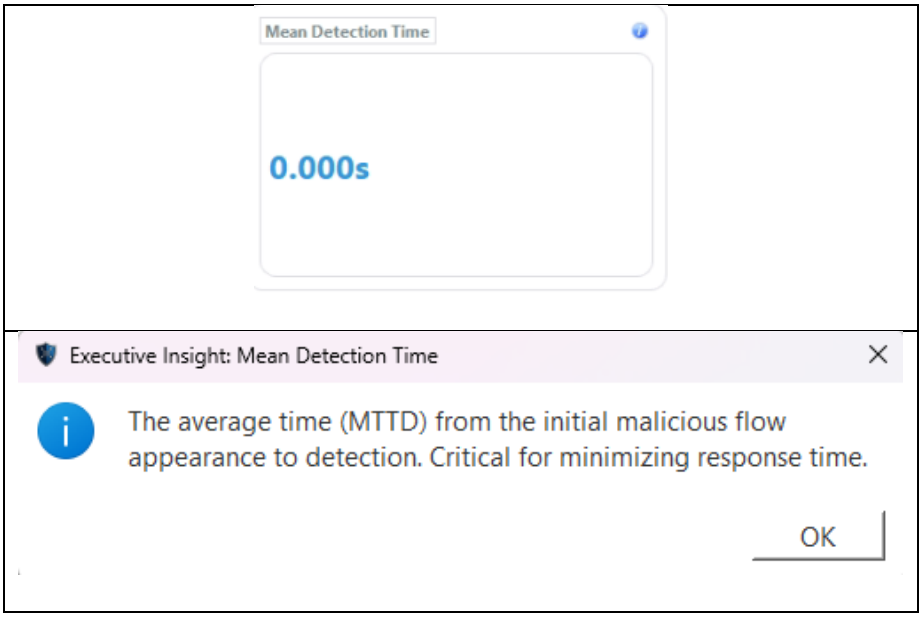
$$\text{Detection Delay} = \text{first\_detect\_ts} - \text{first\_ts}$$

Το MTTD είναι ο μέσος όρος αυτών των διαφορών για όλες τις ανιχνευμένες επιθέσεις στην επιλεγμένη χρονική περίοδο και ορίζεται με τον μαθηματικό τύπο:

$$MTTD = \left(\frac{1}{N}\right) * \sum_{i=1}^{N} detection\_delay_i \quad (14)$$

όπου N είναι ο αριθμός των ανιχνευμένων επιθέσεων στην εξεταζόμενη χρονική περίοδο και $detection\_delay_i$ η καθυστέρηση ανίχνευσης της i-οστής επίθεσης.

Η τιμή του MTTD προβάλλεται εκφρασμένη σε δευτερόλεπτα με ακρίβεια χιλιοστού, επιτρέποντας στον χρήστη να αξιολογήσει με μεγάλη λεπτομέρεια την επιχειρησιακή ταχύτητα του IDS. Επίσης η κάρτα περιλαμβάνει επίσης εικονίδιο πληροφοριών με σύντομη επεξήγηση της σημασίας του δείκτη (αποτυπώνεται στην Εικόνα 215).



Εικόνα 215 Οπτικοποίηση δείκτη MTTD με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου

Ο 7-Day Risk Forecast είναι ένας δείκτης έγκαιρης προειδοποίησης ο οποίος επιτρέπει στον χρήστη να κατανοήσει αν οι επιθέσεις στο σύστημα κινούνται σε μια αυξητική τάση ή εάν

παραμένουν σταθερές (πρόβλεψη τάσης των επιθέσεων). Ο 7-Day Risk Forecast αναλύει τη δραστηριότητα των τελευταίων 24 ωρών σε σύγκριση με τον μέσο όρο της εβδομάδας. Πιο αναλυτικά, αρχικά ορίζεται ένα παράθυρο 24 ωρών πριν από την επιλεγμένη τελική ημερομηνία της ανάλυσης από την τρέχουσα χρονική στιγμή και υπολογίζεται το πλήθος των συμβάντων που καταγράφηκαν μέσα σε αυτό. Στη συνέχεια, λαμβάνεται το συνολικό πλήθος των κακόβουλων συμβάντων του αντίστοιχου επταήμερου παραθύρου αναφοράς και διαιρείται με το επτά, ώστε να εξαχθεί ο μέσος ημερήσιος αριθμός απειλών. Εάν η πρόσφατη δραστηριότητα κινείται στα επίπεδα του μέσου όρου ή είναι μηδενική τότε η κατάσταση χαρακτηρίζεται σταθερή (STABLE). Εάν όμως η δραστηριότητα του τελευταίου 24ώρου υπερβαίνει τον ημερήσιο μέσο όρο της εβδομάδας τότε το σύστημα προειδοποιεί για αυξημένο κίνδυνο (εμφανίζει την ένδειξη INCREASING).

Ο δείκτης προβάλλεται στην διεπαφή σε κάρτα με χρωματική ένδειξη (Εικόνα 216). Η κατάσταση "STABLE" αποδίδεται με μωβ χρώμα, ενώ η προειδοποίηση "INCREASING" εμφανίζεται με κόκκινο, ώστε να προκαλεί την άμεση προσοχή του τεχνικού ασφαλείας. Επιπλέον, η κάρτα ενσωματώνει εικονίδιο πληροφοριών το οποίο παρέχει στον χρήστη μια σύντομη επεξήγηση της μεθοδολογίας πρόβλεψης και της σημασίας της (Εικόνα 216).

Ο δείκτης Critical Events Detected εμφανίζει τον αριθμό των κακόβουλων συμβάντων που εντοπίστηκαν από το IDS σε επιλεγμένο χρονικό διάστημα έτσι ώστε ο χρήστης να αποκτά άμεση εικόνα της έντασης των επιθέσεων.



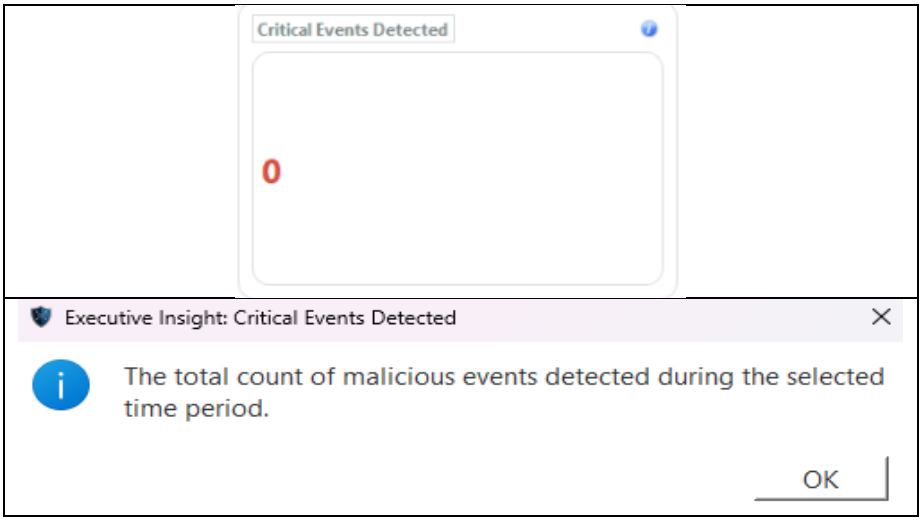
Εικόνα 216 Οπτικοποίηση του δείκτη 7-Day Risk Forecast με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου

fCIC-IDS2017 και UNSW-NB15. Συγκεκριμένα, το σύστημα εντοπίζει τις επιθέσεις για τις οποίες έχουν εκπαιδευτεί τα μοντέλα, χρησιμοποιεί την μέθοδο μείωσης ψευδών θετικών αποτελεσμάτων (FPR), δηλαδή εξαιρεί έμπιστες πηγές κίνησης μέσω λευκής λίστας

(whitelist), που θα αναλύσουμε παρακάτω στην υπό ενότητα 8.8.2 και εμφανίζει τις πραγματικές επιθέσεις. Έτσι παρέχει μεγαλύτερη αξιοπιστία προς τον χρήστη.

Ο δείκτης απεικονίζεται στο Executive Dashboard με ευδιάκριτη αριθμητική ένδειξη (όπως δείχνει η Εικόνα 217) για την άμεση ενημέρωση του τεχνικού ασφαλείας. Επίσης περιλαμβάνει εικονίδιο πληροφοριών για την παροχή λεπτομερειών σχετικά με τον συνολικό αριθμό των σοβαρών απειλών που απαιτούν άμεση χειροκίνητη αξιολόγηση και παρέμβαση (παρουσιάζεται επίσης στην Εικόνα 217).

Συνοψίζοντας, οι παραπάνω στρατηγικοί δείκτες προσφέρουν μέσω της οπτικοποίησης των δεδομένων μια συνολική εικόνα της ασφάλειας σε στελέχη και μη τεχνικούς χρήστες. Αυτή η οπτικοποίηση στο Executive Dashboard μετατρέπει τις τεχνικές πληροφορίες του IDS σε γνώση επιτρέποντας στη διοίκηση να λάβει τις σωστές αποφάσεις.



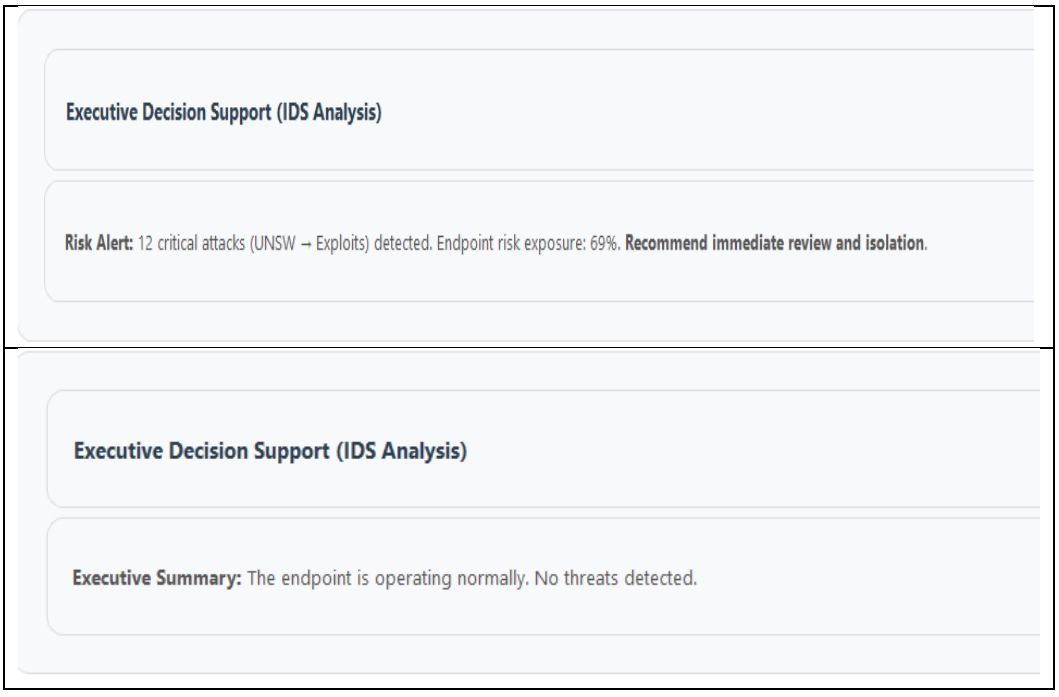
Εικόνα 217 Οπτικοποίηση των σημαντικών συμβάντων με ενεργοποίηση αναδυόμενου πληροφοριακού παραθύρου

7.2.3. Μοντέλο λήψης αποφάσεων: Παροχή προτάσεων βάσει της σοβαρότητας των επιθέσεων

Το Executive Dashboard ενσωματώνει το μοντέλο Υποστήριξης Λήψης Αποφάσεων (Decision Support). Σκοπός του είναι να παρουσιάζει κατανοητές προτάσεις δράσης και όχι τεχνικούς δείκτες στον χρήστη ή τα στελέχη της διοίκησης [48]. Ουσιαστικά, προσφέρει πληροφορία έτοιμη για αξιοποίηση (Actionable Intelligence), παρέχοντας ξεκάθαρη εικόνα του επιπέδου κινδύνου και υποστηρίζοντας τη λήψη αποφάσεων [44].

Το μοντέλο λήψης αποφάσεων βασίζεται στον συνδυασμό πολλαπλών δεικτών ασφάλειας που προκύπτουν από την ανάλυση της δικτυακής δραστηριότητας. Τα δεδομένα επεξεργάζονται ώστε να εκτιμηθεί το επίπεδο απειλής, η σοβαρότητα των περιστατικών και η επίπτωση στο σύστημα. Με βάση αυτή την εικόνα κινδύνου, το σύστημα παράγει συμπεράσματα για την κατάσταση και διαμορφώνει κατάλληλες προτάσεις δράσης.

Για να εμφανιστεί στον χρήστη η κατάλληλη ενέργεια αρχικά πραγματοποιείται έλεγχος εάν υπάρχει κακόβουλη κίνηση. Εάν δεν εντοπιστούν απειλές, το σύστημα εμφανίζει ένα μήνυμα ομαλής λειτουργίας (Εικόνα 218). Εάν όμως ο αριθμός των απειλών είναι μεγαλύτερος του



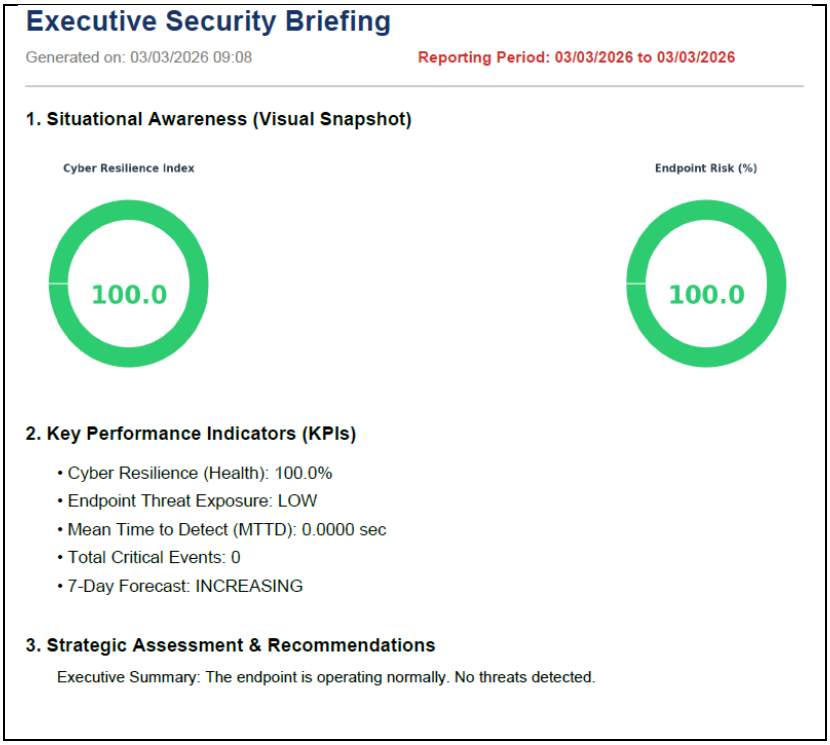
Εικόνα 218 Οπτικοποίηση των δράσεων

μηδενός ενεργοποιείται το σενάριο ειδοποίησης (βλέπε Εικόνα 218) το οποίο συνδυάζει δεδομένα από τις επιθέσεις που βρέθηκαν, τον πιο συχνό τύπο επίθεσης καθώς και πόσο έχει μειωθεί η ακεραιότητα του endpoint.

7.2.4. Δημιουργία Εκτελεστικής Αναφοράς Ασφαλείας (Executive Security Report)

Σε ένα Σύστημα Ανίχνευσης Εισβολών εκτός από την οπτικοποίηση των δεδομένων μέσω στρατηγικών δεικτών και γραφημάτων απαιτείται και η εξαγωγή αναφορών. Οι αναφορές αυτές αποτυπώνουν την τρέχουσα κατάσταση κινδύνου και τους βασικούς δείκτες απόδοσης. Για τον σκοπό αυτό συμπεριλήφθηκε στην εφαρμογή η εξαγωγή ολοκληρωμένης αναφοράς σε μορφή PDF έτσι ώστε τα αποτελέσματα να είναι διαθέσιμα στην διοίκηση και τα στελέχη ασφάλειας

Πατώντας ο χρήστης την σχετική λειτουργία από το Executive Dashboard η εφαρμογή δημιουργεί ένα έγγραφο μεγέθους A4 (Εικόνα 219), το οποίο περιέχει τις κρίσιμες πληροφορίες από το Executive Dashboard για το χρονικό διάστημα που έχει επιλέξει ο χρήστης μέσω του φίλτρου ημερομηνίας. Αρχικά, δημιουργείται η επικεφαλίδα της αναφοράς η οποία περιλαμβάνει τον τίτλο του εγγράφου, την ημερομηνία και ώρα δημιουργίας, όπως και το χρονικό διάστημα στο οποίο αναφέρεται. Στην συνέχεια περιλαμβάνονται τα βασικά γραφήματα που είναι το CRI καθώς και το Endpoint Risk.



Εικόνα 219 Εκτελεστική Αναφορά Ασφαλείας σε μορφή PDF

Εκτός από τα κεντρικά γραφήματα η αναφορά ενσωματώνει όλους τους στρατηγικούς δείκτες συμπεριλαμβανομένου του MTTD και του 7-Day Risk Forecast διασφαλίζοντας με αυτό το τρόπο ότι η διοίκηση έχει πρόσβαση στο σύνολο της πληροφορίας που παράγει το σύστημα.

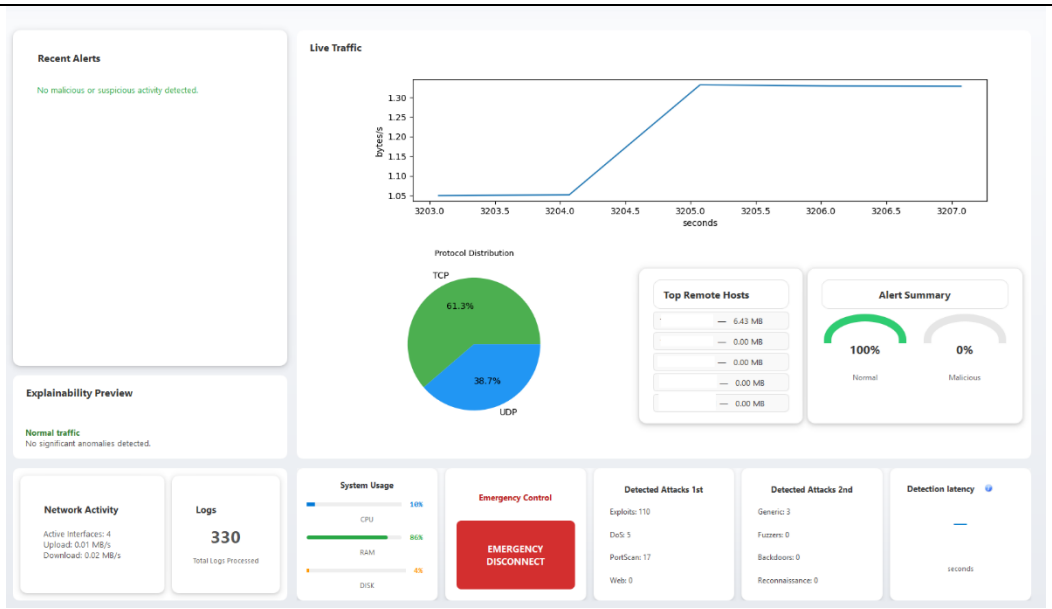
Τέλος η αναφορά περιλαμβάνει την στρατηγική αξιολόγηση και τις συστάσεις (Strategic Assessment & Recommendations), οι οποίες βασίζονται στα αποτελέσματα της ανάλυσης. Συνολικά, η αναφορά είναι διαμορφωμένη ώστε να είναι ευανάγνωστη και κατανοητή από τη διοίκηση και τα στελέχη ασφάλειας.

7.3. Κεντρικό dashboard

Η κεντρική καρτέλα (Dashboard) του IDS (αρχική οθόνη της οποίας αποτυπώνεται στην Εικόνα 220) αποτελεί την πιο σημαντική λειτουργία καθώς είναι η βασική εικόνα του συστήματος. Αποτελείται από τα πιο κρίσιμα στοιχεία που χρειάζεται ο χρήστης για να αξιολογήσει άμεσα την ασφάλεια και τη συμπεριφορά της δικτυακής κίνησης. Σχεδιάστηκε έτσι ώστε να μετατρέπει μεγάλους όγκους ροών δικτύου σε χρήσιμες ενδείξεις, μειώνοντας τον χρόνο που απαιτείται για να εντοπιστεί τυχόν κακόβουλη κίνηση. Οι κάρτες του dashboard ενημερώνονται δυναμικά και παρουσιάζουν την πληροφορία με ιεραρχημένο τρόπο. Πρώτα εμφανίζεται η συνολική εικόνα μετά τα σημεία που απαιτούν προσοχή και αφορούν την ασφάλεια και τέλος τα αναλυτικά στοιχεία.

Το Dashboard οπτικοποιεί την εξέλιξη της δικτυακής κίνησης σε πραγματικό χρόνο ώστε να γίνονται αντιληπτές οι αλλαγές της επικοινωνίας. Επίσης εμφανίζεται η κατανομή των πρωτοκόλλων μέσω διαγράμματος και υπάρχει και η κάρτα “Top Remote Hosts”, η οποία εμφανίζει τους πιο ενεργούς απομακρυσμένους κόμβους επικοινωνίας, διευκολύνοντας τον εντοπισμό ασυνήθιστης δικτυακής δραστηριότητας.

Σχετικά με την ασφάλεια παρέχεται μια συνοπτική εικόνα μέσω δεικτών που ξεχωρίζουν την κανονική από την κακόβουλη δραστηριότητα, ενώ υπάρχει και κάρτα με τα πιο πρόσφατα συμβάντα (Recent Alerts), ώστε ο χρήστης να βλέπει άμεσα τι ανιχνεύθηκε, πότε συνέβη και από ποια διεύθυνση προήλθε. Επίσης το Dashboard περιλαμβάνει προεπισκόπηση επεξήγησης (Explainability Preview), η οποία εμφανίζει τους βασικούς λόγους για τους οποίους μια ροή χαρακτηρίστηκε κακόβουλη και κάρτες στατιστικών που παρουσιάζουν τον αριθμό των επιθέσεων που έχουν ανιχνευθεί ανά κατηγορία.



Εικόνα 220 Κεντρικό Ταμπλό (Dashboard)

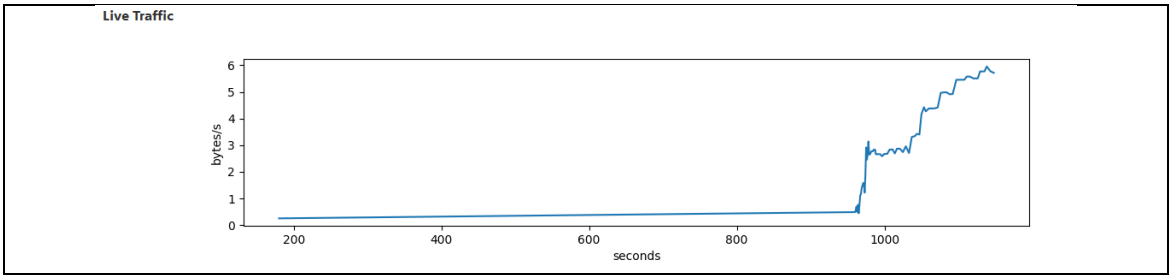
Οι κατηγορίες αυτές περιλαμβάνουν επιθέσεις όπως Exploits, DoS, Port Scanning, Web attacks, καθώς και άλλες μορφές επιθέσεων όπως Generic, Fuzzers, Backdoors και Reconnaissance. Επιπλέον, εμφανίζεται η ένδειξη Detection Latency, η οποία παρουσιάζει τον χρόνο που απαιτείται από τη στιγμή που παρατηρείται μια ύποπτη ροή μέχρι τη στιγμή που το σύστημα την χαρακτηρίζει ως κακόβουλη. Η πληροφορία αυτή είναι ιδιαίτερα σημαντική για την αξιολόγηση της αποτελεσματικότητας του συστήματος.

Στο Dashboard ενσωματώνονται δείκτες μέτρησης χρήσης πόρων (CPU, μνήμη, δίσκος) και μετρητές καταγραφής (logs), ώστε να γίνεται σαφές αν το σύστημα λειτουργεί ομαλά και καταγράφει επαρκώς τα γεγονότα. Επίσης υπάρχει μηχανισμός άμεσης διακοπής του δικτύου (emergency control) έτσι ώστε όταν ο κίνδυνος είναι αυξημένος να υπάρχει άμεση αντίδραση από τους τεχνικούς.

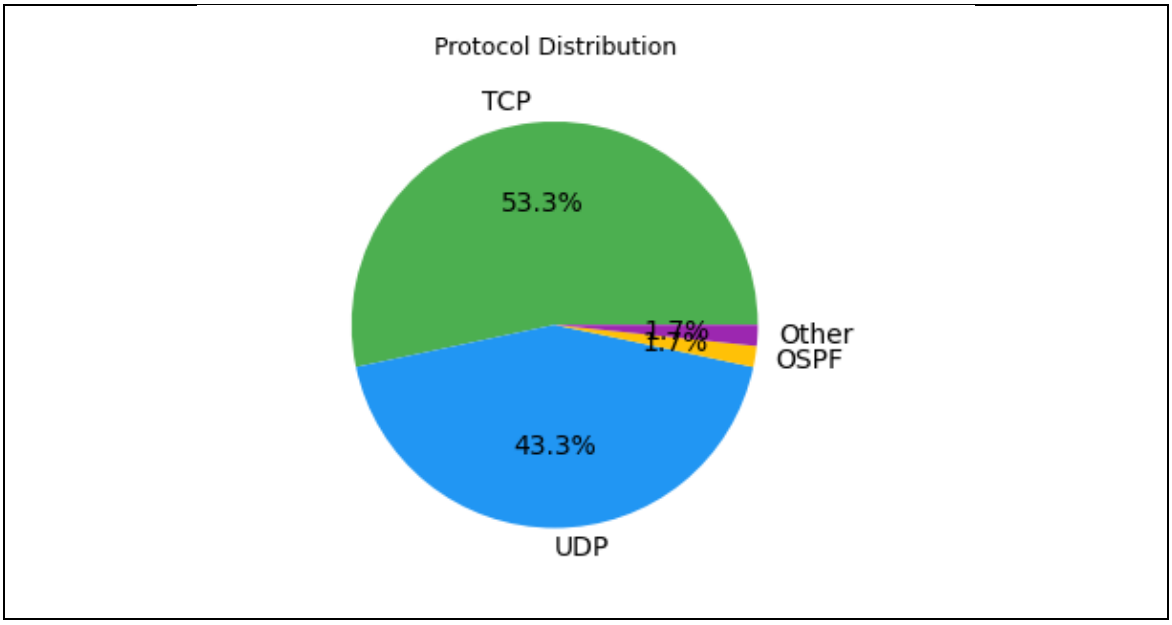
Πιο αναλυτικά, όσον αφορά το γράφημα που εμφανίζει την ζωντανή ροή της δικτυακής κίνησης (όπως δείχνει η Εικόνα 221), στον οριζόντιο άξονα παρουσιάζεται ο χρόνος, ενώ στον κατακόρυφο άξονα εμφανίζεται ο ρυθμός μεταφοράς δεδομένων σε bytes ανά δευτερόλεπτο. Το γράφημα ενημερώνεται δυναμικά καθώς το σύστημα αναλύει νέες ροές δικτύου. Έτσι ο τεχνικός ασφαλείας μπορεί να εντοπίσει απότομες αυξήσεις οι οποίες ενδέχεται να σχετίζονται με κακόβουλη δραστηριότητα.

Το διάγραμμα πίτας (pie chart) με τίτλο Protocol Distribution (βλέπε Εικόνα 222), παρουσιάζει την κατανομή των πρωτοκόλλων που χρησιμοποιούνται στη δικτυακή κίνηση.

Το διάγραμμα απεικονίζει το ποσοστό συμμετοχής κάθε πρωτοκόλλου στο σύνολο της κίνησης, όπως για παράδειγμα TCP, UDP, OSPF ενώ τα λιγότερο συχνά ή μη αναγνωρισμένα πρωτόκολλα ομαδοποιούνται στη κατηγορία Other ώστε το διάγραμμα να παραμένει ευανάγνωστο. Αυτό αφορά μόνο το διάγραμμα, καθώς στις καρτέλες Logs Analysis και Alerts & Reports, τα πρωτόκολλα προβάλλονται μέσω των αντίστοιχων εγγραφών της δικτυακής ροής με πιο αναλυτικό τρόπο. Αυτή η οπτικοποίηση επιτρέπει στον τεχνικό ασφαλείας να αποκτήσει μια γρήγορη εικόνα της δικτυακής επικοινωνίας. Αυτή η οπτικοποίηση επιτρέπει στον τεχνικό ασφαλείας να αποκτήσει μια γρήγορη εικόνα της δικτυακής επικοινωνίας. Επίσης από μια σημαντική μεταβολή στην κατανομή της κίνησης μπορεί να καταλάβει εάν αφορά ύποπτη δραστηριότητα ή κακόβουλη κίνηση στο δίκτυο.



Εικόνα 221 Οπτικοποίηση ζωντανής ροής στο Dashboard



Εικόνα 222 Διάγραμμα πίτας για την κατανομή των πρωτοκόλλων

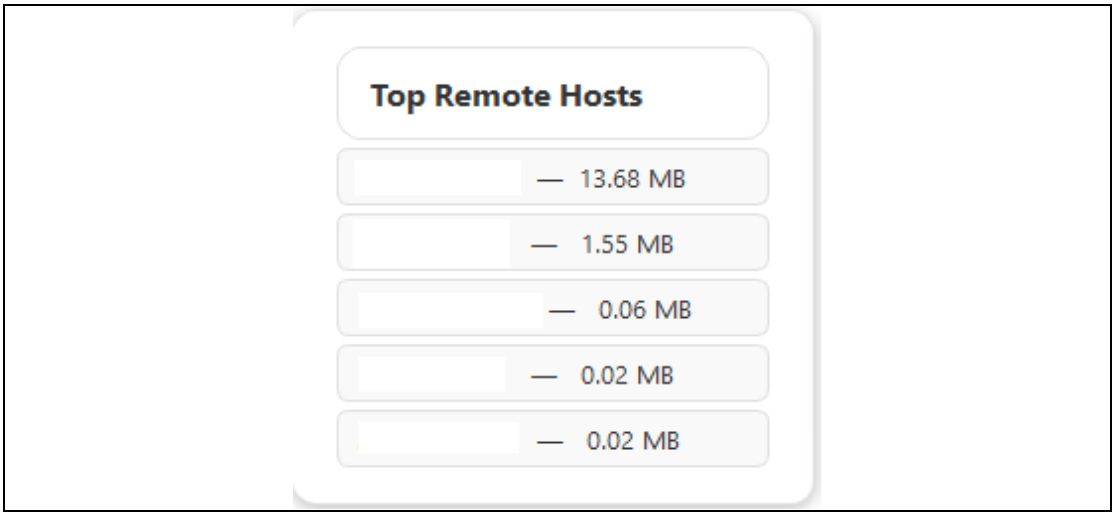
Η κάρτα Top Remote Hosts (φαίνεται στην Εικόνα 223) εμφανίζει τις πέντε πιο δραστήριες απομακρυσμένες διευθύνσεις IP που επικοινωνούν με το σύστημα κατά τη διάρκεια της δικτυακής κίνησης. Για κάθε απομακρυσμένο κόμβο παρουσιάζεται η διεύθυνση IP καθώς και ο συνολικός όγκος δεδομένων που έχει ανταλλαχθεί. Έτσι ο χρήστης μπορεί να εντοπίσει εύκολα ποιες εξωτερικές διευθύνσεις εμφανίζουν αυξημένη δραστηριότητα και να κατανοήσει πιθανές ανωμαλίες στη συμπεριφορά του δικτύου.

Η κάρτα επιτρέπει την καλύτερη κατανόηση της εξερχόμενης και εισερχόμενης επικοινωνίας του δικτύου και μπορεί να βοηθήσει στον εντοπισμό ύποπτων συνδέσεων, όπως επικοινωνία με άγνωστους εξωτερικούς διακομιστές ή ασυνήθιστα υψηλή ανταλλαγή δεδομένων με συγκεκριμένες διευθύνσεις IP.

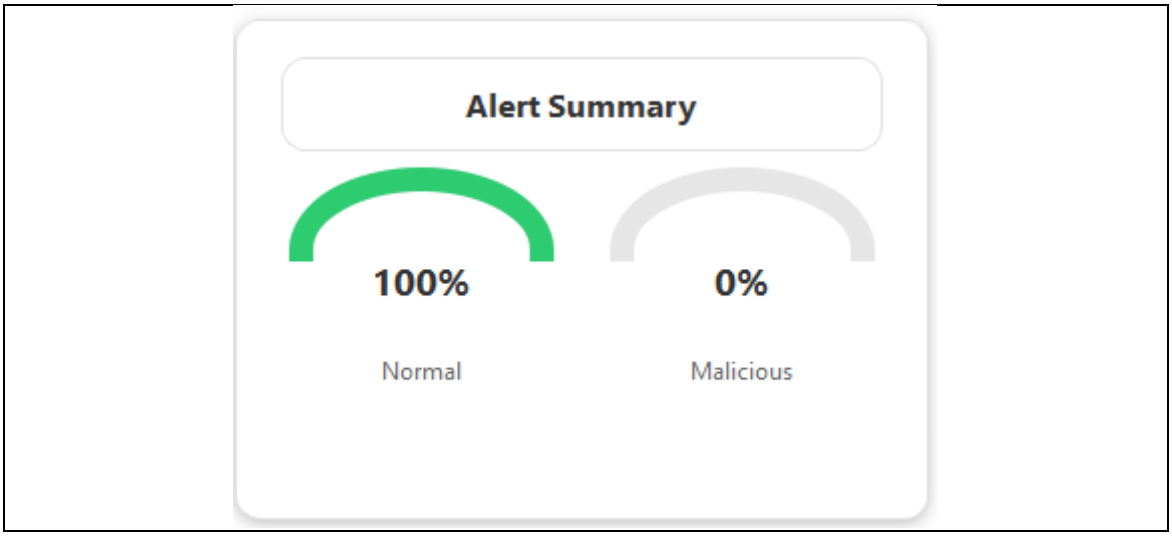
Η κάρτα Alert Summary (Εικόνα 224) παρουσιάζει μια συνοπτική εικόνα της κατάστασης ασφαλείας του δικτύου εμφανίζοντας το ποσοστό της κανονικής και της κακόβουλης δικτυακής κίνησης που έχει εντοπιστεί από το σύστημα. Η απεικόνιση γίνεται μέσω δύο ημικυκλικών δεικτών (gauges), οι οποίοι δείχνουν την αναλογία των ροών που έχουν χαρακτηριστεί ως κανονική (Normal) και ως κακόβουλη (Malicious).

Οι τιμές των δεικτών ενημερώνονται δυναμικά κατά τη διάρκεια της λειτουργίας του συστήματος, προσφέροντας άμεση εικόνα της συνολικής κατάστασης του δικτύου κι επιτρέποντας γρήγορη αξιολόγηση της συμπεριφοράς της δικτυακής κίνησης και διευκολύνει τον εντοπισμό περιπτώσεων όπου παρατηρείται αυξημένη κακόβουλη δραστηριότητα.

Η κάρτα Recent Alerts (Εικόνα 225) οπτικοποιεί τα πιο πρόσφατα συμβάντα ασφάλειας που έχουν εντοπιστεί. Στόχος της είναι να ενημερώνει τον χρήστη σε πραγματικό χρόνο για τις τελευταίες κακόβουλες δραστηριότητες στο δίκτυο. Η πληροφορία που εμφανίζεται προκύπτει από την ανάλυση των δικτυακών ροών, οι οποίες ταξινομούνται από τα μοντέλα



Εικόνα 223 Κάρτα Top Remote Hosts (Για λόγους προσωπικών δεδομένων οι IPs έχουν διαγραφεί)



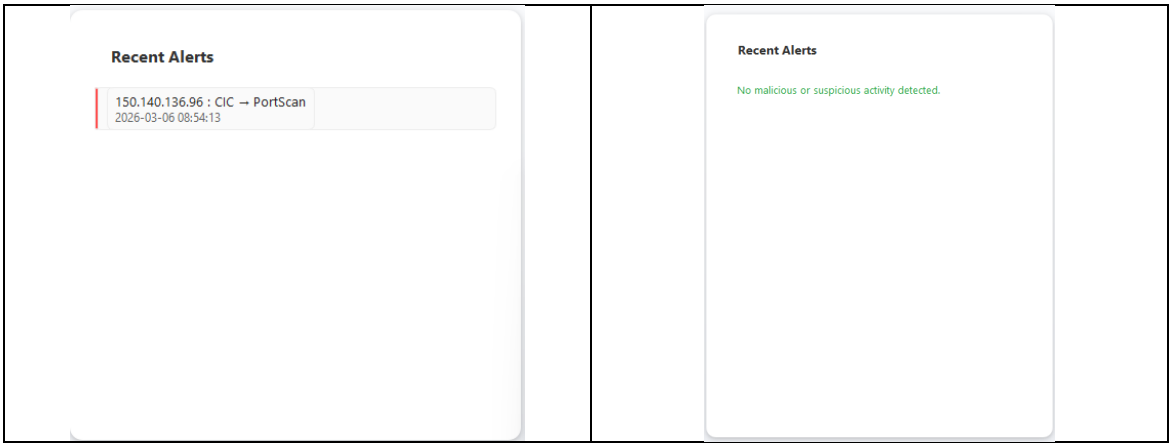
Εικόνα 224 Κάρτα Alert Summary

ανίχνευσης και επισημαίνονται ως κανονικές ή κακόβουλες. Όταν μια ροή χαρακτηριστεί ως κακόβουλη, το σύστημα καταγράφει το συμβάν και το εμφανίζει στη συγκεκριμένη κάρτα σε μια εγγραφή ως πρόσφατο alert.

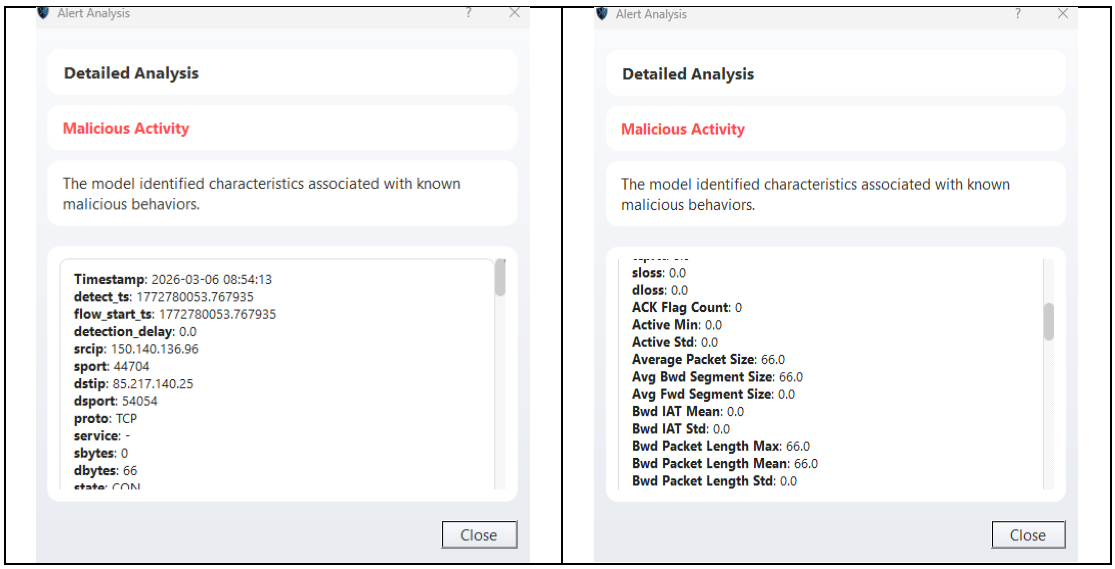
Κάθε εγγραφή περιλαμβάνει βασικές πληροφορίες σχετικά με την κακόβουλη ροή, όπως τη χρονική στιγμή καταγραφής, την διεύθυνση IP προέλευσης καθώς και τον τύπο της επίθεσης που εντοπίστηκε. Έτσι, ο χρήστης μπορεί να αποκτήσει άμεσα μια πρώτη εικόνα για το περιστατικό και να κατανοήσει τα βασικά χαρακτηριστικά της ανιχνευμένης δραστηριότητας. Σε περιπτώσεις όπου δεν έχουν εντοπιστεί κακόβουλες ροές εμφανίζεται ενημερωτικό μήνυμα που υποδεικνύει ότι δεν υπάρχει ενεργή απειλή για τη συγκεκριμένη χρονική στιγμή.

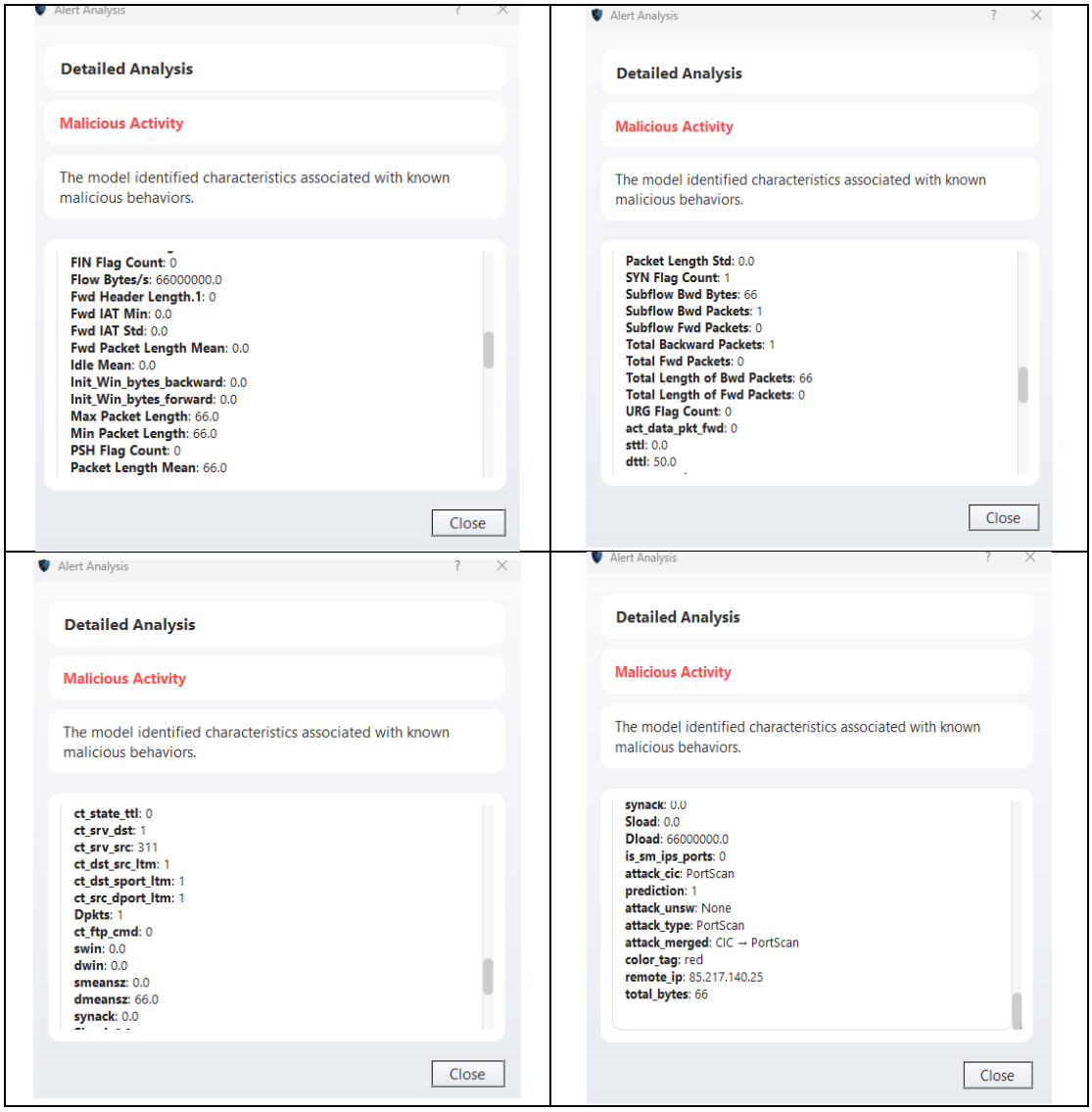
Επιπλέον, το σύστημα παρέχει τη δυνατότητα περαιτέρω διερεύνησης των ανιχνευμένων συμβάντων. Όταν ο χρήστης επιλέξει μια εγγραφή που αντιστοιχεί σε κακόβουλη δραστηριότητα, εμφανίζεται ένα αναδυόμενο παράθυρο με αναλυτικές πληροφορίες (Εικόνα 226). Στο παράθυρο παρουσιάζονται οι διευθύνσεις IP προέλευσης και προορισμού, ο χρόνος καταγραφής οι πόρτες προέλευσης και προορισμού καθώς και οι τιμές των χαρακτηριστικών που οδήγησαν το σύστημα στον χαρακτηρισμό της ροής ως κακόβουλης.

Έτσι παρέχεται στον τεχνικό ασφαλείας μια πιο κατανοητή εικόνα της απόφασης του συστήματος ανίχνευσης, ενισχύοντας παράλληλα τη διαφάνεια της διαδικασίας ανίχνευσης.



Εικόνα 225 Αριστερά: Παρουσίαση του πιο πρόσφατου συμβάντος κακόβουλης δραστηριότητας στην κάρτα Recent Alerts, Δεξιά: Ενημέρωση ότι δεν υπάρχει ενεργή απειλή τη συγκεκριμένη χρονική στιγμή





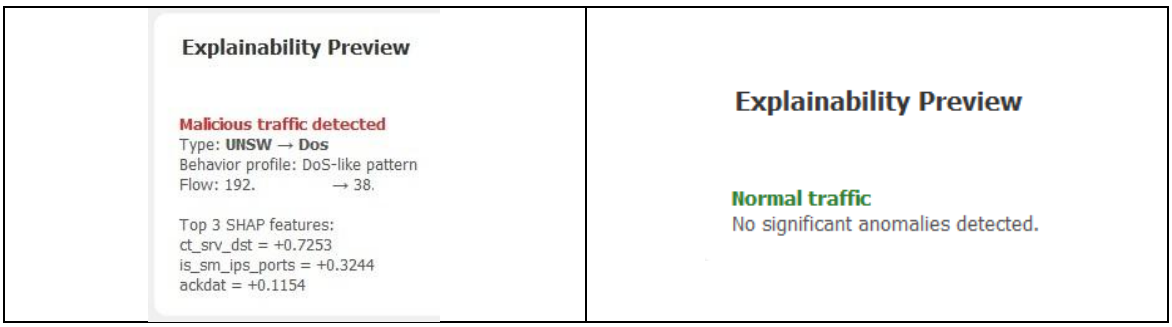
Εικόνα 226 Αναδυόμενο παράθυρο με αναλυτικές πληροφορίες για την κακόβουλη δραστηριότητα

Η κάρτα Explainability Preview (όπως δείχνει η Εικόνα 227) παρέχει μια σύντομη επεξήγηση της απόφασης του συστήματος για την ταξινόμηση της δικτυακής δραστηριότητας. Όταν η κίνηση του δικτύου χαρακτηρίζεται ως φυσιολογική, εμφανίζεται σχετικό μήνυμα που υποδεικνύει ότι δεν εντοπίστηκαν σημαντικές ανωμαλίες στη συμπεριφορά της ροής. Σε περιπτώσεις όπου εντοπίζεται κακόβουλη δραστηριότητα η κάρτα εμφανίζει μια σύντομη περιγραφή των βασικών παραγόντων που συνέβαλαν στην απόφαση του συστήματος ενισχύοντας με αυτό το τρόπο την διαφάνεια στην διαδικασία ανίχνευσης. Επίσης, λειτουργεί ως μια προεπισκόπηση της πιο αναλυτικής σελίδας Explainability Analysis, επιτρέποντας στον χρήστη να αποκτήσει μια πρώτη εικόνα πριν προχωρήσει σε λεπτομερέστερη διερεύνηση του περιστατικού.

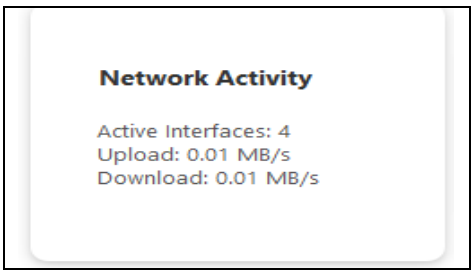
Η περιγραφή αυτή περιλαμβάνει πληροφορίες σχετικά με τον τύπο της επίθεσης που ανιχνεύθηκε, τις εμπλεκόμενες διευθύνσεις IP καθώς και ορισμένα βασικά χαρακτηριστικά (features) της ροής που ενεργοποίησαν την ανίχνευση. Επιπλέον παρουσιάζεται συνοπτικά το προφίλ συμπεριφοράς της ύποπτης δραστηριότητας, δηλαδή το είδος του μοτίβου που αναγνώρισε το σύστημα, όπως για παράδειγμα επιθέσεις τύπου DoS, PortScan, Web Attack ή άλλες κατηγορίες ανώμαλης συμπεριφοράς. Έτσι ο χρήστης μπορεί να κατανοήσει ποια στοιχεία της δικτυακής δραστηριότητας θεωρήθηκαν ύποπτα και οδήγησαν το σύστημα στον χαρακτηρισμό της ροής ως κακόβουλης. Επίσης τα σημαντικότερα χαρακτηριστικά που επηρέασαν την τελική απόφαση του μοντέλου προβάλλονται είτε μέσω της τεχνικής SHAP είτε, όταν αυτή δεν είναι διαθέσιμη, μέσω βασικών δεικτών συμπεριφοράς της ροής.

Η κάρτα Network Activity (φαίνεται στην Εικόνα 228) παρέχει πληροφορίες σχετικά με τη δικτυακή δραστηριότητα του συστήματος σε πραγματικό χρόνο. Συγκεκριμένα, η κάρτα εμφανίζει τον αριθμό των ενεργών δικτυακών διεπαφών (Active Interfaces), καθώς και τον τρέχοντα ρυθμό αποστολής (Upload) και λήψης (Download) δεδομένων. Οι τιμές αυτές ενημερώνονται δυναμικά κατά τη διάρκεια λειτουργίας του συστήματος.

Η πληροφορία αυτή επιτρέπει στον χρήστη να παρακολουθεί τη συνολική δραστηριότητα του δικτύου. Σε περίπτωση που υπάρχει αυξημένη χρήση του δικτύου αυτό μπορεί να σχετίζεται με πιθανή κακόβουλη δραστηριότητα.



Εικόνα 227 Αριστερά: Σύντομη περιγραφή των παραγόντων που συνέβαλαν στην απόφαση Δεξιά: Ενημέρωση του συστήματος ότι η κίνηση είναι φυσιολογική



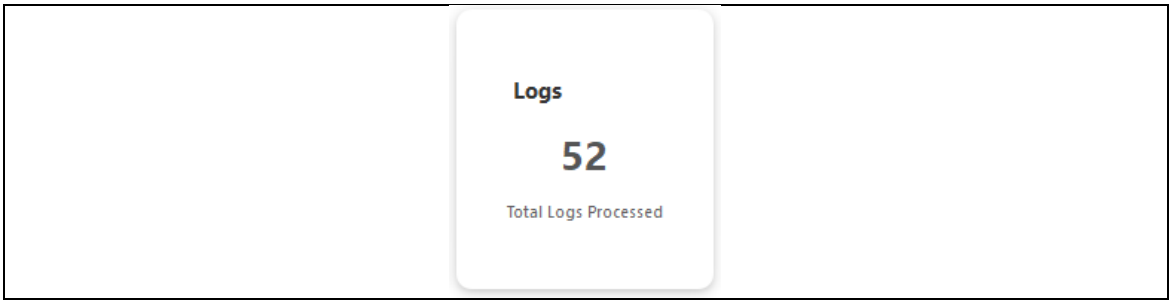
Εικόνα 228 Απεικόνιση της κάρτας Network Activity

Η κάρτα Logs (Εικόνα 229) παρουσιάζει τον συνολικό αριθμό των συμβάντων που έχουν επεξεργαστεί από το σύστημα κατά τη διάρκεια της λειτουργίας του. Αυτές οι εγγραφές προκύπτουν από την επεξεργασία των δικτυακών ροών και περιέχουν πληροφορίες σχετικά με την επικοινωνία μεταξύ των κόμβων του δικτύου, τα χαρακτηριστικά των ροών καθώς και το αποτέλεσμα της ανίχνευσης από τα μοντέλα του συστήματος.

Οι τεχνικοί ασφαλείας με την συγκεκριμένη κάρτα παρακολουθούν τον συνολικό όγκο των δεδομένων που έχουν αναλυθεί από το σύστημα, παρέχοντας μια εικόνα του φόρτου επεξεργασίας του IDS. Με τον τρόπο αυτό μπορεί να γίνει καλύτερη εκτίμηση της λειτουργίας του συστήματος και της ποσότητας των δεδομένων που έχουν καταγραφεί κατά τη διάρκεια της παρακολούθησης του δικτύου.

Η κάρτα System Usage (Εικόνα 230) εμφανίζει πληροφορίες σχετικά με τη χρήση των βασικών πόρων του συστήματος κατά τη διάρκεια λειτουργίας του IDS επιτρέποντας την παρακολούθηση του φόρτου που προκαλείται από τη διαδικασία ανάλυσης της δικτυακής κίνησης.

Συγκεκριμένα, η κάρτα εμφανίζει τρεις βασικούς δείκτες: τη χρήση της CPU, τη χρήση της RAM και τη δραστηριότητα του δίσκου (Disk). Οι τιμές αυτές υπολογίζονται δυναμικά και ενημερώνονται κατά τη διάρκεια εκτέλεσης της εφαρμογής.



Εικόνα 229 Εμφάνιση του συνολικού αριθμού των καταγεγραμμένων συμβάντων στην κάρτα Logs του dashboard

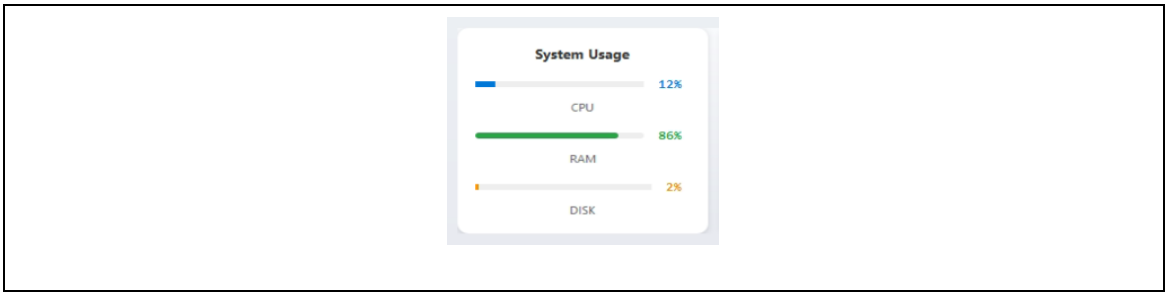
Η χρήση της CPU παρουσιάζει το ποσοστό επεξεργαστικής ισχύος που χρησιμοποιείται από το σύστημα, ενώ η χρήση της RAM δείχνει το ποσοστό της μνήμης που βρίσκεται σε χρήση. Επίσης, ο δείκτης του δίσκου απεικονίζει τη δραστηριότητα εισόδου/εξόδου δεδομένων, η οποία σχετίζεται με τη συνολική δραστηριότητα I/O του συστήματος και επηρεάζεται και από τη διαδικασία καταγραφής των δεδομένων του συστήματος.

Για τους δείκτες χρήσης της CPU και του δίσκου εφαρμόζεται δυναμική αλλαγή χρώματος, η οποία μετατρέπεται σε κόκκινο όταν η χρήση ξεπεράσει ένα προκαθορισμένο όριο κάνοντας έτσι πιο άμεσα αντιληπτή την αυξημένη επιβάρυνση των πόρων του συστήματος. Τέλος, οι δείκτες αυτοί απεικονίζονται με γραμμές προόδου (progress bars), επιτρέποντας στον χρήστη να αντιλαμβάνεται εύκολα τις περιπτώσεις αύξησης κατανάλωσης των πόρων.

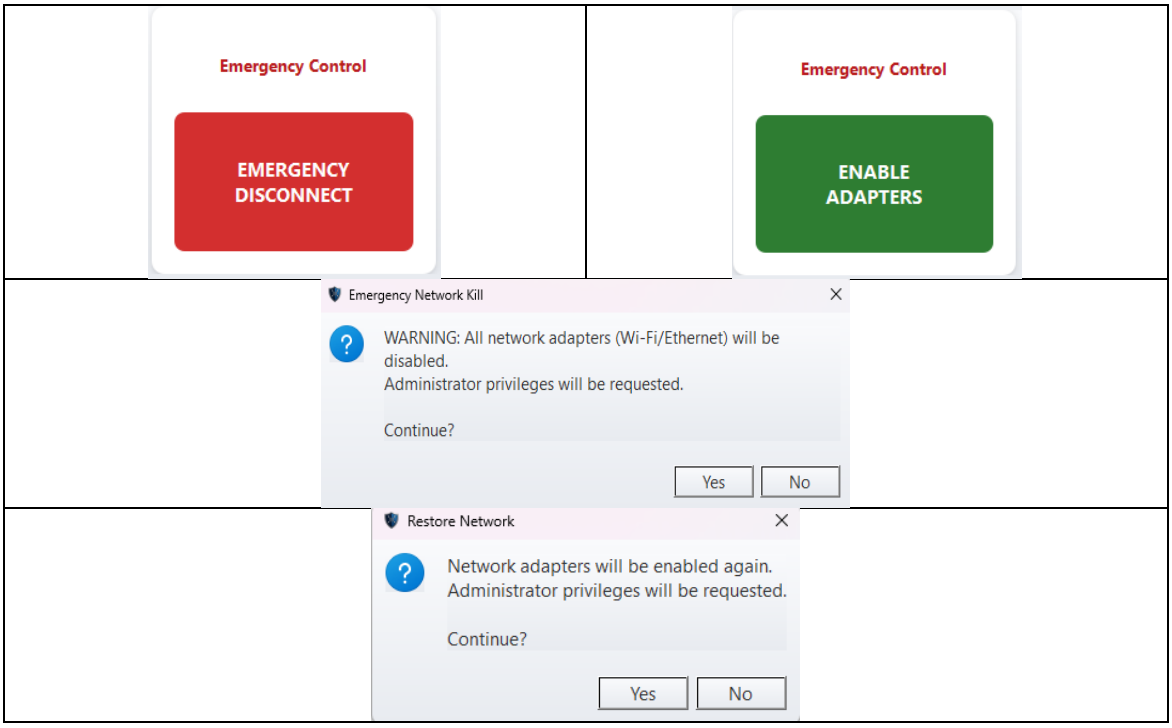
Η λειτουργία Emergency Control (Εικόνα 231) είναι ένας μηχανισμός ο οποίος έχει σχεδιαστεί για περιπτώσεις όπου απαιτείται ταχεία απομόνωση του συστήματος από το δίκτυο. Ο βασικός σκοπός του είναι να δίνει την δυνατότητα στον τεχνικό ασφαλείας να διακόπτει άμεσα τη δικτυακή συνδεσιμότητα του συστήματος, όταν εντοπίζεται σοβαρή κακόβουλη δραστηριότητα.

Η λειτουργία αυτή είναι σημαντική σε περιπτώσεις στις οποίες η συνέχιση της επικοινωνίας με το δίκτυο ενδέχεται να αυξήσει τον κίνδυνο όπως σε περιπτώσεις ενεργής επίθεσης, εξάπλωσης κακόβουλου λογισμικού και μη εξουσιοδοτημένης απομακρυσμένης πρόσβασης. Με την ενεργοποίηση του μηχανισμού εμφανίζεται σχετικό μήνυμα επιβεβαίωσης και αφού ο τεχνικός ασφαλείας εγκρίνει την ενέργεια το σύστημα οδηγείται σε κατάσταση απομόνωσης από το δίκτυο, με σκοπό να περιοριστεί η διάδοση ή η εκμετάλλευση της απειλής.

Επίσης, το Emergency Control εκτός από μέσο διακοπής της συνδεσιμότητας λειτουργεί και ως μέσο επαναφοράς της. Μετά την αντιμετώπιση του κινδύνου δίνεται η δυνατότητα επανενεργοποίησης της δικτυακής επικοινωνίας ώστε το σύστημα να επιστρέψει στην κανονική του λειτουργία. Και σε αυτή την περίπτωση με την ενεργοποίηση του μηχανισμού εμφανίζεται σχετικό μήνυμα επιβεβαίωσης και αφού η ενέργεια εγκριθεί το σύστημα επιστρέφει στην κανονική λειτουργία του.



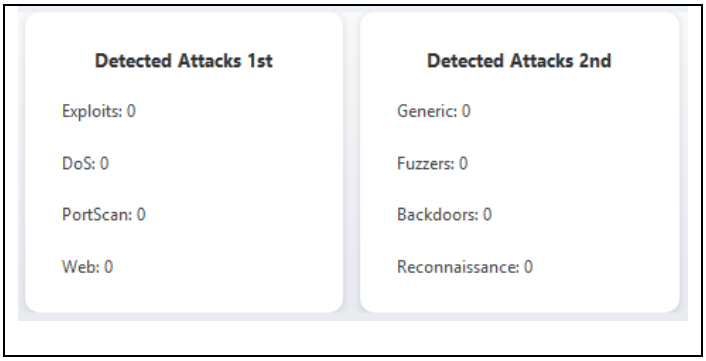
Εικόνα 230 Κάρτα System Usage που εμφανίζει σε πραγματικό χρόνο τη χρήση των πόρων του συστήματος (CPU, μνήμη RAM και δραστηριότητα δίσκου)



Εικόνα 231 Απεικόνιση της κάρτας Emergency Control, η οποία παρέχει δυνατότητα άμεσης απομόνωσης και επαναφοράς του συστήματος μαζί με το μήνυμα επιβεβαίωσης.

Η λειτουργία Emergency Control υλοποιήθηκε για να συμβάλλει στη γρηγορότερη αντίδραση και στη μείωση των πιθανών επιπτώσεων ενός περιστατικού ασφαλείας.

Οι κάρτες Detected Attacks 1st και Detected Attacks 2nd (φαίνονται στην Εικόνα 232) παρέχουν μια συνοπτική στατιστική απεικόνιση των κατηγοριών επιθέσεων που έχουν ανιχνευθεί από το σύστημα, κατά τη διάρκεια της παρακολούθησης της δικτυακής κίνησης και αποσκοπούν στην προσφορά μιας άμεσης εικόνας της κατανομής των ανιχνευμένων απειλών. Η πρώτη κάρτα παρουσιάζει τις πιο συχνά εμφανιζόμενες κατηγορίες επιθέσεων, δηλαδή τις Exploits, DoS, PortScan και Web attacks, ενώ η δεύτερη κάρτα περιλαμβάνει πρόσθετες κατηγορίες επιθέσεων όπως οι Generic, Fuzzers, Backdoors και Reconnaissance.



Εικόνα 232 Απεικόνιση των καρτών Detected Attacks, οι οποίες παρουσιάζουν συγκεντρωτικά τον αριθμό των ανιχνευμένων επιθέσεων ανά κατηγορία

Για κάθε κατηγορία εμφανίζεται ο συνολικός αριθμός των συμβάντων που έχουν καταγραφεί συμβάλλοντας στην άμεση αξιολόγηση της κακόβουλης δραστηριότητας.

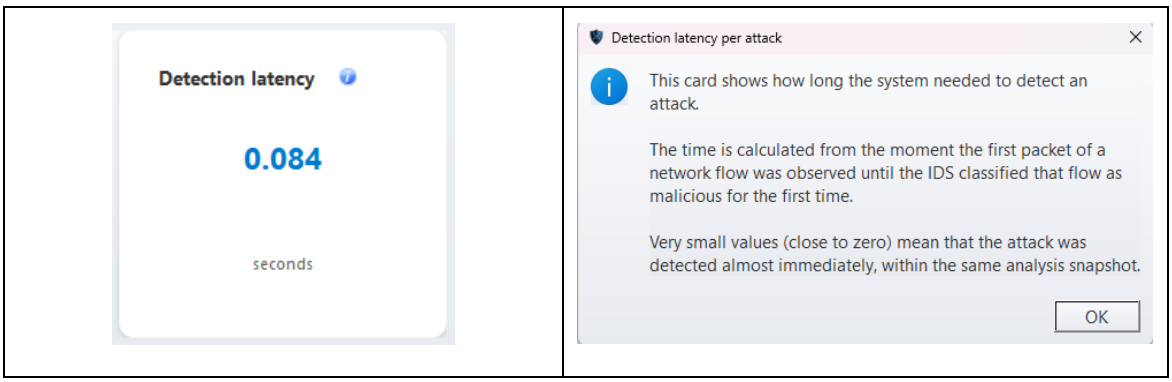
Οι κάρτες ενημερώνονται δυναμικά όσο προστίθενται νέες εγγραφές στο ιστορικό του συστήματος και λειτουργούν ως συνοπτικοί δείκτες της επιθετικής δραστηριότητας που έχει παρατηρηθεί, βοηθώντας στον άμεσο εντοπισμό των απειλών και την περαιτέρω διερεύνηση των περιστατικών αυτών.

Η κάρτα Detection Latency (Εικόνα 233) εμφανίζει την καθυστέρηση ανίχνευσης μιας επίθεσης δηλαδή τον χρόνο που απαιτείται μέχρι το σύστημα να αναγνωρίσει μια δικτυακή ροή ως κακόβουλη. Ο δείκτης αυτός εκφράζεται σε δευτερόλεπτα και υπολογίζεται από τον μαθηματικό τύπο:

$$\text{Detection Delay} = \text{first\_detect\_ts} - \text{first\_ts} \quad (15)$$

Η τιμή που εμφανίζεται προκύπτει από τη διαφορά μεταξύ της χρονικής στιγμής κατά την οποία παρατηρήθηκε για πρώτη φορά η συγκεκριμένη ροή και της χρονικής στιγμής κατά την οποία αυτή ταξινομήθηκε ως κακόβουλη. Η πληροφορία που παρέχει η κάρτα είναι σημαντική για την αξιολόγηση της αποτελεσματικότητας μιας εφαρμογής IDS, καθώς όσο μικρότερη είναι η καθυστέρηση ανίχνευσης τόσο ταχύτερα μπορεί να πραγματοποιηθεί η απόκριση σε μια πιθανή επίθεση.

Στην περίπτωση όπου η ανίχνευση πραγματοποιείται σχεδόν ακαριαία, η τιμή εμφανίζεται πολύ κοντά στο μηδέν, υποδεικνύοντας ότι η κακόβουλη ροή εντοπίστηκε σχεδόν αμέσως μετά την εμφάνισή της. Παράλληλα, στην κάρτα παρέχεται παράθυρο πληροφοριών (Εικόνα 233) το οποίο επεξηγεί τον τρόπο με τον οποίο υπολογίζεται ο χρόνος ανίχνευσης μιας επίθεσης παρέχοντας έτσι την πλήρη ερμηνεία του δείκτη.



Εικόνα 233 Αριστερά: Οπτικοποίηση Detection latency , Δεξιά: Πληροφοριακό παράθυρο της κάρτας Detection Latency, το οποίο επεξηγεί τον τρόπο υπολογισμού της καθυστέρησης ανίχνευσης μιας επίθεσης

7.4. Live detection και παρακολούθηση σε πραγματικό χρόνο

Η καρτέλα Live Detection (χαρακτηριστικά στιγμιότυπα βλέπουμε στην Εικόνα 234) είναι η οθόνη του Shield IDS η οποία δείχνει τη δικτυακή κίνηση σε πραγματικό χρόνο, παρουσιάζει δηλαδή στον χρήστη τις τρέχουσες ροές δικτυακής κίνησης που έχουν ήδη περάσει από το στάδιο συλλογής πακέτων, ομαδοποίησης σε ροές, εξαγωγής χαρακτηριστικών και ταξινόμησης από τα μοντέλα ανίχνευσης. Η συγκεκριμένη καρτέλα δεν εμφανίζει ακατέργαστα πακέτα, αλλά επεξεργασμένες ροές ώστε να αποκτάται άμεσα εικόνα της κατάστασης του δικτύου. Η διεπαφή της είναι απλή και αφορά έναν δυναμικό πίνακα, ο οποίος ενημερώνεται αυτόματα όσο εκτελείται η διαδικασία σύλληψης, χωρίς να απαιτείται χειροκίνητη ανανέωση.

Η λειτουργία της καρτέλας βασίζεται σε περιοδικά στιγμιότυπα των ενεργών ροών. Για κάθε ροή που παρατηρείται, το σύστημα υπολογίζει ένα σύνολο στατιστικών και χρονικών χαρακτηριστικών, τα οποία αποστέλλονται στα μοντέλα ταξινόμησης. Το Shield IDS υποστηρίζει συνδυαστική χρήση μοντέλων από δύο διαφορετικά σύνολα δεδομένων, το CIC-IDS2017 και το UNSW-NB15, ώστε η ίδια ροή να εξετάζεται μέσα από διαφορετικές προσεγγίσεις ανίχνευσης. Το αποτέλεσμα αυτής της διαδικασίας αποτυπώνεται στον πίνακα της καρτέλας Live Detection, όπου ο χρήστης μπορεί να δει αν η συγκεκριμένη επικοινωνία θεωρήθηκε φυσιολογική ή κακόβουλη, αλλά και ποια οικογένεια επίθεσης αναγνωρίστηκε, για τη περίπτωση της κακόβουλης επικοινωνίας.

Σε αυτή την καρτέλα οι πληροφορίες δίνονται άμεσα και με σαφήνεια. Οι εγγραφές ταξινομούνται, ώστε να εμφανίζονται πρώτα οι πιο πρόσφατες ροές, ενώ εφαρμόζεται χρωματική επισήμανση των γραμμών του πίνακα. Αναλυτικότερα, οι κακόβουλες ροές

επισημαίνονται με κόκκινο, οι κανονικές με πράσινο και οι μη ταξινομημένες περιπτώσεις με ουδέτερο γκρι.

Επίσης ένα πολύ σημαντικό χαρακτηριστικό της καρτέλας είναι η σύνδεσή της με τον μηχανισμό explainability του συστήματος. Όταν ο χρήστης επιλέγει μία συγκεκριμένη γραμμή του πίνακα το σύστημα τον προωθεί στην καρτέλα Explainability. Εκεί παρουσιάζεται μια πολύ αναλυτική ερμηνεία της απόφασης, όπως ο τύπος της επίθεσης, τα χαρακτηριστικά που συνέβαλαν περισσότερο στην πρόβλεψη και όπου υπάρχει αντιστοίχιση, η σχέση της δραστηριότητας που ανιχνεύθηκε με το πλαίσιο MITRE ATT&CK. Συνεπώς η καρτέλα Live Detection οδηγεί τον χρήστη από την απλή ανίχνευση στην ερμηνεία του περιστατικού.

Ο πίνακας της καρτέλας περιλαμβάνει ένα μεγάλο πλήθος στηλών που αντιστοιχούν τόσο στην ταυτότητα της ροής όσο και στα υπολογισμένα χαρακτηριστικά της. Οι βασικές στήλες ταυτοποίησης και χρονισμού είναι οι Timestamp, detect_ts, flow_start_ts, detection_delay, srcip, sport, dstip, dsport, proto, service και state. Οι στήλες αυτές περιγράφουν μεταξύ άλλων πότε παρατηρήθηκε η ροή, ποια είναι η πηγή και ο προορισμός της, ποιες θύρες και ποιο πρωτόκολλο χρησιμοποιεί, καθώς και ορισμένα στοιχεία κατάστασης που αξιοποιούνται από τα μοντέλα.

Παράλληλα, εμφανίζονται στήλες που αφορούν τον όγκο και τη συμπεριφορά της κίνησης, όπως οι sbytes, dbytes, ackdat, tcprtt, sloss, dloss, Flow Bytes/s, Sload, Dload, Dpkts, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Subflow Fwd Packets, Subflow Fwd bytes, Subflow Bwd Packets, Subflow Bwd Bytes και act_data_pkt_fwd. Οι στήλες αυτές δίνουν πληροφορία για τον όγκο δεδομένων ανά κατεύθυνση, το ρυθμό μεταφοράς, την καθυστέρηση επιβεβαίωσης, τις απώλειες, αλλά και τη γενικότερη εικόνα της ροής.

Η καρτέλα περιλαμβάνει επίσης χαρακτηριστικά που περιγράφουν μήκη πακέτων, σημαίες TCP και χρονικές αποστάσεις μεταξύ πακέτων, τα οποία είναι ιδιαίτερα χρήσιμα για την αναγνώριση ανώμαλων προτύπων. Σε αυτή την κατηγορία ανήκουν οι στήλες ACK Flag Count, FIN Flag Count, PSH Flag Count, SYN Flag Count, URG Flag Count, Average Packet Size, Packet Length Mean, Packet Length Std, Max Packet Length, Min Packet Length, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Packet Length Mean, Bwd Packet Length Max, Bwd Packet Length Mean, Bwd Packet Length Std, Fwd Header

Length.1, Fwd IAT Min, Fwd IAT Std, Bwd IAT Mean, Bwd IAT Std, Active Min, Active Mean, Active Std, Idle Mean, Init_Win_bytes_forward και Init_Win_bytes_backward.

Εφόσον χρησιμοποιείται το προφίλ χαρακτηριστικών του UNSW-NB15, εμφανίζονται και πρόσθετες στήλες εξειδικευμένων δεικτών, όπως οι sttl, dttl, ct_state_ttl, ct_srv_dst, ct_srv_src, ct_dst_src_ltm, ct_dst_sport_ltm, ct_src_dport_ltm, ct_ftp_cmd, swin, dwin, smeansz, dmeansz, synack και is_sm_ips_ports. Τα πεδία αυτά προσθέτουν πληροφορία σχετική με TTL, συσχετισμούς υπηρεσιών, επαναλαμβανόμενα μοτίβα προορισμών και θυρών, μεγέθη παραθύρων TCP και ειδικούς δείκτες συμπεριφοράς, οι οποίοι είναι χρήσιμοι κυρίως για ταξινόμηση σε πολλαπλές κατηγορίες επιθέσεων σε περιβάλλοντα που ακολουθούν τη λογική του UNSW.

Επίσης εμφανίζονται και στήλες που σχετίζονται άμεσα με το αποτέλεσμα της ταξινόμησης. Ο χρήστης μπορεί να δει τις στήλες attack_cic, pred_unsw, attack_unsw, prediction, attack_type και την ενοποιημένη στήλη Attack Type, η οποία εμφανίζει τον τύπο επίθεσης. Η συγκεκριμένη στήλη τοποθετείται σκόπιμα μετά το Timestamp, ώστε το σημαντικότερο επιχειρησιακό συμπέρασμα, δηλαδή ο τύπος της επίθεσης να είναι ορατό με την πρώτη ματιά.

Στο εσωτερικό της εφαρμογής ενώ επεξεργάζονται περισσότερες εγγραφές ανά στιγμιότυπο, στον ορατό πίνακα της καρτέλας προβάλλεται ένα περιορισμένο υποσύνολο των πιο πρόσφατων ροών το οποίο είναι εκατό εγγραφές, ώστε να αποφεύγεται η υπερφόρτωση λόγω υπερβολικά μεγάλου όγκου πληροφορίας. Έτσι, η καρτέλα Live Detection πετυχαίνει μια ισορροπία ανάμεσα στη λεπτομέρεια των χαρακτηριστικών και στη χρηστικότητα του περιβάλλοντος.

Real-Time IDS Monitor

	Timestamp	Attack Type	detect_ts	flow_start_ts	detection_delay	srcip	sport	dstip	dsport	proto	service	sbytes	dbytes	state	ackdat	tcprtt	sloss	dloss
0	2026-03-09 11:47:26	Normal	1773049646.9229543	1773049646.6631882	0.0	150.	42759	239	1900	UDP	-	1988	0	INT	0	0.0	0.0	0.0
1	2026-03-09 11:47:26	UNSW → Exploits	1773049646.5142822	1773049646.4671617	0.0	150.	445	150	51703	TCP	-	885	1449	CON	2268	766.0388946533203	0.0	0.0
2	2026-03-09 11:47:26	Normal	1773049646.2181876	1773049646.2181876	0.0	192.	0	224	0	ospf	-	94	0	INT	0	0.0	0.0	0.0
3	2026-03-09 11:47:26	UNSW → Exploits	1773049646.4663455	1773049644.3588746	0.061525821685791016	150.	445	150	51702	TCP	-	1015	1503	CON	2452	419.1398620605469	0.0	0.0
4	2026-03-09 11:47:26	UNSW → Exploits	1773049646.536671	1773049642.2821689	0.05406808853149414	150.	445	150	32857	TCP	-	3798	4404	CON	8202	0.0	0.0	0.0
5	2026-03-09 11:47:26	UNSW → Exploits	1773049646.8823924	1773049640.38072	0.7232508659362793	150.	7070	94.6	7964	TCP	-	449826	8265	CON	458121	0.0	0.0	0.0
6	2026-03-09 11:47:26	Normal	1773049646.653044	1773049646.396513	0.0	150.	31285	239	1900	UDP	-	1988	0	INT	0	0.0	0.0	0.0
7	2026-03-09 11:47:25	Normal	1773049645.334582	1773049640.5557528	0.0002152919769287109	150.	64735	239	1900	UDP	-	980	0	INT	0	0.0	0.0	0.0
8	2026-03-09 11:47:25	Normal	1773049645.0853186	1773049640.884309	0.0	150.	1900	239	1900	UDP	-	4212	0	INT	0	0.0	0.0	0.0
9	2026-03-09 11:47:25	Normal	1773049645.7124844	1773049641.679828	0.0	150.	52718	239	1900	UDP	-	286	0	INT	0	0.0	0.0	0.0
10	2026-03-09 11:47:25	Normal	1773049645.7191877	1773049645.7191877	0.0	150.	42705	224	5353	UDP	-	86	0	INT	0	0.0	0.0	0.0
11	2026-03-09 11:47:23	Normal	1773049645.703289	1773049645.703289	0.0	150.	60028	224	5353	UDP	-	86	0	INT	0	0.0	0.0	0.0
12	2026-03-09 11:47:25	Normal	1773049645.7001803	1773049642.694178	0.0	150.	62394	239	1900	UDP	-	286	0	INT	0	0.0	0.0	0.0
13	2026-03-09 11:47:24	Normal	1773049644.5867686	1773049641.5798697	0.0	150.	51551	239	1900	UDP	-	286	0	INT	0	0.0	0.0	0.0
14	2026-03-09 11:47:24	Normal	1773049644.7852302	1773049641.7786813	0.0	150.	60630	239	1900	UDP	-	286	0	INT	0	0.0	0.0	0.0
15	2026-03-09 11:47:24	UNSW → Exploits	1773049644.0126042	1773049643.962837	0.04976725578308105	150.	49674	150	9079	TCP	-	130	54	CON	184	0.0	0.0	0.0
16	2026-03-09 11:47:24	Normal	1773049644.9877722	1773049644.9877722	0.0	150.	55127	239	1900	UDP	-	143	0	INT	0	0.0	0.0	0.0
17	2026-03-09 11:47:24	UNSW → Exploits	1773049644.3579812	1773049642.2820396	0.049667835235595703	150.	445	150	51698	TCP	-	1015	1503	CON	2452	54.12101745605469	0.0	0.0
18	2026-03-09 11:47:24	Normal	1773049644.0048013	1773049642.015862	0.009181261062622070	150.	6771	239	6771	UDP	-	1449	0	INT	0	0.0	0.0	0.0
19	2026-03-09 11:47:24	Normal	1773049644.0878386	1773049641.0857222	0.0	150.	49537	239	1900	UDP	-	368	0	INT	0	0.0	0.0	0.0
20	2026-03-09 11:47:23	Normal	1773049643.401657	1773049643.3536406	0.048016548156738281	138.	443	150	49927	TCP	https	66	55	CON	121	0.0	0.0	0.0
21	2026-03-09 11:47:22	Normal	1773049642.2819686	1773049642.2814584	0.0	150.	445	150	51695	TCP	-	130	54	CON	184	0.0	0.0	0.0
22	2026-03-09 11:47:22	CIC → PortScan	1773049642.3316278	1773049642.3316278	0.0	150.	8868	216.	21780	TCP	-	0	60	CON	0	0.0	0.0	0.0
23	2026-03-09 11:47:21	UNSW → Exploits	1773049641.8530471	1773049641.8021433	0.050903797149658203	150.	49675	150	9079	TCP	-	130	54	CON	184	0.0	0.0	0.0
24	2026-03-09 11:47:20	Normal	1773049640.5569108	1773049640.4839854	0.07292532920837402	13.6	443	150.	51682	TCP	https	622	4422	CON	5044	0.0	0.0	0.0
25	2026-03-09 11:47:20	Normal	1773049640.7113035	1773049640.7113035	0.0	150.	50888	239	1900	UDP	-	179	0	INT	0	0.0	0.0	0.0

Real-Time IDS Monitor

	ACK Flag Count	Active Min	Active Std	Average Packet Size	Avg Bwd Segment Size	Avg Fwd Segment Size	Bwd IAT Mean	Bwd IAT Std	Bwd Packet Length Max	Bwd Packet Length Mean
0	0	9829.25	83959.80934283379	331.3333333333333	0.0	331.3333333333333	0.0	0.0	0.0	0.0
1	8	44.25	13550.84544568968	259.3333333333333	289.8	221.25	1348.3125	751.1838651214161	759.0	289.8
2	0	0.0	0.0	94.0	0.0	94.0	0.0	0.0	0.0	0.0
3	10	46.75	611876.5093892333	228.9090909090909	250.5	203.0	421494.2	840683.3412764494	759.0	250.5
4	33	37.75	492330.75328228966	248.54545454545453	244.66666666666666	253.2	250264.83823529413	653405.0853626779	430.0	244.66666666666666
5	249	12.75	63655.93308327184	1839.8433734939758	80.53398058252426	3081.0	63362.575980392154	112255.84598429274	105.0	80.53398058252426
6	0	6499.75	84443.37221200371	331.3333333333333	0.0	331.3333333333333	0.0	0.0	0.0	0.0
7	0	39.5	1911356.8253982405	163.33333333333334	0.0	163.33333333333334	0.0	0.0	0.0	0.0
8	0	1518.5	465342.1014616641	526.5	0.0	526.5	0.0	0.0	0.0	0.0
9	0	4032656.25	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
10	0	0.0	0.0	86.0	0.0	86.0	0.0	0.0	0.0	0.0
11	0	0.0	0.0	86.0	0.0	86.0	0.0	0.0	0.0	0.0
12	0	3006002.25	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
13	0	3006899.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
14	0	3006549.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
15	2	49767.25	0.0	92.0	54.0	130.0	0.0	0.0	54.0	54.0
16	0	0.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
17	10	41.0	606272.0630172481	228.9090909090909	250.5	203.0	415188.3	828734.1738341885	759.0	250.5
18	0	260.5	653251.8905422408	161.0	0.0	161.0	0.0	0.0	0.0	0.0
19	0	3002116.5	0.0	184.0	0.0	184.0	0.0	0.0	0.0	0.0
20	2	48016.5	0.0	60.5	55.0	66.0	0.0	0.0	55.0	55.0
21	2	510.0	0.0	92.0	54.0	130.0	0.0	0.0	54.0	54.0
22	0	0.0	0.0	60.0	60.0	0.0	0.0	0.0	60.0	60.0
23	2	50904.0	0.0	92.0	54.0	130.0	0.0	0.0	54.0	54.0
24	6	55.5	25078.27831730081	840.6666666666666	1474.0	207.33333333333334	36462.625	36335.625	3858.0	1474.0
25	0	0.0	0.0	179.0	0.0	179.0	0.0	0.0	0.0	0.0

Real-Time IDS Monitor

	Bwd Packet Length Mean	Bwd Packet Length Std	FIN Flag Count	Flow Bytes/s	Fwd Header Length.1	Fwd IAT Min	Fwd IAT Std	Fwd Packet Length Mean	Idle Mean	Init_Win_bytes_backward
0	0.0	0.0	0	7653.038583326143	0	9829.25	83959.80934283379	331.3333333333333	0.0	0.0
1	289.8	258.5856917928755	0	49532.506582742186	92	1565.25	19565.33626830029	221.25	0.0	0.0
2	0.0	0.0	0	94000000.0	0	0.0	0.0	94.0	0.0	0.0
3	250.5	251.88208219985268	0	1194.7969926377139	112	1457.0	877291.5314143903	203.0	0.0	0.0
4	244.66666666666666	135.91500612106641	0	1927.8401768594833	300	181.0	728809.6539932288	253.2	0.0	0.0
5	80.53398058252426	22.413941368043872	0	70462.0233568834	2920	12.75	88329.61443992467	3081.0	0.0	0.0
6	0.0	0.0	0	7749.550732409577	0	6499.75	84443.37221200371	331.3333333333333	0.0	0.0
7	0.0	0.0	0	205.07114422005378	0	39.5	1911356.8253982405	163.33333333333334	0.0	0.0
8	0.0	0.0	0	1002.6161540508564	0	1518.5	465342.1014616641	526.5	0.0	0.0
9	0.0	0.0	0	70.92099336491904	0	4032656.25	0.0	143.0	0.0	0.0
10	0.0	0.0	0	86000000.0	0	0.0	0.0	86.0	0.0	0.0
11	0.0	0.0	0	86000000.0	0	0.0	0.0	86.0	0.0	0.0
12	0.0	0.0	0	95.14297799499639	0	3006002.25	0.0	143.0	0.0	0.0
13	0.0	0.0	0	95.11460525055487	0	3006899.0	0.0	143.0	0.0	0.0
14	0.0	0.0	0	95.12567773532598	0	3006549.0	0.0	143.0	0.0	0.0
15	54.0	0.0	0	3697.2100853218612	20	0.0	0.0	130.0	0.0	0.0
16	0.0	0.0	0	143000000.0	0	0.0	0.0	143.0	0.0	0.0
17	250.5	251.88208219985268	0	1212.9435843957767	112	161.75	870312.616999283	203.0	0.0	0.0
18	0.0	0.0	0	728.528938530369	0	260.5	653251.8905422408	161.0	0.0	0.0
19	0.0	0.0	0	122.58018872457282	0	3002116.5	0.0	184.0	0.0	0.0
20	55.0	0.0	0	2519.9645673201057	32	0.0	0.0	66.0	0.0	0.0
21	54.0	0.0	0	360631.74579439254	20	0.0	0.0	130.0	0.0	0.0
22	60.0	0.0	0	60000000.0	0	0.0	0.0	0.0	0.0	0.0
23	54.0	0.0	0	3614.661583281032	20	0.0	0.0	130.0	0.0	0.0
24	1474.0	1695.9905660114975	0	69166.64010645011	60	55.5	4109.0	207.33333333333334	0.0	0.0
25	0.0	0.0	0	179000000.0	0	0.0	0.0	179.0	0.0	0.0

Real-Time IDS Monitor

	Init_Win_bytes_forward	Max Packet Length	Min Packet Length	PSH Flag Count	Packet Length Mean	Packet Length Std	SYN Flag Count	Subflow Bwd Bytes	Subflow Bwd Packets	Subflow Fwd Packets
0	0.0	375.0	305.0	0	331.3333333333333	31.09483701338357	0	0	0	6
1	0.0	759.0	54.0	5	259.3333333333333	222.5833177336822	2	1449	5	4
2	0.0	94.0	94.0	0	94.0	0.0	0	0	0	1
3	0.0	759.0	54.0	6	228.9090909090909	212.04564454829253	2	1503	6	5
4	0.0	430.0	54.0	30	248.54545454545453	125.09513734925811	0	4404	18	15
5	0.0	11058.0	54.0	171	1839.8433734939758	3160.8568426111541	0	8295	103	146
6	0.0	375.0	305.0	0	331.3333333333333	31.09483701338357	0	0	0	6
7	0.0	179.0	136.0	0	163.33333333333334	19.396448701301 54	0	0	0	6
8	0.0	565.0	485.0	0	526.5	28.818396901979124	0	0	0	8
9	0.0	143.0	143.0	0	143.0	0.0	0	0	0	2
10	0.0	86.0	86.0	0	86.0	0.0	0	0	0	1
11	0.0	86.0	86.0	0	86.0	0.0	0	0	0	1
12	0.0	143.0	143.0	0	143.0	0.0	0	0	0	2
13	0.0	143.0	143.0	0	143.0	0.0	0	0	0	2
14	0.0	143.0	143.0	0	143.0	0.0	0	0	0	2
15	0.0	130.0	54.0	1	92.0	38.0	0	54	1	1
16	0.0	143.0	143.0	0	143.0	0.0	0	0	0	1
17	0.0	759.0	54.0	6	228.9090909090909	212.04564454829253	2	1503	6	5
18	0.0	161.0	161.0	0	161.0	0.0	0	0	0	9
19	0.0	184.0	184.0	0	184.0	0.0	0	0	0	2
20	0.0	66.0	55.0	0	60.5	5.5	0	55	1	1
21	0.0	130.0	54.0	1	92.0	38.0	0	54	1	1
22	0.0	60.0	60.0	0	60.0	0.0	1	60	1	0
23	0.0	130.0	54.0	1	92.0	38.0	0	54	1	1
24	0.0	3858.0	54.0	3	840.6666666666666	1364.1884848591203	0	4422	3	3
25	0.0	179.0	179.0	0	179.0	0.0	0	0	0	1

Real-Time IDS Monitor

	Subflow Bwd Packets	Subflow Fwd Packets	Total Backward Packets	Total Fwd Packets	Total Length of Bwd Packets	Total Length of Fwd Packets	URG Flag Count	act_data_pkt_fwd	sttl	dttl	ct_state_ttl
0	0	6	0	6	0	1988	0	6	2.0	0.0	41
1	5	4	5	4	1449	885	0	2	127.0	128.0	123
2	0	1	0	1	0	94	0	1	1.0	0.0	41
3	6	5	6	5	1503	1015	0	3	127.0	128.0	123
4	18	15	18	15	4404	3798	0	15	128.0	128.0	203
5	103	146	103	146	8295	449826	0	124	128.0	120.0	203
6	0	6	0	6	0	1988	0	6	2.0	0.0	41
7	0	6	0	6	0	980	0	6	3.0	0.0	41
8	0	8	0	8	0	4212	0	8	2.0	0.0	41
9	0	2	0	2	0	286	0	2	2.0	0.0	41
10	0	1	0	1	0	86	0	1	1.0	0.0	41
11	0	1	0	1	0	86	0	1	1.0	0.0	41
12	0	2	0	2	0	286	0	2	2.0	0.0	41
13	0	2	0	2	0	286	0	2	2.0	0.0	41
14	0	2	0	2	0	286	0	2	2.0	0.0	41
15	1	1	1	1	54	130	0	1	127.0	128.0	123
16	0	1	0	1	0	143	0	1	2.0	0.0	41
17	6	5	6	5	1503	1015	0	3	127.0	128.0	123
18	0	9	0	9	0	1449	0	9	253.0	0.0	9
19	0	2	0	2	0	368	0	2	2.0	0.0	41
20	1	1	1	1	55	66	0	0	54.0	128.0	2
21	1	1	1	1	54	130	0	1	127.0	128.0	123
22	1	0	1	0	60	0	0	0	0.0	57.0	0
23	1	1	1	1	54	130	0	1	127.0	128.0	123
24	3	3	3	3	4422	622	0	1	111.0	128.0	123
25	0	1	0	1	0	179	0	1	3.0	0.0	41

Real-Time IDS Monitor

	ct_srv_dst	ct_srv_src	ct_dst_src_ltm	ct_dst_sport_ltm	ct_src_dport_ltm	Dpkts	ct_ftp_cmd	swin	dwin	smeansz	dmeansz	synack	Sload	Dload	is_sm_ips_ports
0	38	12	12	6	12	0	0	0.0	0.0	331.3333333333333	0.0	0.0	7653.038583326143	0.0	0
1	70	35	35	66	9	5	0	0.0	0.0	221.25	289.8	766.0388946533203	18781.60596646394	30750.900616278246	0
2	1	1	1	1	1	0	0	0.0	0.0	94.0	0.0	0.0	94000000.0	0.0	1
3	70	35	35	66	11	6	0	0.0	0.0	203.0	250.5	419.1398620605469	481.61991561845895	713.177077019255	0
4	70	33	33	66	33	18	0	0.0	0.0	253.2	244.66666666666666	0.0	892.7014132787513	1035.138763580732	0
5	249	250	249	249	249	103	0	0.0	0.0	3081.0	80.53398058252426	0.0	69186.19779170444	1275.8255651789543	0
6	38	12	12	6	12	0	0	0.0	0.0	331.3333333333333	0.0	0.0	7749.550732409577	0.0	0
7	38	6	6	6	6	0	0	0.0	0.0	163.33333333333334	0.0	0.0	205.07114422005378	0.0	0
8	38	8	8	8	8	0	0	0.0	0.0	526.5	0.0	0.0	1002.6161540508564	0.0	1
9	38	2	2	2	2	0	0	0.0	0.0	143.0	0.0	0.0	70.92099336491904	0.0	0
10	2	2	2	1	2	0	0	0.0	0.0	86.0	0.0	0.0	86000000.0	0.0	0
11	2	2	2	1	2	0	0	0.0	0.0	86.0	0.0	0.0	86000000.0	0.0	0
12	38	2	2	2	2	0	0	0.0	0.0	143.0	0.0	0.0	95.14297799499639	0.0	0
13	38	2	2	2	2	0	0	0.0	0.0	143.0	0.0	0.0	95.11460525055487	0.0	0
14	38	2	2	2	2	0	0	0.0	0.0	143.0	0.0	0.0	95.12567773532598	0.0	0
15	70	35	35	2	2	1	0	0.0	0.0	130.0	54.0	0.0	2612.1592994121847	1085.0507859096767	0
16	38	1	1	1	1	0	0	0.0	0.0	143.0	0.0	0.0	143000000.0	0.0	0
17	70	35	35	66	11	6	0	0.0	0.0	203.0	250.5	54.12101745605469	488.934764956995	724.0088194387818	0
18	9	9	9	9	9	0	0	0.0	0.0	161.0	0.0	0.0	728.528938530369	0.0	1
19	38	2	2	2	2	0	0	0.0	0.0	184.0	0.0	0.0	122.58018872457282	0.0	0
20	8	2	2	8	2	1	0	0.0	0.0	66.0	55.0	0.0	1374.5261276291485	1145.438439690957	0
21	70	35	35	66	2	1	0	0.0	0.0	130.0	54.0	0.0	254794.16822429906	105837.57757009346	0
22	1	250	1	1	1	1	0	0.0	0.0	0.0	60.0	0.0	0.0	60000000.0	0
23	70	2	2	2	2	1	0	0.0	0.0	130.0	54.0	0.0	2553.8369881876856	1060.8245950933463	0
24	8	6	6	8	6	3	0	0.0	0.0	207.33333333333334	1474.0	0.0	8529.272431842182	60637.36767460792	0
25	38	1	1	1	1	0	0	0.0	0.0	179.0	0.0	0.0	179000000.0	0.0	0

is_sm_ips_ports	attack_cic	pred_unsw	attack_unsw	prediction	attack_type	color_tag
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	4	Exploits	1	Exploits	red
1	BENIGN	9	Normal	0	Normal	green
0	BENIGN	4	Exploits	1	Exploits	red
0	BENIGN	4	Exploits	1	Exploits	red
0	BENIGN	4	Exploits	1	Exploits	red
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	4	Normal	0	Exploits	green
1	BENIGN	9	Normal	0	Normal	green
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	9	Normal	0	Normal	green
0	BENIGN	9	Normal	0	Normal	green
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	4	Exploits	1	Exploits	red
0	BENIGN	9	Normal	0	Normal	green
0	BENIGN	4	Exploits	1	Exploits	red
1	BENIGN	2	Normal	0	Backdoors	green
0	BENIGN	4	Normal	0	Exploits	green
0	BENIGN	4	Normal	1	Normal	green
0	BENIGN	9	Normal	0	Normal	green
0	PortScan	9	Normal	1	PortScan	red
0	BENIGN	4	Exploits	1	Exploits	red
0	BENIGN	4	Normal	1	Normal	green
0	BENIGN	9	Normal	0	Normal	green

Εικόνα 234 Καρτέλα Live Detection. Εμφάνιση ροών με κανονική και κακόβουλη κίνηση

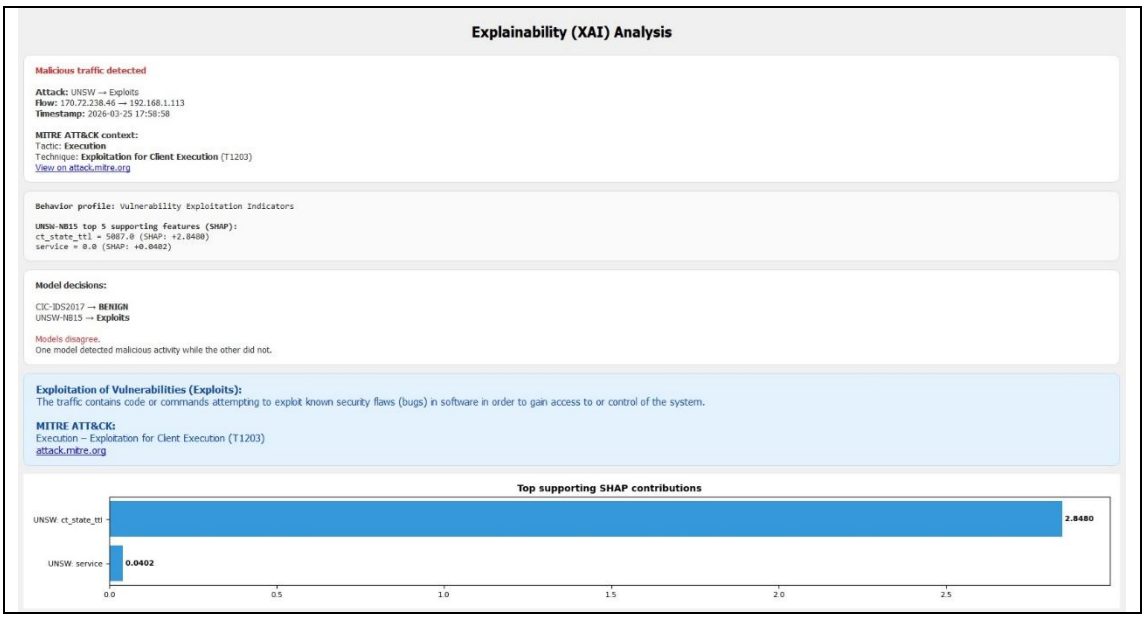
7.5. Περιβάλλον του Explainability

Μια σημαντική λειτουργία του Shield IDS είναι η καρτέλα Explainability (φαίνεται στην Εικόνα 235) η οποία σχεδιάστηκε με σκοπό να επιτρέπει στον χρήστη να κατανοεί τις αποφάσεις του συστήματος, δηλαδή τους λόγους για τους οποίους μια συγκεκριμένη ροή ταξινομήθηκε ως κανονική ή ως συγκεκριμένο είδος επίθεσης. Η δυνατότητα κατανόησης των αποφάσεων ενός συστήματος τεχνητής νοημοσύνης είναι ένας από τους βασικούς στόχους της XAI, καθώς οι τεχνικοί ασφαλείας χρειάζονται πληροφορίες για τον τρόπο με τον οποίο ένα μοντέλο καταλήγει σε μια συγκεκριμένη πρόβλεψη [49]. Η ένδειξη ότι μια ροή χαρακτηρίστηκε ως κακόβουλη δεν είναι πάντοτε επαρκής διότι ο τεχνικός ασφαλείας χρειάζεται πρόσθετα στοιχεία προκειμένου να αξιολογήσει το περιστατικό. Στη βιβλιογραφία αναφέρεται ότι πολλά μοντέλα μηχανικής μάθησης λειτουργούν ως μαύρα κουτιά - black boxes, το οποίο καθιστά δύσκολη την κατανόηση της λογικής πίσω από τις αποφάσεις τους με αποτέλεσμα να μειώνεται η εμπιστοσύνη των χρηστών στα αποτελέσματα του συστήματος [49]. [49].

Για τον λόγο αυτό, το Explainability περιβάλλον δεν περιορίζεται στην εμφάνιση της τελικής πρόβλεψης, αλλά παρέχει μια αναλυτική ερμηνεία χάρη στην οποία ο χρήστης μπορεί να εξετάσει γιατί μια συγκεκριμένη ροή θεωρήθηκε κακόβουλη. Αφού μια ροή

ταξινομηθεί από τα μοντέλα του Shield IDS η πληροφορία αυτή προωθείται στο περιβάλλον Explainability όπου οργανώνεται σε κατανοητή μορφή. Εκεί παρουσιάζονται συνοπτικά στοιχεία για το περιστατικό, το είδος της δραστηριότητας που ανιχνεύθηκε, τα βασικά χαρακτηριστικά της ροής που συνέβαλαν στην απόφαση, καθώς και πρόσθετο επεξηγηματικό περιεχόμενο που είναι χρήσιμο για τον χρήστη. Η παρουσίαση των χαρακτηριστικών που επηρέασαν την πρόβλεψη αποτελεί βασική διαδικασία των τεχνικών επεξηγησιμότητας, οι οποίες επιτρέπουν να αναδειχθούν το πόσο σημαντικά είναι τα χαρακτηριστικά ενός μοντέλου κατά τη διαδικασία λήψης αποφάσεων [49], [50].

Το Explainability περιβάλλον δεν αντιμετωπίζει όλες τις επιθέσεις με τον ίδιο τρόπο αλλά προσπαθεί να αποδώσει σε κάθε ειδοποίηση μία ερμηνεία ανάλογα με την οικογένεια της επίθεσης. Έτσι η ερμηνεία δεν είναι γενική αλλά συνδέεται με τα ιδιαίτερα χαρακτηριστικά που αναμένονται σε κατηγορίες όπως DoS, PortScan, Web Attacks, Exploits, Fuzzers ή άλλες μορφές επιθέσεων. Η δυνατότητα παροχής κατανοητών εξηγήσεων θεωρείται σημαντική για τη συνεργασία ανθρώπου και συστημάτων τεχνητής νοημοσύνης, καθώς ενισχύει την εμπιστοσύνη των χρηστών και επιτρέπει να αξιολογούνται καλύτερα τα αποτελέσματα [45]. Η προσέγγιση αυτή ενισχύει σημαντικά την αξιοπιστία και τη χρηστικότητα του συστήματος, καθώς ο χρήστης δεν ενημερώνεται μόνο για το αποτέλεσμα, αλλά και για τη λογική που οδήγησε σε αυτό.



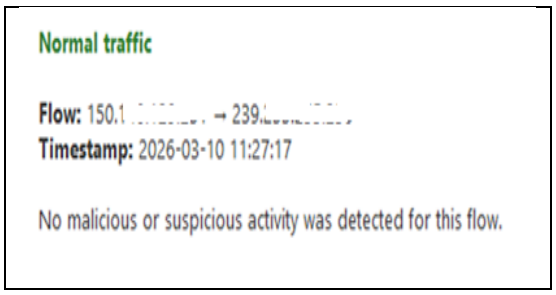
Εικόνα 235 Περιβάλλον Επεξηγησιμότητας

7.5.1. Σύνοψη περιστατικού στο XAI περιβάλλον

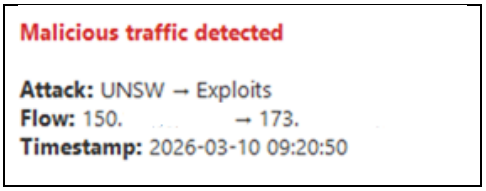
Η καρτέλα Explainability του Shield IDS ξεκινά με μια ενότητα σύνοψης η οποία παρέχει τα βασικά στοιχεία της ροής που πρόκειται να αναλυθεί. Σκοπός της ενότητας αυτής είναι να προσφέρει μια άμεση και κατανοητή πρώτη εικόνα του συμβάντος, πριν παρουσιαστούν οι πιο εξειδικευμένες πληροφορίες που αφορούν το προφίλ συμπεριφοράς της επίθεσης, τη διασύνδεση με το MITRE ATT&CK και τη σύγκριση των μοντέλων ανίχνευσης.

Στη συγκεκριμένη περιοχή απεικονίζεται αρχικά ο γενικός χαρακτηρισμός του περιστατικού, όπως για παράδειγμα η ένδειξη με κόκκινο χρώμα (βλέπε Εικόνα 237) ότι ανιχνεύθηκε κακόβουλη δικτυακή κίνηση ή αντίθετα, ότι η ροή αξιολογήθηκε ως φυσιολογική, με πράσινο χρώμα (Εικόνα 236). Στη συνέχεια, παρουσιάζονται συνοπτικά, η ροή επικοινωνίας και η χρονική σήμανση του συμβάντος, ενώ στις περιπτώσεις κακόβουλης δραστηριότητας εμφανίζεται και η κατηγορία της επίθεσης που αποδόθηκε στη ροή. Έτσι, ο χρήστης μπορεί να δει τις διευθύνσεις προέλευσης και προορισμού της επικοινωνίας και τη χρονική στιγμή κατά την οποία καταγράφηκε το περιστατικό, ενώ παράλληλα, μπορεί να γνωρίζει από την αρχή ποιο περιστατικό εξετάζεται και ποια είναι τα βασικά του στοιχεία.

Στην υλοποίηση του Shield IDS η ενότητα αυτή ενημερώνεται δυναμικά με βάση το περιστατικό που έχει επιλεγεί προς ανάλυση. Το σύστημα ανακτά τα σχετικά πεδία της ροής, όπως τον τύπο της επίθεσης, τις διευθύνσεις IP πηγής και προορισμού και τη χρονική σήμανση και τα παρουσιάζει σε συνοπτική μορφή στην κορυφή της καρτέλας Explainability.



Εικόνα 236 Εμφάνιση σύνοψης όταν η δικτυακή κίνηση είναι κανονική



Εικόνα 237 Εμφάνιση σύνοψης όταν η ροή έχει χαρακτηριστεί ως κακόβουλη

Έτσι, πριν ο χρήστης προχωρήσει στην εξέταση των χαρακτηριστικών που συνέβαλαν στην απόφαση ή στη μελέτη της αντιστοίχισης με το MITRE ATT&CK, έχει ήδη στη διάθεσή του μια ξεκάθαρη σύνοψη του ίδιου του περιστατικού.

7.5.2. Ερμηνεία χαρακτηριστικών μέσω SHAP και προφίλ ανά τύπο επίθεσης

Το περιβάλλον Explainability του Shield IDS δεν αντιμετωπίζει όλες τις επιθέσεις με τον ίδιο τρόπο αλλά προσαρμόζει την ερμηνεία του ανάλογα με το είδος της κακόβουλης δραστηριότητας που έχει ανιχνευθεί. Κάθε κατηγορία επίθεσης εμφανίζει διαφορετικά μοτίβα συμπεριφοράς στη δικτυακή κίνηση και επομένως δεν είναι κατάλληλο να παρουσιάζεται η ίδια μορφή εξήγησης για όλα τα περιστατικά. Για τον λόγο αυτό, το σύστημα δεν περιορίζεται μόνο στην εμφάνιση της τελικής πρόβλεψης, αλλά εμφανίζει τα χαρακτηριστικά της ροής που υποστήριξαν τη συγκεκριμένη απόφαση, παρέχοντας έτσι μια πιο κατανοητή ερμηνεία.

Η διαδικασία αυτή έχει σχεδιαστεί και υλοποιηθεί με τέτοιο τρόπο ώστε να πραγματοποιείται σε δύο επίπεδα. Όταν είναι διαθέσιμα τα κατάλληλα μοντέλα επεξήγησης, το σύστημα αξιοποιεί την τεχνική SHAP ώστε να εντοπίσει τα σημαντικότερα χαρακτηριστικά που υποστήριξαν την πρόβλεψη για μία συγκεκριμένη επίθεση. Επίσης δίνεται έμφαση στην απομόνωση των χαρακτηριστικών που έχουν θετική συνεισφορά στην κατηγορία της επίθεσης, διασφαλίζοντας ότι ο χρήστης βλέπει μόνο τα στοιχεία που ενίσχυσαν την υποψία κακόβουλης δραστηριότητας και όχι αυτά που την αποδυνάμωσαν**.** Για να επιτευχθεί αυτό ο αλγόριθμος φιλτράρει τις τιμές SHAP και διατηρεί μόνο τα θετικά ταξινομώντας τα από τη μεγαλύτερη προς τη μικρότερη συνεισφορά. Έτσι η εξήγηση δεν στηρίζεται σε μια γενική υπόθεση για το είδος της επίθεσης, αλλά στα ίδια τα χαρακτηριστικά της συγκεκριμένης ροής που συνέβαλαν στην τελική απόφαση. Σε περίπτωση που η πληροφορία SHAP δεν είναι διαθέσιμη, το σύστημα έχει σχεδιαστεί με τέτοιο τρόπο ώστε να χρησιμοποιεί έναν εναλλακτικό μηχανισμό ερμηνείας, ο οποίος βασίζεται σε προκαθορισμένα προφίλ χαρακτηριστικών ανά οικογένεια επίθεσης.

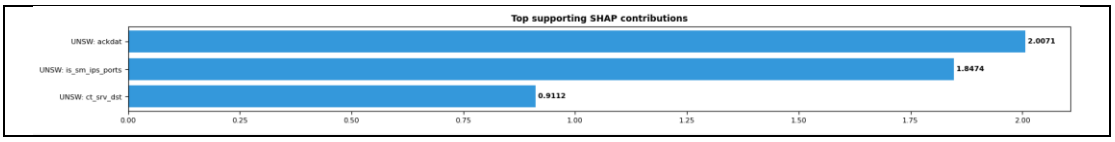
Τα προκαθορισμένα προφίλ XAI έχουν οριστεί για διαφορετικές κατηγορίες κακόβουλης δραστηριότητας, όπως DoS, PortScan, Web Attacks, Exploits, Fuzzers, Backdoors, Analysis, Shellcode, Worms, Reconnaissance και Generic. Κάθε προφίλ περιλαμβάνει έναν τίτλο που αποδίδει συνοπτικά το είδος της συμπεριφοράς που αναγνωρίστηκε, καθώς και ένα σύνολο χαρακτηριστικών της ροής, τα οποία ανήκουν στα χαρακτηριστικά που αξιοποιεί το μοντέλο. Έτσι το σύστημα παρέχει μια κατανοητή ερμηνεία ακόμη και στις

περιπτώσεις όπου δεν είναι εφικτή η απευθείας εξαγωγή επεξηγήσεων από τη διαδικασία SHAP.

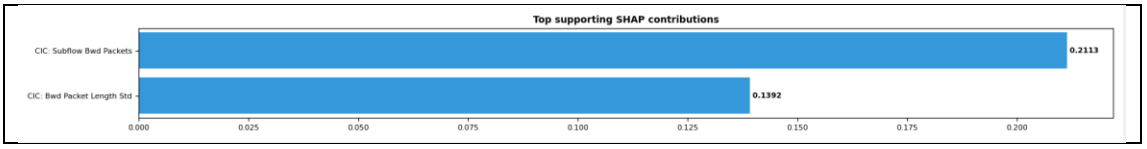
Η διαδικασία ερμηνείας ξεκινά μόλις προσδιοριστεί η οικογένεια της επίθεσης. Το σύστημα αξιοποιεί τα αποτελέσματα των μοντέλων CIC-IDS2017 και UNSW-NB15 και με βάση τη μορφή της τελικής ανίχνευσης αποδίδει το περιστατικό στην κατάλληλη κατηγορία συμπεριφοράς. Όταν και τα δύο μοντέλα ανιχνεύσουν κακόβουλη κίνηση το σύστημα δίνει προτεραιότητα στις πιο συγκεκριμένες κατηγορίες του CIC-IDS2017, ενώ αν η ανίχνευση προέρχεται από το UNSW-NB15, πραγματοποιείται αυτόματη κατηγοριοποίηση (mapping) των κλάσεων σε ευρύτερες οικογένειες επιθέσεων. Στη συνέχεια, επιχειρείται η ανάκτηση των σημαντικότερων χαρακτηριστικών που υποστήριξαν την πρόβλεψη. Όταν υπάρχει SHAP ερμηνεία προβάλλονται (όπως αποτυπώνει η Εικόνα 238, Εικόνα 239, Εικόνα 240, Εικόνα 241, Εικόνα 243, Εικόνα 244, Εικόνα 245 καί Εικόνα 246) τα σημαντικότερα χαρακτηριστικά με βάση τη συμβολή τους στην απόφαση του μοντέλου. Όταν αυτό δεν είναι δυνατό, ανακτάται το αντίστοιχο προφίλ XAI και εξετάζονται τα χαρακτηριστικά που έχουν οριστεί για τη συγκεκριμένη κατηγορία επίθεσης (Εικόνα 242). Από αυτά επιλέγονται όσα διαθέτουν χρήσιμη τιμή ώστε η τελική εξήγηση να παραμένει σύντομη. Με αυτό το τρόπο η λειτουργία Explainability έχει συγκεκριμένο χαρακτήρα και δεν περιορίζεται σε μια γενική αναφορά ότι μια ροή χαρακτηρίστηκε ως κακόβουλη.

Τα χαρακτηριστικά που προκύπτουν από αυτή τη διαδικασία παρουσιάζονται και γραφικά μέσω οριζόντιου ραβδογράμματος. Στις περιπτώσεις όπου υπάρχει διαθέσιμη SHAP ερμηνεία, το ραβδόγραμμα απεικονίζει τις σημαντικότερες συνεισφορές των χαρακτηριστικών στην τελική πρόβλεψη επιτρέποντας στον χρήστη να δει ποια από αυτά ενίσχυσαν περισσότερο την κατηγορία που αποδόθηκε στη ροή. Πάνω από κάθε ράβδο του γραφήματος εμφανίζεται η ακριβής τιμή SHAP του αντίστοιχου χαρακτηριστικού, συμβάλλοντας στην κατανόηση της συνεισφοράς του. Επίσης, δίπλα σε κάθε χαρακτηριστικό εμφανίζεται και ένδειξη της προέλευσής του, όπως CIC ή UNSW, ώστε να είναι σαφές από ποιο μοντέλο προέρχεται η συγκεκριμένη πληροφορία. Με τον τρόπο αυτό διευκολύνεται η ερμηνεία των αποτελεσμάτων, ιδιαίτερα στις περιπτώσεις όπου τα δύο μοντέλα οδηγούνται σε διαφορετική αναγνώριση ή στηρίζουν την απόφασή τους σε διαφορετικά χαρακτηριστικά. Όταν η SHAP ανάλυση δεν είναι διαθέσιμη το γράφημα εμφανίζει τα βασικά χαρακτηριστικά του επιλεγμένου προφίλ με βάση τις τιμές τους στη συγκεκριμένη ροή. Έτσι συμπληρώνει τη λεκτική περιγραφή και βοηθά τον τεχνικό

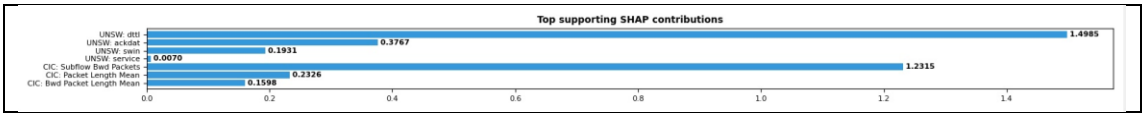
ασφαλείας να αντιληφθεί άμεσα ποια στοιχεία της δικτυακής δραστηριότητας συνδέονται με την ερμηνεία του περιστατικού.



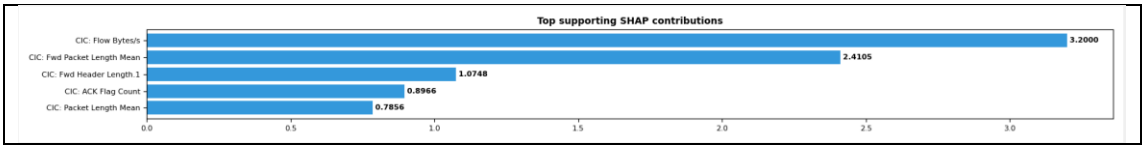
Εικόνα 238 Απεικόνιση τιμών Shap για την επίθεση τύπου Exploits μέσω ραβδογράμματος



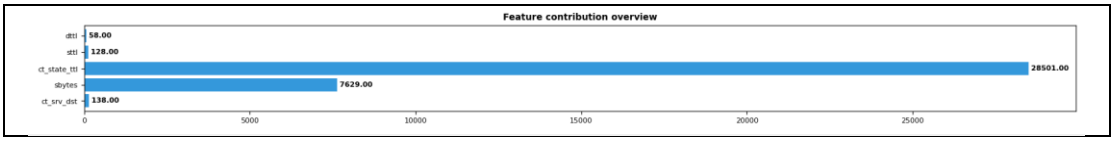
Εικόνα 239 Απεικόνιση τιμών Shap για την επίθεση τύπου Dos μέσω ραβδογράμματος



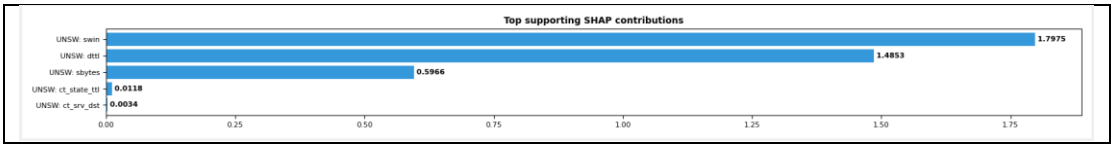
Εικόνα 240 Απεικόνιση τιμών Shap για την επίθεση τύπου Exploits για το UNSW-NB15 και τύπου Dos για το CIC-IDS2017



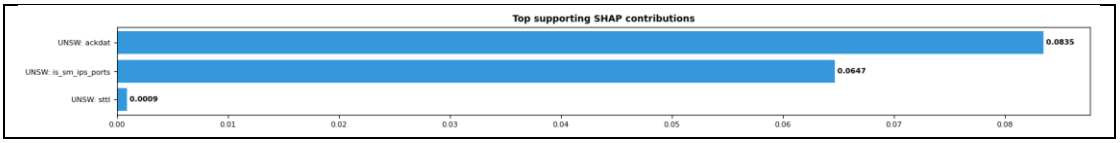
Εικόνα 241 Απεικόνιση τιμών Shap για την επίθεση τύπου Port Scan μέσω ραβδογράμματος



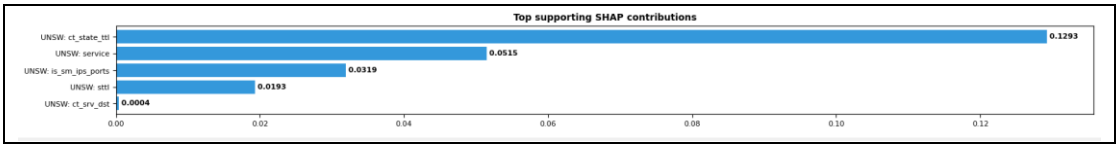
Εικόνα 242 Απεικόνιση τιμών XAI προφίλ για την επίθεση τύπου Generic μέσω ραβδογράμματος



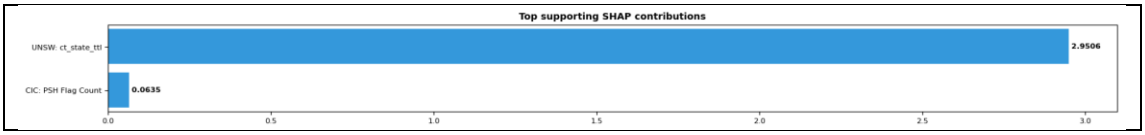
Εικόνα 243 Απεικόνιση τιμών Shap για την επίθεση τύπου Backdoor μέσω ραβδογράμματος



Εικόνα 244 Απεικόνιση τιμών Shap για την επίθεση τύπου Fuzzers μέσω ραβδογράμματος



Εικόνα 245 Απεικόνιση τιμών Shap για την επίθεση τύπου Reconnaissance μέσω ραβδογράμματος



Εικόνα 246 Απεικόνιση τιμών Shap για την επίθεση τύπου Web Attack μέσω ραβδογράμματος

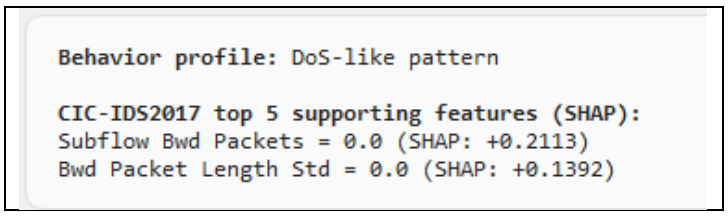
Τα χαρακτηριστικά που συνδέονται με την τελική ερμηνεία του περιστατικού προβάλλονται επιπλέον και σε ξεχωριστή κάρτα μέσα στο XAI Analysis. Στην κάρτα αυτή εμφανίζεται και το αντίστοιχο προφίλ συμπεριφοράς (behavior profile), το οποίο αντιστοιχεί στην οικογένεια της επίθεσης που εξετάζεται. Όταν είναι διαθέσιμη η τεχνική SHAP εμφανίζονται τα σημαντικότερα χαρακτηριστικά που υποστήριξαν την προβλεπόμενη κατηγορία του μοντέλου συνοδευόμενα από την τιμή συμβολής τους στην τελική απόφαση (Εικόνα 247, Εικόνα 248, Εικόνα 249, Εικόνα 250, Εικόνα 252, Εικόνα 253, Εικόνα 254). Για κάθε χαρακτηριστικό εμφανίζεται η πραγματική τιμή του στη ροή καθώς και η αντίστοιχη τιμή SHAP δίνοντας στον χρήστη όλη την σχετική πληροφορία. Όταν όμως δεν είναι διαθέσιμη αυτή η πληροφορία η κάρτα εμφανίζει ένα σύνολο βασικών χαρακτηριστικών της ροής, τα οποία έχουν προκαθοριστεί και αντιστοιχιστεί στη συγκεκριμένη οικογένεια επίθεσης και λειτουργούν ως εναλλακτικός μηχανισμός ερμηνείας (όπως φαίνεται χαρακτηριστικά και στην Εικόνα 251).

Αυτά τα χαρακτηριστικά παρουσιάζονται σε συνοπτική μορφή μαζί με τις αντίστοιχες τιμές τους ώστε να φαίνεται ποια στοιχεία της δικτυακής ροής θεωρήθηκαν πιο σημαντικά για την ερμηνεία της ανιχνευμένης δραστηριότητας. Ειδικότερα, όταν υπάρχουν διαθέσιμες τιμές και επαρκής πληροφορία, εμφανίζονται έως και πέντε σημαντικότερα χαρακτηριστικά της ροής που σχετίζονται με την τελική ερμηνεία του περιστατικού. Εξαιτίας αυτής της λειτουργιάς και συνδυαστικά με τα ραβδογράμματα, είναι πιο εύκολο να κατανοηθεί ποια χαρακτηριστικά της επικοινωνίας συνδέθηκαν με το προφίλ της επίθεσης που εξετάζεται.

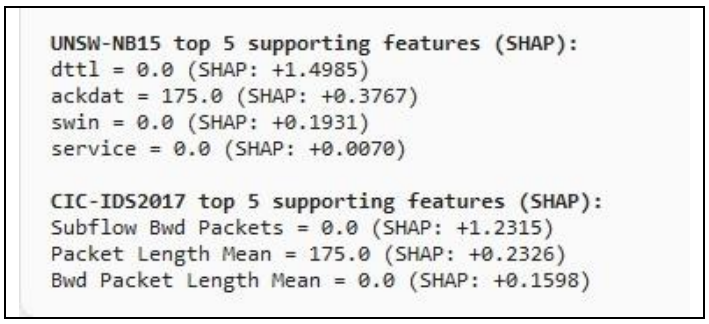
```
Behavior profile: Vulnerability Exploitation Indicators

UNSW-NB15 top 5 supporting features (SHAP):
ackdat = 0.0 (SHAP: +2.0071)
is_sm_ips_ports = 0.0 (SHAP: +1.8474)
ct_srv_dst = 2.0 (SHAP: +0.9112)
```

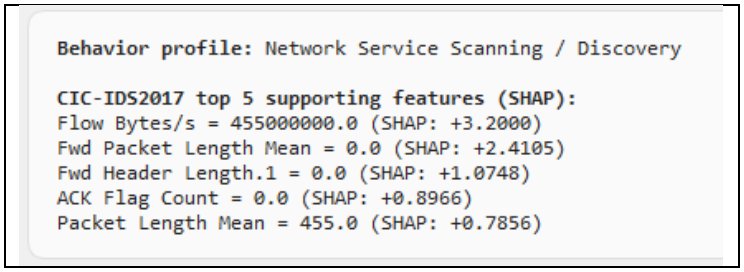
Εικόνα 247 Εμφάνιση τιμών Shap για την επίθεση τύπου Exploits.



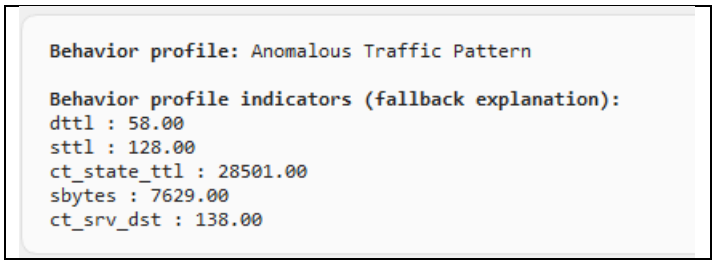
Εικόνα 248 Εμφάνιση τιμών Shap για την επίθεση τύπου Dos.



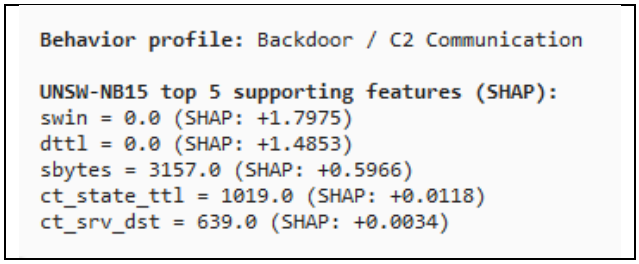
Εικόνα 249 Εμφάνιση τιμών Shap για την επίθεση τύπου Exploits για το UNSW-NB15 και τύπου Dos για το CIC-IDS2017



Εικόνα 250 Εμφάνιση τιμών Shap για την επίθεση τύπου PortScan.



Εικόνα 251 Εμφάνιση τιμών XAI προφίλ για την επίθεση τύπου Generic.



Εικόνα 252 Εμφάνιση τιμών Shap για την επίθεση τύπου Backdoor.

```
Behavior profile: Fuzzing / Abnormal Payload behavior

UNSW-NB15 top 5 supporting features (SHAP):
ackdat = 0.0 (SHAP: +0.0835)
is_sm_ips_ports = 1.0 (SHAP: +0.0647)
sttl = 254.0 (SHAP: +0.0009)
```

Εικόνα 253 Εμφάνιση τιμών Shap για την επίθεση τύπου Fuzzers.

```
Behavior profile: Network Scanning & Reconnaissance

UNSW-NB15 top 5 supporting features (SHAP):
ct_state_ttl = 4.0 (SHAP: +0.1293)
service = 0.0 (SHAP: +0.0515)
is_sm_ips_ports = 1.0 (SHAP: +0.0319)
sttl = 254.0 (SHAP: +0.0193)
ct_srv_dst = 1.0 (SHAP: +0.0004)
```

Εικόνα 254 Εμφάνιση τιμών Shap για την επίθεση τύπου Reconnaissance.

```
Behavior profile: Web-attack-like pattern

UNSW-NB15 top 5 supporting features (SHAP):
ct_state_ttl = 366220.0 (SHAP: +2.9506)

CIC-IDS2017 top 5 supporting features (SHAP):
PSH Flag Count = 8.0 (SHAP: +0.0635)
```

Εικόνα 255 Εμφάνιση τιμών Shap για την επίθεση τύπου Web Attack

7.5.3. Διασύνδεση με το MITRE ATT&CK: Αντιστοίχιση των ανιχνευμένων επιθέσεων με επίσημες τεχνικές και τακτικές κυβερνοεπιθέσεων

Μία από τις βασικές λειτουργίες του Explainability είναι η διασύνδεσή του με το MITRE ATT&CK, μια διεθνώς αναγνωρισμένη βάση η οποία οργανώνει και περιγράφει τις τακτικές και τις τεχνικές που χρησιμοποιούνται σε πραγματικές κυβερνοεπιθέσεις. Η λειτουργία αυτή έχει ως στόχο την ενίσχυση της ερμηνείας των ανιχνευμένων περιστατικών, αντιστοιχίζοντας κάθε κατηγορία επίθεσης με μια αναγνωρισμένη τακτική και τεχνική κυβερνοεπίθεσης. Έτσι το σύστημα δεν δείχνει μόνο τον τύπο της επίθεσης αλλά και μια βαθύτερη ανάλυση ώστε να ενισχυθεί η επεξηγησιμότητα.

Η αντιστοίχιση με το MITRE ATT&CK πραγματοποιείται μέσω μιας προκαθορισμένης συσχέτισης ανάμεσα στην οικογένεια της επίθεσης και στα αντίστοιχα στοιχεία του MITRE ATT&CK. Συγκεκριμένα, για κάθε οικογένεια επίθεσης έχει οριστεί μια δομή δεδομένων η οποία περιλαμβάνει την τακτική, την τεχνική, το αναγνωριστικό της τεχνικής και τον σύνδεσμο προς την επίσημη καταχώριση της τεχνικής στον ιστότοπο του MITRE ATT&CK.

Η διαδικασία ενεργοποιείται αφού πρώτα προσδιοριστεί η οικογένεια της επίθεσης από τα μοντέλα ανίχνευσης. Στη συνέχεια, το σύστημα αναζητά το αντίστοιχο πλαίσιο MITRE ATT&CK και εφόσον υπάρχει διαθέσιμη αντιστοίχιση, εμφανίζει στον χρήστη τη σχετική πληροφορία (Εικόνα 256). Η πληροφορία αυτή περιλαμβάνει την τακτική στην οποία εντάσσεται η ανιχνευμένη δραστηριότητα, την τεχνική που θεωρείται περισσότερο αντιπροσωπευτική, το αναγνωριστικό της τεχνικής και έναν σύνδεσμο προς την επίσημη σελίδα τεκμηρίωσης. Έτσι ο χρήστης μπορεί να μεταβεί από την απλή, σε μια πλήρη ερμηνεία της κακόβουλης ανίχνευσης.

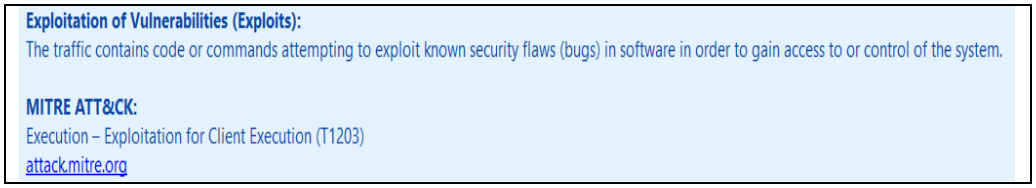
Οι συσχετίσεις γίνονται ως εξής: Οι επιθέσεις τύπου DoS αντιστοιχίζονται στην τακτική Impact και στην τεχνική Άρνηση Υπηρεσίας Δικτύου -Network Denial of Service (T1498), οι επιθέσεις τύπου PortScan αντιστοιχίζονται στην τακτική Discovery και στην τεχνική Σάρωση υπηρεσίας δικτύου - Network Service Scanning (T1046), οι επιθέσεις Web Attacks συνδέονται με την τακτική Initial Access και την τεχνική Εκμετάλλευση εφαρμογής που απευθύνεται στο κοινό - Exploit Public Facing Application (T1190), ενώ οι επιθέσεις τύπου Exploits συσχετίζονται με την τακτική Execution και την τεχνική Εκμετάλλευση για την Εκτέλεση Πελατών - Exploitation for Client Execution (T1203). Έτσι η πληροφορία που εμφανίζει το Shield IDS συνδέεται με ένα τυποποιημένο λεξιλόγιο περιγραφής απειλών.

Προκειμένου η παραπάνω πληροφορία να είναι προσβάσιμη, όχι μόνο σε τεχνικούς ασφαλείας αλλά και σε λιγότερο έμπειρους χρήστες, ενσωματώθηκε ένα ειδικό πλαίσιο επεξήγησης για αρχαρίους (Εικόνα 257). Σε αυτό το τμήμα της καρτέλας, το σύστημα εμφανίζει την ορολογία της επίθεσης με απλό και κατανοητό τρόπο.

Τέλος η διασύνδεση αυτή παρουσιάζεται απευθείας στην καρτέλα Explainability, ώστε ο χρήστης να βλέπει σε ενιαίο περιβάλλον τόσο την κατηγορία της επίθεσης όσο και το αντίστοιχο πλαίσιο MITRE ATT&CK. Το σύστημα απεικονίζει τα πεδία Tactic, Technique και το αναγνωριστικό της τεχνικής, ενώ παράλληλα παρέχεται σύνδεσμος προς την επίσημη σελίδα του MITRE ATT&CK. Σχεδιάστηκε και υλοποιήθηκε με αυτό τον τρόπο έτσι ώστε να μην απαιτείται χειροκίνητη αναζήτηση της τεχνικής που αντιστοιχεί στην επίθεση αλλά να λαμβάνεται άμεσα η πληροφορία.



Εικόνα 256 MITRE ATT&CK περιεχόμενο για ανιχνευθείσα επίθεση τύπου Exploits.



Εικόνα 257 Περιγραφή της επίθεσης Exploits για αρχάριους χρήστες και αντιστοίχισή της με το MITRE ATT&CK.

7.5.4. Σύγκριση Μοντέλων: Απεικόνιση της συμφωνίας/διαφωνίας μεταξύ CIC-IDS και UNSW-NB15 μοντέλων

Μία ακόμη σημαντική λειτουργία είναι η σύγκριση των αποφάσεων των δύο μοντέλων ανίχνευσης που αξιοποιούνται από το σύστημα, δηλαδή του CIC-IDS2017 και του UNSW-NB15. Σκοπός της λειτουργίας αυτής είναι να παρέχεται μια πιο ολοκληρωμένη εικόνα του περιστατικού, προβάλλοντας όχι μόνο το τελικό αποτέλεσμα της ανίχνευσης, αλλά και την επιμέρους απόφαση κάθε μοντέλου. Έτσι γίνετε αντιληπτό αν τα δύο μοντέλα κατέληξαν στο ίδιο συμπέρασμα ή αν υπήρξε διαφοροποίηση της αξιολόγησης.

Η πληροφορία αυτή απεικονίζεται μέσα από την ενότητα μοντέλα αποφάσεων (Model decisions). Εκεί εμφανίζεται ξεχωριστά η πρόβλεψη του μοντέλου CIC-IDS2017 και ξεχωριστά η πρόβλεψη του μοντέλου UNSW-NB15 (Εικόνα 258). Για το CIC-IDS2017 η έξοδος αποτυπώνεται είτε ως BENIGN είτε ως συγκεκριμένη κατηγορία επίθεσης, ενώ για το UNSW-NB15 η έξοδος αποτυπώνεται είτε ως Normal είτε ως αντίστοιχη κατηγορία κακόβουλης δραστηριότητας (όπως για παράδειγμα Exploits, Fuzzers ή DoS). Η ταυτόχρονη εμφάνιση των δύο προβλέψεων επιτρέπει την άμεση σύγκριση των αποτελεσμάτων των μοντέλων πάνω στην ίδια ροή.

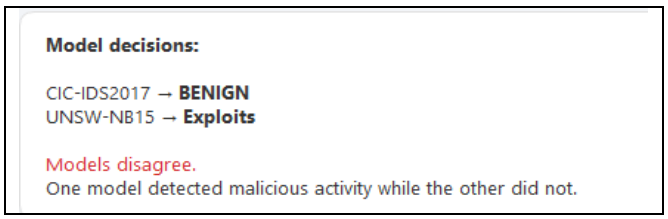
Η σύγκριση αυτή βασίζεται στο αν κάθε μοντέλο θεωρεί τη ροή κανονική ή κακόβουλη. Πιο συγκεκριμένα, το σύστημα ελέγχει αν το αποτέλεσμα του CIC-IDS2017 είναι διαφορετικό από BENIGN και αν το αποτέλεσμα του UNSW-NB15 είναι διαφορετικό από Normal. Εφόσον και τα δύο μοντέλα κατατάσσουν τη ροή στην ίδια γενική κατηγορία, δηλαδή είτε και τα δύο ως φυσιολογική είτε και τα δύο ως κακόβουλη τότε το σύστημα εμφανίζει ότι τα μοντέλα συμφωνούν. Αντίθετα, όταν το ένα μοντέλο χαρακτηρίζει τη ροή ως κακόβουλη και το άλλο ως φυσιολογική εμφανίζεται ένδειξη ότι τα μοντέλα διαφωνούν (Εικόνα 258). Σε περιπτώσεις όπου τα δύο μοντέλα συμφωνούν ότι η ροή είναι κακόβουλη αλλά διαφωνούν ως προς την κατηγορία της επίθεσης, το σύστημα ενημερώνει τον χρήστη ότι και τα δύο μοντέλα ανίχνευσαν κακόβουλη δραστηριότητα χωρίς όμως να καταλήγουν στον ίδιο χαρακτηρισμό της επίθεσης (Εικόνα 260).

Αυτό είναι ιδιαίτερα σημαντικό διότι επιτρέπει στον χρήστη να αξιολογήσει τον βαθμό συνέπειας της ανίχνευσης. Όταν τα δύο μοντέλα συμφωνούν η απόφαση του συστήματος παρουσιάζει μεγαλύτερη συνέπεια καθώς προκύπτει από δύο ανεξάρτητες προσεγγίσεις ανάλυσης της ίδιας ροής. Αντίθετα όταν υπάρχει διαφωνία, αποκτάται μια σημαντική ένδειξη ότι το περιστατικό χρειάζεται προσεκτικότερη διερεύνηση, επειδή η ίδια δικτυακή συμπεριφορά ερμηνεύτηκε διαφορετικά από τα δύο μοντέλα.

Η διαφωνία αυτή μπορεί να προκύψει επειδή τα δύο μοντέλα έχουν εκπαιδευτεί σε διαφορετικά σύνολα δεδομένων και βασίζονται σε διαφορετικά χαρακτηριστικά και πρότυπα επιθετικής συμπεριφοράς. Το CIC-IDS2017 χρησιμοποιείται στο Shield IDS κυρίως για συγκεκριμένες οικογένειες επιθέσεων, όπως DoS, Web Attacks και PortScan, ενώ το UNSW-NB15 χρησιμοποιείται για ταξινόμηση σε πολλαπλές κατηγορίες επιθέσεων που μπορεί να αποδώσει ευρύτερο φάσμα επιθετικών δραστηριοτήτων. Επομένως είναι λογικό σε ορισμένες περιπτώσεις ένα μοντέλο να εμφανίζει μεγαλύτερη ευαισθησία σε ένα συγκεκριμένο μοτίβο κίνησης σε σχέση με το άλλο.

Όπως φαίνεται και στο παράδειγμα στην Εικόνα 258, το CIC-IDS2017 κατατάσσει μια ροή ως BENIGN, ενώ το UNSW-NB15 την ταξινομεί ως Exploits. Σε αυτή την περίπτωση εμφανίζεται η ένδειξη ότι τα μοντέλα διαφωνούν με μήνυμα που αναφέρει ότι το ένα μοντέλο εντόπισε κακόβουλη δραστηριότητα ενώ το άλλο όχι. Η παρουσίαση αυτή είναι ιδιαίτερα χρήσιμη, επειδή δεν αποκρύπτει τη διαφοροποίηση ανάμεσα στα δύο μοντέλα αλλά την απεικονίζει ως μέρος της επεξηγηματικής ανάλυσης. Στην περίπτωση που η κίνηση είναι κανονική εμφανίζεται σχετικό μήνυμα που ενημερώνει τον χρήστη (όπως χαρακτηριστικά φαίνεται στην Εικόνα 259).

Αξίζει να σημειωθεί ότι η τελική συνδυασμένη πρόβλεψη είναι επιθετική ως προς την ανίχνευση. Δηλαδή, εφόσον έστω ένα από τα δύο μοντέλα χαρακτηρίσει τη ροή ως κακόβουλη, η τελική πρόβλεψη του συστήματος θεωρεί τη ροή κακόβουλη. Αυτό διασφαλίζει ότι το σύστημα ελαχιστοποιεί την πιθανότητα ψευδώς αρνητικών αποτελεσμάτων (False Negatives), προτιμώντας να προειδοποιήσει τον χρήστη για μια ύποπτη ροή ακόμα και αν υπάρχει διαφωνία μεταξύ των μοντέλων.



Εικόνα 258 Παράδειγμα διαφωνίας μεταξύ των μοντέλων CIC-IDS2017 και UNSW-NB15



Εικόνα 259 Απεικόνιση όταν η κίνηση είναι φυσιολογική



Εικόνα 260 Απεικόνιση όταν έχουμε διαφορετικό είδος ανίχνευσης επίθεσης

7.6. Δημιουργία και διαχείριση αναφορών (Alerts & Reports)

Η ενότητα Alerts & Reports του Shield IDS (στιγμιότυπα της οποίας αποτυπώνει η Εικόνα 261) σχεδιάστηκε έτσι ώστε να συγκεντρώνει και να παρουσιάζει την κίνηση που έχει χαρακτηριστεί ως κακόβουλη από το σύστημα. Σε αντίθεση με την ενότητα Live Detection η οποία περιλαμβάνει τόσο κανονική όσο και κακόβουλη κίνηση, η συγκεκριμένη καρτέλα επικεντρώνεται αποκλειστικά στα περιστατικά που έχουν αναγνωριστεί ως επιθέσεις. Έτσι ο τεχνικός ασφαλείας έχει άμεση πρόσβαση σε μια καθαρή προβολή των σημαντικότερων συμβάντων, χωρίς να επιβαρύνεται με τις εγγραφές της κανονικής δικτυακής δραστηριότητας.

Η δημιουργία των εγγραφών βασίζεται στα δεδομένα που έχουν ήδη παραχθεί από τη διαδικασία ανίχνευσης και έχουν αποθηκευτεί στο ιστορικό του συστήματος. Κατά την ενημέρωση της καρτέλας, το Shield IDS συγχρονίζει πρώτα τα πιο πρόσφατα δεδομένα, από τον προσωρινό χώρο αποθήκευσης (buffer), στο κεντρικό σύνολο καταγραφών και στη συνέχεια φιλτράρει μόνο εκείνες τις εγγραφές που περιλαμβάνουν ένδειξη επίθεσης. Δηλαδή το σύστημα διατηρεί στην καρτέλα Alerts & Reports μόνο τις ροές για τις οποίες έχει παραχθεί τελική ένδειξη επίθεσης από το CIC-IDS2017, από το UNSW-NB15 ή από συνδυασμό των δύο. Έτσι η συγκεκριμένη καρτέλα ενημερώνεται συνεχώς όσο το σύστημα βρίσκεται σε λειτουργία.

Η διεπαφή χρήστη οργανώνεται γύρω από έναν πίνακα, στον οποίο παρουσιάζονται τα περιστατικά που έχουν ανιχνευθεί σε μορφή γραμμών. Κάθε γραμμή αντιστοιχεί σε μία ροή που έχει χαρακτηριστεί ως κακόβουλη και συνοδεύεται από τα χαρακτηριστικά της. Οι εγγραφές στον πίνακα εμφανίζονται με κόκκινο χρώμα ώστε να διευκολύνεται η άμεση αναγνώριση της επικινδυνότητας των ροών. Στο πάνω μέρος της σελίδας εμφανίζεται ο συνολικός αριθμός των επιθέσεων που έχουν καταγραφεί, ώστε ο χρήστης να μπορεί να αποκτήσει άμεσα εικόνα του όγκου των περιστατικών. Η πληροφορία αυτή ανανεώνεται αυτόματα απεικονίζοντας τις νέες ανιχνεύσεις που παράγονται κατά τη διάρκεια της λειτουργίας της εφαρμογής.

Ο πίνακας απεικονίζει πεδία καταγραφής της ροής, όπως τα Timestamp, detect_ts, flow_start_ts και detection_delay, τα οποία περιγράφουν τη χρονική στιγμή της κακόβουλης κίνησης, τον χρόνο ανίχνευσης, την έναρξη της ροής και την καθυστέρηση ανίχνευσης. Επίσης εμφανίζονται τα βασικά πεδία ταυτοποίησης της επικοινωνίας, όπως srcip, sport, dstip, dsport, proto, service και state, τα οποία αντιστοιχούν στις διευθύνσεις IP πηγής και προορισμού, στις αντίστοιχες θύρες, στο πρωτόκολλο, στην υπηρεσία και στην κατάσταση της σύνδεσης.

Επιπλέον, εμφανίζονται χαρακτηριστικά που σχετίζονται με την συμπεριφορά της δικτυακής κίνησης, όπως τα sbytes, dbytes, ackdat, tcprtt, sloss, dloss, Flow Bytes/s, Sload, Dload, Dpkts, total_bytes, καθώς και πλήθος στατιστικών χαρακτηριστικών της ροής, όπως Average Packet Size, Packet Length Mean, Packet Length Std, Max Packet Length, Min Packet Length, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Packet Length Mean, Bwd Packet Length Max, Bwd Packet Length Mean, Bwd Packet Length Std, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Subflow Fwd Packets, Subflow Fwd bytes, Subflow Bwd Packets, Subflow Bwd Bytes, act_data_pkt_fwd, Active Min, Active Mean, Active Std, Idle Mean, Fwd IAT Min, Fwd IAT Std, Bwd IAT Mean, Bwd IAT Std και Fwd Header Length.1.

Ο πίνακας περιλαμβάνει επίσης χαρακτηριστικά που σχετίζονται με τις σημαίες TCP και μετρικές της σύνδεσης, όπως ACK Flag Count, FIN Flag Count, PSH Flag Count, SYN Flag Count και URG Flag Count, καθώς και παραμέτρους του TCP πρωτοκόλλου όπως Init_Win_bytes_forward, Init_Win_bytes_backward, swin, dwin, smeansz, dmeansz και synack. Παράλληλα απεικονίζονται χαρακτηριστικά που προέρχονται από το προφίλ του συνόλου δεδομένων UNSW-NB15, όπως sttl, dttl, ct_state_ttl, ct_srv_dst, ct_srv_src,

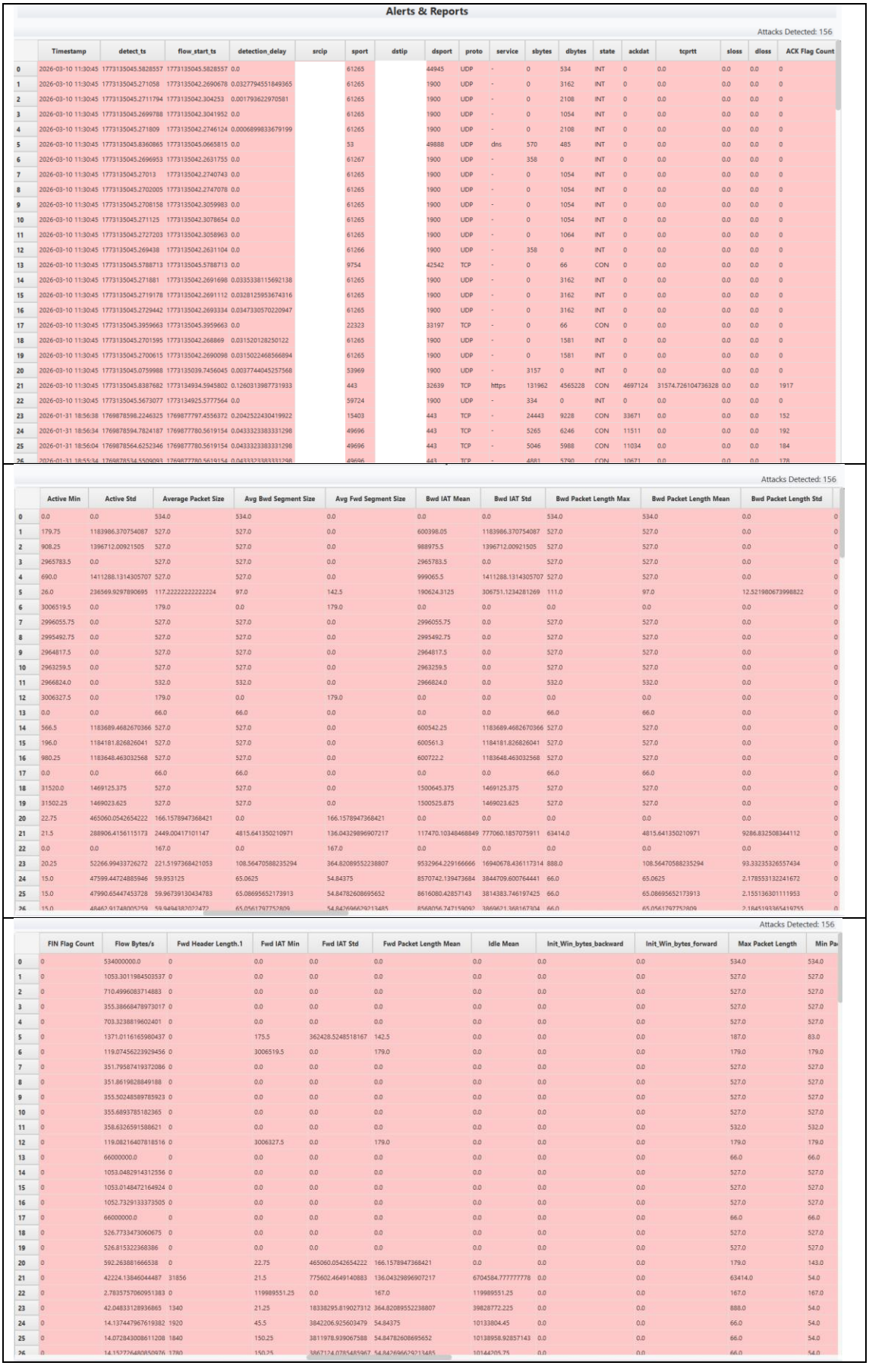
ct_dst_src_ltm, ct_dst_sport_ltm, ct_src_dport_ltm, ct_ftp_cmd και is_sm_ips_ports, τα οποία περιγράφουν συσχετισμούς υπηρεσιών, συμπεριφορά θυρών και χαρακτηριστικά της δικτυακής επικοινωνίας.

Στον πίνακα εμφανίζονται και τα πεδία που σχετίζονται με τα αποτελέσματα των μοντέλων ανίχνευσης και την τελική ερμηνεία του περιστατικού, όπως attack_cic, pred_unsw, attack_unsw, prediction, attack_type, attack_merged, color_tag και remote_ip, τα οποία προβάλλουν τις προβλέψεις των μοντέλων, την τελική κατηγορία επίθεσης που αποδίδεται στη ροή και βοηθητικές πληροφορίες που χρησιμοποιούνται για την οπτική επισήμανση και οργάνωση των εγγραφών.

Οι εγγραφές ταξινομούνται χρονικά, ώστε οι πιο πρόσφατες επιθέσεις να εμφανίζονται πρώτες, επιτρέποντας στον χρήστη να εστιάζει άμεσα στα νεότερα συμβάντα. Επίσης κάθε γραμμή του πίνακα είναι διαδραστική. Με την επιλογή μιας εγγραφής το αντίστοιχο συμβάν προωθείται στην καρτέλα Explainability, όπου παρουσιάζεται αναλυτικά η ερμηνεία της απόφασης, το προφίλ συμπεριφοράς της επίθεσης, η διασύνδεσή της με το MITRE ATT&CK και η σύγκριση των αποτελεσμάτων των μοντέλων. Έτσι η καρτέλα Alerts & Reports δεν απεικονίζει απλά την κακόβουλη δικτυακή κίνηση αλλά βοηθάει στην περαιτέρω διερεύνηση.

Ένα σημαντικό χαρακτηριστικό στην λειτουργία της καρτέλας είναι η αυτόματη ανανέωσή της. Στην υλοποίηση του Shield IDS, η σελίδα Alerts & Reports ανανεώνεται όταν είναι ενεργή στην οθόνη, ώστε να εμφανίζει τα τελευταία διαθέσιμα περιστατικά χωρίς να απαιτείται χειροκίνητη παρέμβαση. Έτσι με αυτό τον τρόπο η ενότητα παραμένει συνεχώς ενημερωμένη και μπορεί να χρησιμοποιηθεί ως ζωντανός πίνακας παρακολούθησης των επιθέσεων που έχουν ανιχνευθεί.

Επιπλέον οι εγγραφές που εμφανίζονται στην καρτέλα βασίζονται στην τελική ετικέτα επίθεσης (attack_merged), η οποία προκύπτει από τον συνδυασμό των αποτελεσμάτων των δύο μοντέλων ανίχνευσης. Αυτό σημαίνει ότι κάθε περιστατικό παρουσιάζεται με κατανοητό τρόπο ακόμη και όταν η ανίχνευση προέρχεται από διαφορετικό τύπο μοντέλου, επιτρέποντας έτσι την γρήγορη και άμεση ερμηνεία των εγγραφών του πίνακα. Σε περίπτωση που δεν υπάρχουν ανιχνευμένες επιθέσεις, η καρτέλα εμφανίζει σχετικό ενημερωτικό μήνυμα.



Alerts & Reports

Attacks Detected: 156

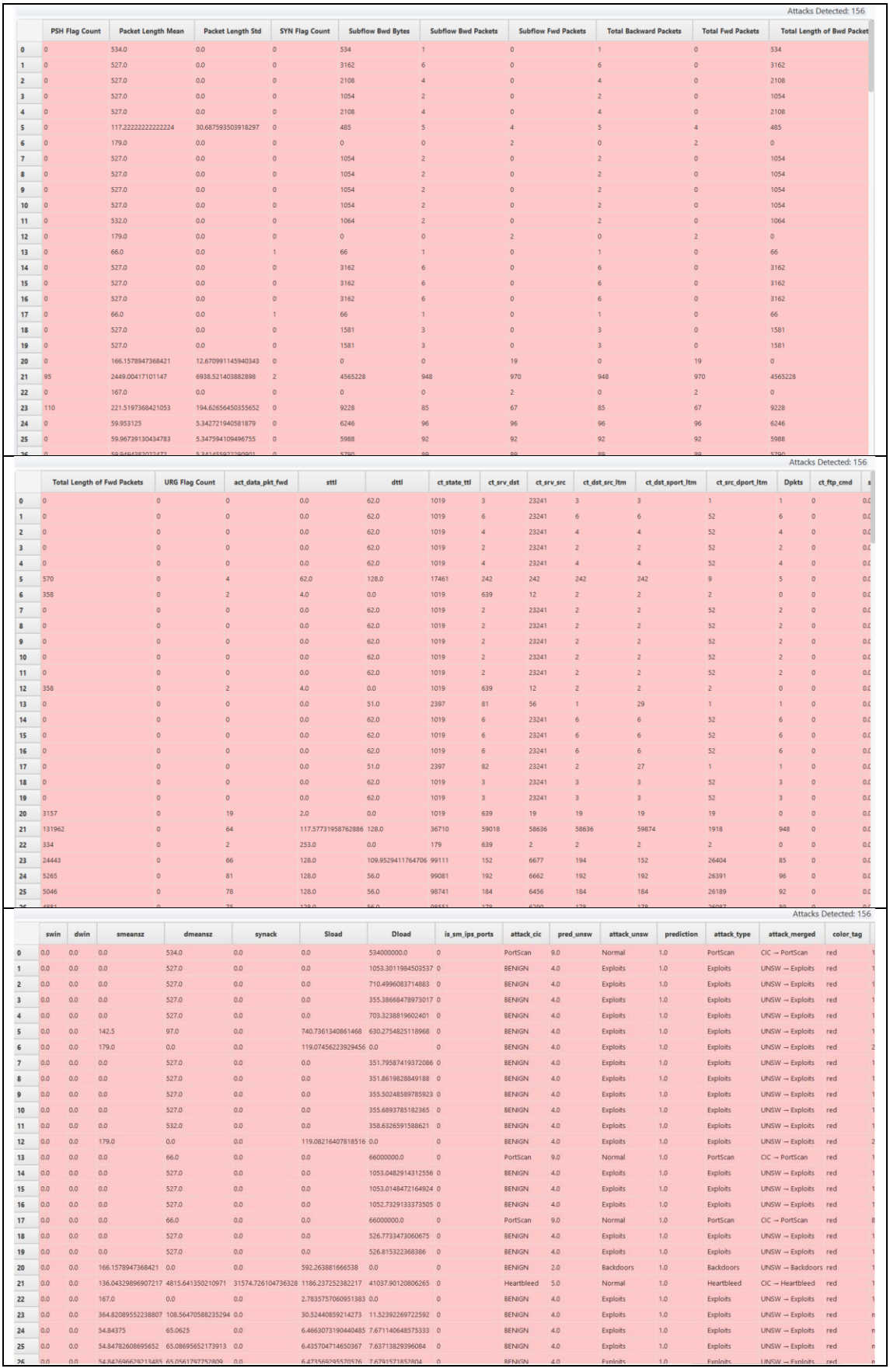
	Timestamp	detect_ts	flow_start_ts	detection_delay	srcip	sport	dstip	dsport	proto	service	sbytes	dbytes	state	ackdat	tcprtt	sloss	dloss	ACK Flag Count
0	2026-03-10 11:30:45	1773135045.5828557	1773135045.5828557	0.0		61265		44945	UDP	-	0	534	INT	0	0.0	0.0	0.0	0
1	2026-03-10 11:30:45	1773135045.271058	1773135042.2690678	0.032779455184936		61265		1900	UDP	-	0	3162	INT	0	0.0	0.0	0.0	0
2	2026-03-10 11:30:45	1773135045.2711794	1773135042.304253	0.001793622970581		61265		1900	UDP	-	0	2108	INT	0	0.0	0.0	0.0	0
3	2026-03-10 11:30:45	1773135045.2699788	1773135042.3041952	0.0		61265		1900	UDP	-	0	1054	INT	0	0.0	0.0	0.0	0
4	2026-03-10 11:30:45	1773135045.271809	1773135042.2746124	0.000689983679199		61265		1900	UDP	-	0	2108	INT	0	0.0	0.0	0.0	0
5	2026-03-10 11:30:45	1773135045.8360865	1773135045.0665815	0.0		53		49888	UDP	dns	570	485	INT	0	0.0	0.0	0.0	0
6	2026-03-10 11:30:45	1773135045.2696953	1773135042.2631755	0.0		61267		1900	UDP	-	358	0	INT	0	0.0	0.0	0.0	0
7	2026-03-10 11:30:45	1773135045.27013	1773135042.2740743	0.0		61265		1900	UDP	-	0	1054	INT	0	0.0	0.0	0.0	0
8	2026-03-10 11:30:45	1773135045.2702005	1773135042.2747078	0.0		61265		1900	UDP	-	0	1054	INT	0	0.0	0.0	0.0	0
9	2026-03-10 11:30:45	1773135045.2708158	1773135042.3059983	0.0		61265		1900	UDP	-	0	1054	INT	0	0.0	0.0	0.0	0
10	2026-03-10 11:30:45	1773135045.271125	1773135042.3078654	0.0		61265		1900	UDP	-	0	1054	INT	0	0.0	0.0	0.0	0
11	2026-03-10 11:30:45	1773135045.2727203	1773135042.3058963	0.0		61265		1900	UDP	-	0	1064	INT	0	0.0	0.0	0.0	0
12	2026-03-10 11:30:45	1773135045.269438	1773135042.2631104	0.0		61266		1900	UDP	-	358	0	INT	0	0.0	0.0	0.0	0
13	2026-03-10 11:30:45	1773135045.5788713	1773135045.5788713	0.0		9754		42542	TCP	-	0	66	CON	0	0.0	0.0	0.0	0
14	2026-03-10 11:30:45	1773135045.271881	1773135042.2691698	0.0335338115692138		61265		1900	UDP	-	0	3162	INT	0	0.0	0.0	0.0	0
15	2026-03-10 11:30:45	1773135045.2719178	1773135042.2691112	0.0328125953674316		61265		1900	UDP	-	0	3162	INT	0	0.0	0.0	0.0	0
16	2026-03-10 11:30:45	1773135045.2729442	1773135042.2693334	0.0347330570220947		61265		1900	UDP	-	0	3162	INT	0	0.0	0.0	0.0	0
17	2026-03-10 11:30:45	1773135045.3959663	1773135045.3959663	0.0		22323		33197	TCP	-	0	66	CON	0	0.0	0.0	0.0	0
18	2026-03-10 11:30:45	1773135045.2701595	1773135042.268869	0.0315201282501 22		61265		1900	UDP	-	0	1581	INT	0	0.0	0.0	0.0	0
19	2026-03-10 11:30:45	1773135045.2700615	1773135042.2690098	0.0315022468566894		61265		1900	UDP	-	0	1581	INT	0	0.0	0.0	0.0	0
20	2026-03-10 11:30:45	1773135045.0759988	1773135039.7456045	0.0037744045257568		53969		1900	UDP	-	3157	0	INT	0	0.0	0.0	0.0	0
21	2026-03-10 11:30:45	1773135045.8387682	1773134934.5945802	0.126031398731933		443		32639	TCP	https	131962	4565228	CON	4697124	31574.726104736328	0.0	0.0	1917
22	2026-03-10 11:30:45	1773135045.5673077	1773134925.5777564	0.0		59724		1900	UDP	-	334	0	INT	0	0.0	0.0	0.0	0
23	2026-01-31 18:56:38	1769878598.2246325	1769877797.4556372	0.204252243041992 2		15403		443	TCP	-	24443	9228	CON	33671	0.0	0.0	0.0	152
24	2026-01-31 18:56:34	1769878594.7824187	1769877780.5619154	0.043332338333129 8		49696		443	TCP	-	5265	6246	CON	11511	0.0	0.0	0.0	192
25	2026-01-31 18:56:04	1769878564.6252346	1769877780.5619154	0.043332338333129 8		49696		443	TCP	-	5046	5988	CON	11034	0.0	0.0	0.0	184
26	2026-01-31 18:55:34	1769878534.5509093	1769877780.5619154	0.043332338333129 8		49696		443	TCP	-	4881	5790	CON	10671	0.0	0.0	0.0	178

Attacks Detected: 156

	Active Min	Active Std	Average Packet Size	Avg Bwd Segment Size	Avg Fwd Segment Size	Bwd IAT Mean	Bwd IAT Std	Bwd Packet Length Max	Bwd Packet Length Mean	Bwd Packet Length Std	
0	0.0	0.0	534.0	534.0	0.0	0.0	0.0	534.0	534.0	0.0	0
1	179.75	1183986.370754087	527.0	527.0	0.0	600398.05	1183986.370754087	527.0	527.0	0.0	0
2	908.25	1396712.00921505	527.0	527.0	0.0	988975.5	1396712.00921505	527.0	527.0	0.0	0
3	2965783.5	0.0	527.0	527.0	0.0	2965783.5	0.0	527.0	527.0	0.0	0
4	690.0	1411288.1314305707	527.0	527.0	0.0	999065.5	1411288.1314305707	527.0	527.0	0.0	0
5	26.0	236569.9297890695	117.22222222222224	97.0	142.5	190624.3125	306751.1234281269	111.0	97.0	12.521980673998822	0
6	3006519.5	0.0	179.0	0.0	179.0	0.0	0.0	0.0	0.0	0.0	0
7	2996055.75	0.0	527.0	527.0	0.0	2996055.75	0.0	527.0	527.0	0.0	0
8	2995492.75	0.0	527.0	527.0	0.0	2995492.75	0.0	527.0	527.0	0.0	0
9	2964817.5	0.0	527.0	527.0	0.0	2964817.5	0.0	527.0	527.0	0.0	0
10	2963259.5	0.0	527.0	527.0	0.0	2963259.5	0.0	527.0	527.0	0.0	0
11	2966824.0	0.0	532.0	532.0	0.0	2966824.0	0.0	532.0	532.0	0.0	0
12	3006327.5	0.0	179.0	0.0	179.0	0.0	0.0	0.0	0.0	0.0	0
13	0.0	0.0	66.0	66.0	0.0	0.0	0.0	66.0	66.0	0.0	0
14	566.5	1183689.4682670366	527.0	527.0	0.0	600542.25	1183689.4682670366	527.0	527.0	0.0	0
15	196.0	1184181.826826041	527.0	527.0	0.0	600561.3	1184181.826826041	527.0	527.0	0.0	0
16	980.25	1183648.463032568	527.0	527.0	0.0	600722.2	1183648.463032568	527.0	527.0	0.0	0
17	0.0	0.0	66.0	66.0	0.0	0.0	0.0	66.0	66.0	0.0	0
18	31520.0	1469125.375	527.0	527.0	0.0	1500645.375	1469125.375	527.0	527.0	0.0	0
19	31502.25	1469023.625	527.0	527.0	0.0	1500525.875	1469023.625	527.0	527.0	0.0	0
20	22.75	465060.0542654222	166.1578947368421	0.0	166.1578947368421	0.0	0.0	0.0	0.0	0.0	0
21	21.5	288906.4156115173	2449.004171101147	4815.641350210971	136.04329896907217	117470.10348468849	777060.1857075911	63414.0	4815.641350210971	9286.832508344112	0
22	0.0	0.0	167.0	0.0	167.0	0.0	0.0	0.0	0.0	0.0	0
23	20.25	52266.99433726272	221.5197368421053	108.56470588235294	364.82089552238807	9532964.229166666	16940678.436117314	888.0	108.56470588235294	93.33235326557434	0
24	15.0	47599.44724885946	59.953125	65.0625	54.84375	8570742.139473684	3844709.600764441	66.0	65.0625	2.178553132241672	0
25	15.0	47990.65447453728	59.967391304347 83	65.08695652173913	54.84782608695652	8616080.42857143	3814383.746197425	66.0	65.08695652173913	2.155136301111953	0
26	15.0	48462.91748005359	59.949438202247 2	65.05617977752809	54.84769662921348 5	8560056.747159092	3869621.368167304	66.0	65.05617977752809	2.184519336549175 5	

Attacks Detected: 156

	FIN Flag Count	Flow Bytes/s	Fwd Header Length.1	Fwd IAT Min	Fwd IAT Std	Fwd Packet Length Mean	Idle Mean	Init_Win_bytes_backward	Init_Win_bytes_forward	Max Packet Length	Min Pa
0	0	534000000.0	0	0.0	0.0	0.0	0.0	0.0	0.0	534.0	534.0
1	0	1053.3011984503537	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
2	0	710.4996083714883	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
3	0	355.38668478973017	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
4	0	703.3238819602401	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
5	0	1371.0116165980437	0	175.5	362428.5248518167	142.5	0.0	0.0	0.0	187.0	83.0
6	0	119.07456223929456	0	3006519.5	0.0	179.0	0.0	0.0	0.0	179.0	179.0
7	0	351.79587419372086	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
8	0	351.8619828849188	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
9	0	355.50248589785923	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
10	0	355.6893785182365	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
11	0	358.63265915886 21	0	0.0	0.0	0.0	0.0	0.0	0.0	532.0	532.0
12	0	119.08216407818516	0	3006327.5	0.0	179.0	0.0	0.0	0.0	179.0	179.0
13	0	66000000.0	0	0.0	0.0	0.0	0.0	0.0	0.0	66.0	66.0
14	0	1053.0482914312556	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
15	0	1053.0148472164924	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
16	0	1052.7329133373505	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
17	0	66000000.0	0	0.0	0.0	0.0	0.0	0.0	0.0	66.0	66.0
18	0	526.7733473060675	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
19	0	526.815322368386	0	0.0	0.0	0.0	0.0	0.0	0.0	527.0	527.0
20	0	592.263881666538	0	22.75	465060.0542654222	166.1578947368421	0.0	0.0	0.0	179.0	143.0
21	0	42224.13846044487	31856	21.5	775602.4649140883	136.04329896907217	6704584.777777778	0.0	0.0	63414.0	54.0
22	0	2.7835757060951383	0	119989551.25	0.0	167.0	119989551.25	0.0	0.0	167.0	167.0
23	0	42.04833128936865	1340	21.25	18338295.819027312	364.82089552238807	39828772.225	0.0	0.0	888.0	54.0
24	0	14.137447967619382	1920	45.5	3842206.925603479	54.84375	10133804.45	0.0	0.0	66.0	54.0
25	0	14.072843008611208	1840	150.25	3811978.939067588	54.84782608695652	10138958.92857143	0.0	0.0	66.0	54.0
26	0	14.152726480950976	1780	150.25	3867124.078548596 7	54.84769662921348 5	10144205.75	0.0	0.0	66.0	54.0



Εικόνα 261 Καρτέλα Alerts & Reports

7.7. Ανάλυση καταγραφών (Logs Analysis)

Η καρτέλα Logs Analysis της εφαρμογής (όπως παρουσιάζει η Εικόνα 262) έχει σχεδιαστεί και υλοποιηθεί με σκοπό να παρέχει μια αναλυτική προβολή όλων των καταγραφών που παράγονται κατά τη λειτουργία του συστήματος. Εντός της ανάλυσης περιλαμβάνεται το συνολικό ιστορικό των ροών που έχουν καταγραφεί από την εφαρμογή (ενώ σε αντίθεση η καρτέλα Reports απεικονίζει αποκλειστικά στα περιστατικά που έχουν χαρακτηριστεί ως επιθέσεις). Έτσι ο τεχνικός ασφαλείας έχει τη δυνατότητα να εξετάζει όχι μόνο τα κακόβουλα συμβάντα, αλλά και τη φυσιολογική ή μη ταξινομημένη δραστηριότητα, αποκτώντας μια συνολική εικόνα της δικτυακής συμπεριφοράς που έχει παρατηρηθεί.

Η λειτουργία της ενότητας βασίζεται στο κεντρικό ιστορικό καταγραφών της εφαρμογής, το οποίο ενημερώνεται συνεχώς κατά τη διάρκεια της λειτουργίας του IDS. Οι νέες εγγραφές αποθηκεύονται αρχικά σε έναν προσωρινό χώρο μνήμης και στη συνέχεια συγχωνεύονται με το κεντρικό σύνολο δεδομένων, έτσι ώστε να βελτιώνεται η αποδοτικότητα της εφαρμογής. Αυτό επιτρέπει την ομαλή ανανέωση ακόμη και όταν ο αριθμός των καταγραφών αυξάνεται σημαντικά.

Ο πίνακας καταγραφών περιλαμβάνει ένα εργαλείο φιλτραρίσματος και διαχείρισης των δεδομένων στο επάνω μέρος. Πιο αναλυτικά, παρέχεται η δυνατότητα επιλογής χρονικού διαστήματος μέσω δύο πεδίων ημερομηνίας, τα οποία επιτρέπουν στον χρήστη να ορίσει ημερομηνία έναρξης και ημερομηνία λήξης για την προβολή των αρχείων καταγραφής. Κατά την ανανέωση του πίνακα το σύστημα φιλτράρει τις εγγραφές με βάση το επιλεγμένο χρονικό εύρος και προβάλλει μόνο όσες εμπίπτουν σε αυτό. Οι εγγραφές ταξινομούνται αυτόματα με φθίνουσα σειρά βάσει χρόνου έτσι ώστε η πιο πρόσφατη δραστηριότητα να εμφανίζεται πάντα στην αρχή του πίνακα. Με αυτόν τον τρόπο ο χρήστης μπορεί να περιορίσει την ανάλυση σε συγκεκριμένες χρονικές περιόδους διευκολύνοντας τη διερεύνηση περιστατικών.

Ο πίνακας της ενότητας προβάλλει όλα τα χαρακτηριστικά που σχετίζονται με κάθε ροή δικτυακής επικοινωνίας. Μεταξύ αυτών περιλαμβάνονται χρονικά πεδία, όπως τα Timestamp, detect_ts, flow_start_ts και detection_delay, πεδία ταυτοποίησης της επικοινωνίας όπως srcip, sport, dstip, dsport, proto, service και state καθώς και χαρακτηριστικών όπως sbytes, dbytes, ackdat, tcprtt, sloss, dloss, ACK Flag Count, Flow Bytes/s, Active Min, Active Mean, Active Std, Average Packet Size, Avg Bwd Segment

Size, Avg Fwd Segment Size, Bwd IAT Mean, Bwd IAT Std, Bwd Packet Length Max, Bwd Packet Length Mean, Bwd Packet Length Std, FIN Flag Count, Fwd Header Length.1, Fwd IAT Min, Fwd IAT Std, Fwd Packet Length Mean, Idle Mean, Init_Win_bytes_backward, Init_Win_bytes_forward, Max Packet Length, Min Packet Length, PSH Flag Count, Packet Length Mean, Packet Length Std, SYN Flag Count, Subflow Bwd Bytes, Subflow Bwd Packets, Subflow Fwd Packets, Subflow Fwd bytes, Total Backward Packets, Total Fwd Packets, Total Length of Bwd Packets, Total Length of Fwd Packets, URG Flag Count, act_data_pkt_fwd. Επίσης περιλαμβάνει και πιο εξειδικευμένα χαρακτηριστικά όπως sttl, dttl, ct_state_ttl, ct_srv_dst, ct_srv_src, ct_dst_src_ltm, ct_dst_sport_ltm, ct_src_dport_ltm, Dpkts, ct_ftp_cmd, swin, dwin, smeansz, dmeansz, synack, Sload, Dload και is_sm_ips_ports. Επιπλέον απεικονίζονται πεδία που σχετίζονται με τα αποτελέσματα των μοντέλων ανίχνευσης όπως attack_cic, pred_unsw, attack_unsw, prediction, attack_type, καθώς και η ενοποιημένη στήλη Attack Type, η οποία προκύπτει από τη μετονομασία του πεδίου attack_merged και εμφανίζεται αμέσως μετά το Timestamp ώστε το σημαντικότερο συμπέρασμα να είναι άμεσα ορατό.

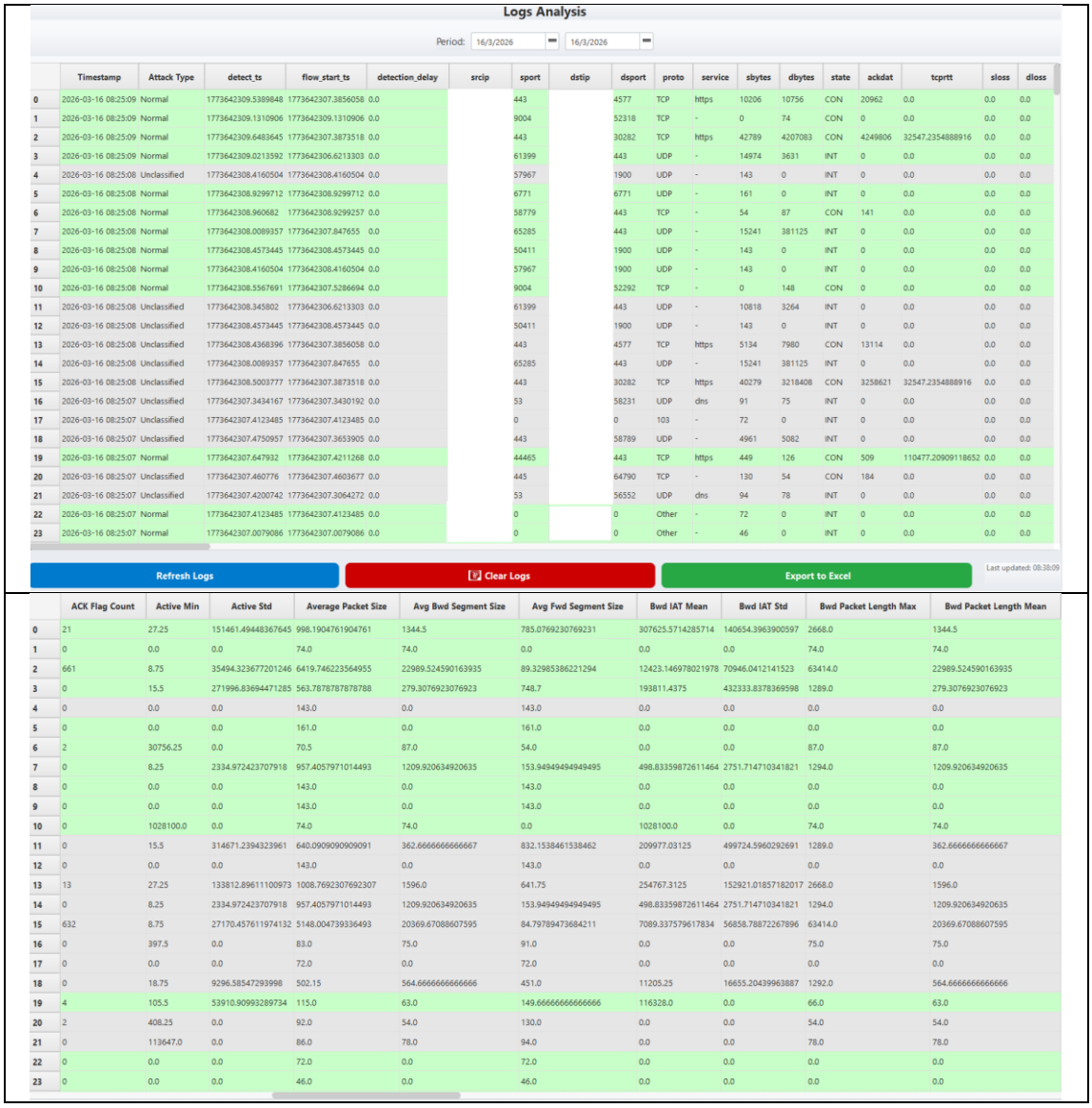
Οι εγγραφές που περιέχουν ένδειξη κανονικής λειτουργίας επισημαίνονται με πράσινο, οι εγγραφές που σχετίζονται με ανιχνευμένες επιθέσεις με κόκκινο και οι μη σαφώς ταξινομημένες περιπτώσεις με γκρι. Αυτό επιτρέπει στον χρήστη να εντοπίζει οπτικά τα πιο κρίσιμα συμβάντα χωρίς να χρειάζεται να διαβάσει κάθε γραμμή ξεχωριστά.

Η ενότητα Logs Analysis έχει σχεδιαστεί με τέτοιο τρόπο ώστε να παρέχει βασικές λειτουργίες διαχείρισης των καταγραφών. Ο χρήστης μπορεί να ανανεώσει χειροκίνητα τον πίνακα να διαγράψει το σύνολο των καταγραφών και να εξάγει τα δεδομένα σε αρχείο συμβατό με την εφαρμογή MS Excel. Η λειτουργία διαγραφής περιλαμβάνει μηχανισμό ασφαλείας που ζητά επιβεβαίωση από τον χρήστη πριν από την οριστική αφαίρεση των δεδομένων προλαμβάνοντας την απώλεια αρχείων από λάθος χειρισμό. Η εξαγωγή πραγματοποιείται σε τύπο αρχείου με κατάληξη .xlsx και περιλαμβάνει χρωματική απεικόνιση ανά γραμμή έτσι ώστε να διακρίνονται οι κακόβουλες και οι κανονικές εγγραφές και στο παραγόμενο αρχείο (Εικόνα 263). Η δυνατότητα αυτή είναι χρήσιμη καθώς η επεξεργασία των δεδομένων μπορεί να γίνει εύκολα και εκτός της εφαρμογής.

Μια εξίσου σημαντική λειτουργία της εφαρμογής είναι ότι οι καταγραφές αποθηκεύονται και σε αρχείο CSV στο υποσύστημα αρχείων της εφαρμογής, όπου όταν το κύριο αρχείο υπερβεί το μέγεθος των 100MB μετακινείται αυτόματα σε φάκελο αρχειοθέτησης,

συνοδευόμενο από χρονοσφραγίδα στο όνομα του αρχείου και δημιουργείται νέο ενεργό αρχείο logs. Κατά την εκκίνηση της εφαρμογής, το σύστημα φορτώνει τόσο το κύριο αρχείο καταγραφών όσο και τα αρχειοθετημένα αρχεία, συνθέτοντας ένα ενιαίο ιστορικό δεδομένων.

Επιπλέον η σελίδα ανανεώνεται αυτόματα όταν είναι ενεργή στην οθόνη, ώστε τα νέα δεδομένα να εμφανίζονται χωρίς χειροκίνητη παρέμβαση. Αυτή η αυτόματη ενημέρωση περιλαμβάνει ένδειξη της ώρας τελευταίας ανανέωσης παρέχοντας άμεση πληροφόρηση για το πόσο πρόσφατα είναι τα δεδομένα που εμφανίζονται. Εάν το φιλτράρισμα με βάση την ημερομηνία δεν επιστρέψει αποτελέσματα, η εφαρμογή προβάλλει κενό πίνακα και σχετικό μήνυμα ότι δεν βρέθηκαν καταγραφές για το επιλεγμένο χρονικό διάστημα.



	Timestamp	Attack Type	detect_ts	flow_start_ts	detection_delay	srcip	sport	dstip	dsport	proto	service	sbytes	dbytes	state	ackdat	tcprtt	sloss	dloss
0	2026-03-16 08:25:09	Normal	1773642309.5389848	1773642307.3856058	0.0		443		4577	TCP	https	10206	10756	CON	20962	0.0	0.0	0.0
1	2026-03-16 08:25:09	Normal	1773642309.1310906	1773642309.1310906	0.0		9004		52318	TCP	-	0	74	CON	0	0.0	0.0	0.0
2	2026-03-16 08:25:09	Normal	1773642309.6483645	1773642307.3873518	0.0		443		30282	TCP	https	42789	4207083	CON	4249806	32547.2354888916	0.0	0.0
3	2026-03-16 08:25:09	Normal	1773642309.0213592	1773642306.6213303	0.0		61399		443	UDP	-	14974	3631	INT	0	0.0	0.0	0.0
4	2026-03-16 08:25:08	Unclassified	1773642308.4160504	1773642308.4160504	0.0		57967		1900	UDP	-	143	0	INT	0	0.0	0.0	0.0
5	2026-03-16 08:25:08	Normal	1773642308.9299712	1773642308.9299712	0.0		6771		6771	UDP	-	161	0	INT	0	0.0	0.0	0.0
6	2026-03-16 08:25:08	Normal	1773642308.960682	1773642308.9299257	0.0		58779		443	TCP	-	54	87	CON	141	0.0	0.0	0.0
7	2026-03-16 08:25:08	Normal	1773642308.0089357	1773642307.847655	0.0		65285		443	UDP	-	15241	381125	INT	0	0.0	0.0	0.0
8	2026-03-16 08:25:08	Normal	1773642308.4573445	1773642308.4573445	0.0		50411		1900	UDP	-	143	0	INT	0	0.0	0.0	0.0
9	2026-03-16 08:25:08	Normal	1773642308.4160504	1773642308.4160504	0.0		57967		1900	UDP	-	143	0	INT	0	0.0	0.0	0.0
10	2026-03-16 08:25:08	Normal	1773642308.5567691	1773642307.5286694	0.0		9004		52292	TCP	-	0	148	CON	0	0.0	0.0	0.0
11	2026-03-16 08:25:08	Unclassified	1773642308.345802	1773642306.6213303	0.0		61399		443	UDP	-	10818	3264	INT	0	0.0	0.0	0.0
12	2026-03-16 08:25:08	Unclassified	1773642308.4573445	1773642308.4573445	0.0		50411		1900	UDP	-	143	0	INT	0	0.0	0.0	0.0
13	2026-03-16 08:25:08	Unclassified	1773642308.4368396	1773642307.3856058	0.0		443		4577	TCP	https	5134	7980	CON	13114	0.0	0.0	0.0
14	2026-03-16 08:25:08	Unclassified	1773642308.0089357	1773642307.847655	0.0		65285		443	UDP	-	15241	381125	INT	0	0.0	0.0	0.0
15	2026-03-16 08:25:08	Unclassified	1773642308.5003777	1773642307.3873518	0.0		443		30282	TCP	https	40279	3218408	CON	3258621	32547.2354888916	0.0	0.0
16	2026-03-16 08:25:07	Unclassified	1773642307.3434167	1773642307.3430192	0.0		53		58231	UDP	dns	91	75	INT	0	0.0	0.0	0.0
17	2026-03-16 08:25:07	Unclassified	1773642307.4123485	1773642307.4123485	0.0		0		0	103	-	72	0	INT	0	0.0	0.0	0.0
18	2026-03-16 08:25:07	Unclassified	1773642307.4750957	1773642307.3653905	0.0		443		58789	UDP	-	4961	5082	INT	0	0.0	0.0	0.0
19	2026-03-16 08:25:07	Normal	1773642307.647932	1773642307.4211268	0.0		44465		443	TCP	https	449	126	CON	509	110477.20909118652	0.0	0.0
20	2026-03-16 08:25:07	Unclassified	1773642307.460776	1773642307.4603677	0.0		445		64790	TCP	-	130	54	CON	184	0.0	0.0	0.0
21	2026-03-16 08:25:07	Unclassified	1773642307.4200742	1773642307.3064272	0.0		53		56552	UDP	dns	94	78	INT	0	0.0	0.0	0.0
22	2026-03-16 08:25:07	Normal	1773642307.4123485	1773642307.4123485	0.0		0		0	Other	-	72	0	INT	0	0.0	0.0	0.0
23	2026-03-16 08:25:07	Normal	1773642307.0079086	1773642307.0079086	0.0		0		0	Other	-	46	0	INT	0	0.0	0.0	0.0

	ACK Flag Count	Active Min	Active Std	Average Packet Size	Avg Bwd Segment Size	Avg Fwd Segment Size	Bwd IAT Mean	Bwd IAT Std	Bwd Packet Length Max	Bwd Packet Length Mean
0	21	27.25	151461.49448367645	998.1904761904761	1344.5	785.0769230769231	307625.5714285714	140654.3963900597	2668.0	1344.5
1	0	0.0	0.0	74.0	74.0	0.0	0.0	0.0	74.0	74.0
2	661	8.75	35494.323677201246	6419.746223564955	22989.524590163935	89.32985386221294	12423.146978021978	70946.0412141523	63414.0	22989.524590163935
3	0	15.5	271996.83694471285	563.7878787878788	279.3076923076923	748.7	193811.4375	432333.8378369598	1289.0	279.3076923076923
4	0	0.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
5	0	0.0	0.0	161.0	0.0	161.0	0.0	0.0	0.0	0.0
6	2	30756.25	0.0	70.5	87.0	54.0	0.0	0.0	87.0	87.0
7	0	8.25	2334.972423707918	957.4057971014493	1209.920634920635	153.94949494949495	498.83359872611464	2751.7147103418 21	1294.0	1209.920634920635
8	0	0.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
9	0	0.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
10	0	1028100.0	0.0	74.0	74.0	0.0	1028100.0	0.0	74.0	74.0
11	0	15.5	314671.2394323961	640.0909090909091	362.6666666666667	832.1538461538462	209977.03125	499724.5960292691	1289.0	362.6666666666667
12	0	0.0	0.0	143.0	0.0	143.0	0.0	0.0	0.0	0.0
13	13	27.25	133812.89611100973	1008.7692307692307	1596.0	641.75	254767.3125	152921.01857182017	2668.0	1596.0
14	0	8.25	2334.972423707918	957.4057971014493	1209.920634920635	153.94949494949495	498.83359872611464	2751.7147103418 21	1294.0	1209.920634920635
15	632	8.75	27170.457611974132	5148.004739336493	20369.67088607595	84.79789473684211	7089.337579617834	56858.78872267896	63414.0	20369.67088607595
16	0	397.5	0.0	83.0	75.0	91.0	0.0	0.0	75.0	75.0
17	0	0.0	0.0	72.0	0.0	72.0	0.0	0.0	0.0	0.0
18	0	18.75	9296.58547293998	502.15	564.6666666666666	451.0	11205.25	16655.20439963887	1292.0	564.6666666666666
19	4	105.5	53910.90993289734	115.0	63.0	149.66666666666666	116328.0	0.0	66.0	63.0
20	2	408.25	0.0	92.0	54.0	130.0	0.0	0.0	54.0	54.0
21	0	113647.0	0.0	86.0	78.0	94.0	0.0	0.0	78.0	78.0
22	0	0.0	0.0	72.0	0.0	72.0	0.0	0.0	0.0	0.0
23	0	0.0	0.0	46.0	0.0	46.0	0.0	0.0	0.0	0.0

	Bwd Packet Length Std	FIN Flag Count	Flow Bytes/s	Fwd Header Length.1	Fwd IAT Min	Fwd IAT Std	Fwd Packet Length Mean	Idle Mean	Init_Win_bytes_backward	Init_Win_bytes_forward
0	1291.0480238937666	0	9734.468644672244	260	27.25	181640.53963779955	785.0769230769231	0.0	0.0	0.0
1	0.0	0	74000000.0	0	0.0	0.0	0.0	0.0	0.0	0.0
2	23072.46400342378	0	1879632.001169204	9592	8.75	36739.882333977584	89.32985386221294	0.0	0.0	0.0
3	392.0393199697791	0	7751.989844285737	0	15.5	354753.9197984718	748.7	0.0	0.0	0.0
4	0.0	0	143000000.0	0	0.0	0.0	143.0	0.0	0.0	0.0
5	0.0	0	161000000.0	0	0.0	0.0	161.0	0.0	0.0	0.0
6	0.0	0	4584.436275687785	20	0.0	0.0	54.0	0.0	0.0	0.0
7	297.571114700938	0	2457616.8572628093	0	9.25	5161.955750482147	153.94949494949495	0.0	0.0	0.0
8	0.0	0	143000000.0	0	0.0	0.0	143.0	0.0	0.0	0.0
9	0.0	0	143000000.0	0	0.0	0.0	143.0	0.0	0.0	0.0
10	0.0	0	143.95489966404332	0	0.0	0.0	0.0	0.0	0.0	0.0
11	446.21096156663634	0	8165.978671799467	0	15.5	416395.01245118206	832.1538461538462	0.0	0.0	0.0
12	0.0	0	143000000.0	0	0.0	0.0	143.0	0.0	0.0	0.0
13	1259.8210984104053	0	12474.865623059453	160	27.25	156696.09854416203	641.75	0.0	0.0	0.0
14	297.571114700938	0	2457616.8572628093	0	9.25	5161.955750482147	153.94949494949495	0.0	0.0	0.0
15	20760.39334748542	0	2927772.8300500666	9512	8.75	31343.859284459748	84.79789473684211	0.0	0.0	0.0
16	0.0	0	417669.1445710858	0	0.0	0.0	91.0	0.0	0.0	0.0
17	0.0	0	72000000.0	0	0.0	0.0	72.0	0.0	0.0	0.0
18	550.5637514806477	0	91545.33339418478	0	18.75	8272.531092001105	451.0	0.0	0.0	0.0
19	3.0	0	2535.2151289301896	72	5544.0	52519.375	149.66666666666666	0.0	64240.0	65535.0
20	0.0	0	450526.5242265032	20	0.0	0.0	130.0	0.0	0.0	0.0
21	0.0	0	1513.4585520380977	0	0.0	0.0	94.0	0.0	0.0	0.0
22	0.0	0	72000000.0	0	0.0	0.0	72.0	0.0	0.0	0.0
23	0.0	0	46000000.0	0	0.0	0.0	46.0	0.0	0.0	0.0

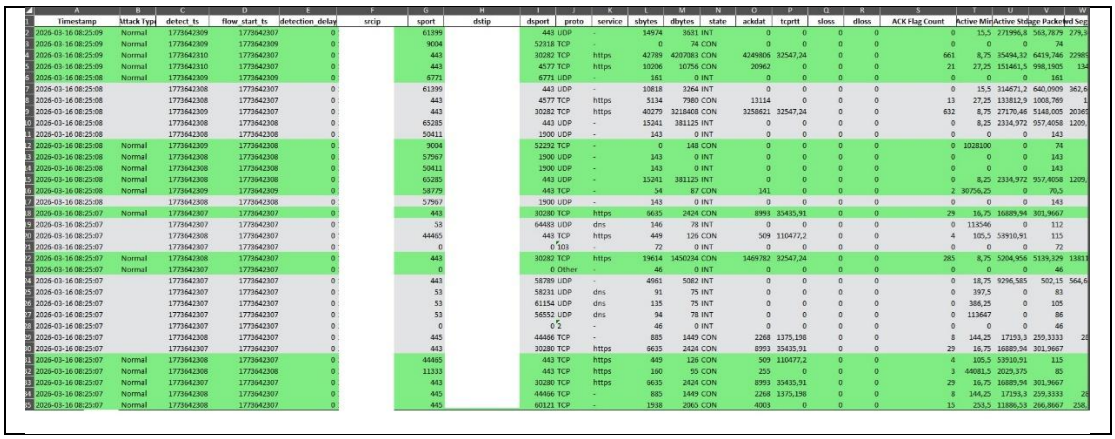
	Min Packet Length	PSH Flag Count	Packet Length Mean	Packet Length Std	SYN Flag Count	Subflow Bwd Bytes	Subflow Bwd Packets	Subflow Fwd Packets	Total Backward Packets	Total Fwd Packets
0	54.0	8	998.1904761904761	967.8929800663284	0	10756	8	13	8	13
1	74.0	0	74.0	0.0	1	74	1	0	1	0
2	54.0	41	6419.746223564955	15876.968434393115	2	4207083	183	479	183	479
3	68.0	0	563.7878787878788	571.4697562227757	0	3631	13	20	13	20
4	143.0	0	143.0	0.0	0	0	0	1	0	1
5	161.0	0	161.0	0.0	0	0	0	1	0	1
6	54.0	1	70.5	16.5	0	87	1	1	1	1
7	68.0	0	957.4057971014493	538.8657613531075	0	381125	315	99	315	99
8	143.0	0	143.0	0.0	0	0	0	1	0	1
9	143.0	0	143.0	0.0	0	0	0	1	0	1
10	74.0	0	74.0	0.0	2	148	2	0	2	0
11	74.0	0	640.0909090909091	575.2165528256537	0	3264	9	13	9	13
12	143.0	0	143.0	0.0	0	0	0	1	0	1
13	54.0	5	1008.7692307692307	1027.9124591465165	0	7980	5	8	5	8
14	68.0	0	957.4057971014493	538.8657613531075	0	381125	315	99	315	99
15	54.0	31	5148.004739336493	13589.455426828768	2	3218408	158	475	158	475
16	75.0	0	83.0	8.0	0	75	1	1	1	1
17	72.0	0	72.0	0.0	0	0	0	1	0	1
18	64.0	0	502.15	516.7257759198781	0	5082	9	11	9	11
19	54.0	1	115.0	107.09248339636167	2	126	2	3	2	3
20	54.0	1	92.0	38.0	0	54	1	1	1	1
21	78.0	0	86.0	8.0	0	78	1	1	1	1
22	72.0	0	72.0	0.0	0	0	0	1	0	1
23	46.0	0	46.0	0.0	0	0	0	1	0	1

	Total Length of Bwd Packets	Total Length of Fwd Packets	URG Flag Count	act_data_pkt_fwd	sttl	dttl	ct_state_ttl	ct_srv_dst	ct_srv_src	ct_dst_src_ltm	ct_dst_sport_ltm
0	10756	10206	0	8	120.0	128.0	520	713	683	683	735
1	74	0	0	0	0.0	53.0	0	3	457	3	3
2	4207083	42789	0	12	116.99791231732776	128.0	520	713	683	683	735
3	3631	14974	0	20	128.0	57.0	135	33	457	33	33
4	0	143	0	1	2.0	0.0	5	2	1	1	1
5	0	161	0	1	253.0	0.0	1	1	1	1	1
6	87	54	0	0	128.0	112.0	234	2	457	2	2
7	381125	15241	0	99	128.0	124.0	135	414	457	414	414
8	0	143	0	1	2.0	0.0	5	2	1	1	1
9	0	143	0	1	2.0	0.0	5	2	1	1	1
10	148	0	0	0	0.0	53.0	0	3	457	3	3
11	3264	10818	0	13	128.0	57.0	128	22	442	22	22
12	0	143	0	1	2.0	0.0	5	2	1	1	1
13	7980	5134	0	4	120.0	128.0	510	676	646	646	698
14	381125	15241	0	99	128.0	124.0	128	414	442	414	414
15	3218408	40279	0	10	116.9978947368421	128.0	510	676	646	646	698
16	75	91	0	1	62.0	128.0	26	12	12	12	12
17	0	72	0	1	1.0	0.0	5	1	1	1	1
18	5082	4961	0	11	57.0	128.0	26	48	20	20	698
19	126	449	0	1	128.0	55.0	148	5	10	5	5
20	54	130	0	1	127.0	128.0	510	48	11	11	26
21	78	94	0	1	62.0	128.0	26	12	12	12	12
22	0	72	0	1	1.0	0.0	3	1	1	1	1
23	0	46	0	1	1.0	0.0	3	1	17	1	1

	Dpkts	ct_ftp_cmd	swin	dwin	smeansz	dmeansz	synack	Sload	Dload	is_sm_ips_ports	attack_cic	pred_unsw	attack_unsw	prediction	attack_type
0	8	0	0.0	0.0	785.0769230769231	1344.5	0.0	4739.528050163387	4994.940594508565	0	BENIGN	Unclassified	Normal	0.0	Unclassified
1	1	0	0.0	0.0	0.0	74.0	0.0	0.0	74000000.0	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
2	183	0	0.0	0.0	89.32985386221294	22989.524590163935	32547.2354888916	18924.704955356083	1860707.2962138478	0	BENIGN	Unclassified	Normal	0.0	Unclassified
3	13	0	0.0	0.0	748.7	279.3076923076923	0.0	6239.091423183801	1512.8984211019356	0	BENIGN	Unclassified	Normal	0.0	Unclassified
4	0	0	0.0	0.0	143.0	0.0	0.0	143000000.0	0.0	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
5	0	0	0.0	0.0	161.0	0.0	0.0	161000000.0	0.0	1	BENIGN	Unclassified	Normal	0.0	Unclassified
6	1	0	0.0	0.0	54.0	87.0	0.0	1755.7415523910668	2828.6947232967186	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
7	315	0	0.0	0.0	153.94949494949495	1209.920634920635	0.0	94499.87769269432	2363116.979570115	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
8	0	0	0.0	0.0	143.0	0.0	0.0	143000000.0	0.0	0	BENIGN	Unclassified	Normal	0.0	Unclassified
9	0	0	0.0	0.0	143.0	0.0	0.0	143000000.0	0.0	0	BENIGN	Unclassified	Normal	0.0	Unclassified
10	2	0	0.0	0.0	0.0	74.0	0.0	0.0	143.95489966404332	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
11	9	0	0.0	0.0	832.1538461538462	362.6666666666667	0.0	6273.225200364056	1892.7534714354113	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
12	0	0	0.0	0.0	143.0	0.0	0.0	143000000.0	0.0	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
13	5	0	0.0	0.0	641.75	1596.0	0.0	4883.785275948393	7591.080347111059	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
14	315	0	0.0	0.0	153.94949494949495	1209.920634920635	0.0	94499.87769269432	2363116.979570115	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
15	158	0	0.0	0.0	84.79789473684211	20369.67088607595	32547.2354888916	36188.73546971116	2891584.0945803556	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
16	1	0	0.0	0.0	91.0	75.0	0.0	228963.20575884823	188705.93881223755	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
17	0	0	0.0	0.0	72.0	0.0	0.0	72000000.0	0.0	1	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
18	9	0	0.0	0.0	451.0	564.6666666666666	0.0	45221.18878507923	46324.14460910555	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
19	2	0	65535.0	64240.0	149.66666666666666	63.0	110477.20909118652	1979.6723354602698	555.5427934699198	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
20	1	0	0.0	0.0	130.0	54.0	0.0	318306.783420899	132219.7408056042	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
21	1	0	0.0	0.0	94.0	78.0	0.0	827.1226970440766	686.335854994021	0	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
22	0	0	0.0	0.0	72.0	0.0	0.0	72000000.0	0.0	1	BENIGN	Unclassified	Normal	0.0	Unclassified
23	0	0	0.0	0.0	46.0	0.0	0.0	46000000.0	0.0	1	BENIGN	Unclassified	Normal	0.0	Unclassified

color_tag	remote_ip	total_bytes
green		20962.0
green		74.0
green		4249872.0
green		18605.0
gray		Unclassified
green		161.0
green		141.0
green		396366.0
green		143.0
green		143.0
green		148.0
gray		Unclassified
gray		Unclassified
gray		Unclassified
gray		Unclassified
gray		Unclassified
gray		Unclassified
gray		Unclassified
gray		Unclassified
green		575.0
gray		Unclassified
gray		Unclassified
green		72.0
green		46.0

Εικόνα 262 Ενότητα Logs Analysis. Στις φωτογραφίες εμφανίζονται όλα τα χαρακτηριστικά του Πίνακα.



	Timestamp	Attack Type	detect_ts	flow_start_ts	detection_delay	srcip	sport	dstip	dsport	proto	service	sbytes	dbytes	state	ackdat	tcprtt	sloss	dloss	ACK Flag Count	Active Mi	Active Std	Page Packe	Fwd Seg
2	2026-03-16 08:25:09	Normal	1773642309	1773642307	0		61395		443	UDP	-	14974	3631	INT	0	0	0	0	0	15,5	271996,8	563,7879	279,5
3	2026-03-16 08:25:09	Normal	1773642309	1773642309	0		9004		52918	TCP	-	0	74	CON	0	0	0	0	0	0	0	74	
4	2026-03-16 08:25:09	Normal	1773642310	1773642307	0		443		30282	TCP	https	42789	4207083	CON	4249806	32547,24	0	0	661	8,75	35494,32	6419,746	2298
5	2026-03-16 08:25:09	Normal	1773642310	1773642307	0		443		4577	TCP	https	10206	10756	CON	20962	0	0	0	21	27,25	151461,5	998,1905	134
6	2026-03-16 08:25:08	Normal	1773642309	1773642309	0		6771		6771	UDP	-	161	0	INT	0	0	0	0	0	0	0	161	
7	2026-03-16 08:25:08		1773642308	1773642307	0		61399		443	UDP	-	10818	3264	INT	0	0	0	0	0	15,5	314671,2	640,0909	362,6
8	2026-03-16 08:25:08		1773642308	1773642307	0		443		4577	TCP	https	5134	7980	CON	13114	0	0	0	13	27,25	131812,9	1008,769	1
9	2026-03-16 08:25:08		1773642309	1773642307	0		443		30282	TCP	https	40279	3218408	CON	3258621	32547,24	0	0	632	8,75	27170,46	5148,005	2036
10	2026-03-16 08:25:08		1773642308	1773642308	0		65285		443	UDP	-	15241	381125	INT	0	0	0	0	0	8,25	2334,972	957,4058	1209
11	2026-03-16 08:25:08		1773642308	1773642308	0		59411		1900	UDP	-	143	0	INT	0	0	0	0	0	0	0	143	
12	2026-03-16 08:25:08	Normal	1773642309	1773642308	0		9004		52292	TCP	-	0	148	CON	0	0	0	0	0	1028100	0	74	
13	2026-03-16 08:25:08	Normal	1773642308	1773642308	0		57567		1900	UDP	-	143	0	INT	0	0	0	0	0	0	0	143	
14	2026-03-16 08:25:08	Normal	1773642308	1773642308	0		59411		1900	UDP	-	143	0	INT	0	0	0	0	0	0	0	143	
15	2026-03-16 08:25:08	Normal	1773642308	1773642308	0		65285		443	UDP	-	15241	381125	INT	0	0	0	0	0	8,25	2334,972	957,4058	1209
16	2026-03-16 08:25:08	Normal	1773642309	1773642309	0		58779		443	TCP	-	54	87	CON	141	0	0	0	2	30756,25	0	70,5	
17	2026-03-16 08:25:08		1773642308	1773642308	0		57567		1900	UDP	-	143	0	INT	0	0	0	0	0	0	0	143	
18	2026-03-16 08:25:07	Normal	1773642307	1773642307	0		443		30280	TCP	https	9635	2424	CON	8993	35435,91	0	0	29	16,75	16889,94	301,9667	
19	2026-03-16 08:25:07		1773642307	1773642307	0		53		64483	UDP	dns	146	78	INT	0	0	0	0	0	133546	0	112	
20	2026-03-16 08:25:07		1773642308	1773642307	0		44465		443	TCP	https	449	126	CON	509	110477,2	0	0	4	105,5	53910,91	115	
21	2026-03-16 08:25:07		1773642307	1773642307	0		0		0	103	-	72	0	INT	0	0	0	0	0	0	0	72	
22	2026-03-16 08:25:07	Normal	1773642308	1773642307	0		443		30282	TCP	https	19614	1450234	CON	1469782	32547,24	0	0	285	8,75	5204,956	5139,329	1381
23	2026-03-16 08:25:07	Normal	1773642307	1773642307	0		0		0	Other	-	46	0	INT	0	0	0	0	0	0	0	46	
24	2026-03-16 08:25:07		1773642307	1773642307	0		443		58789	UDP	-	4961	5082	INT	0	0	0	0	0	18,75	9296,585	502,15	564,6
25	2026-03-16 08:25:07		1773642307	1773642307	0		53		58231	UDP	dns	91	75	INT	0	0	0	0	0	397,5	0	83	
26	2026-03-16 08:25:07		1773642307	1773642307	0		53		61154	UDP	dns	135	75	INT	0	0	0	0	0	386,25	0	105	
27	2026-03-16 08:25:07		1773642307	1773642307	0		53		56552	UDP	dns	94	78	INT	0	0	0	0	0	113647	0	86	
28	2026-03-16 08:25:07		1773642307	1773642307	0		0		0	5	-	46	0	INT	0	0	0	0	0	0	0	46	
29	2026-03-16 08:25:07		1773642308	1773642307	0		445		44466	TCP	-	885	1449	CON	2268	1375,198	0	0	8	144,25	17193,3	259,3333	28
30	2026-03-16 08:25:07		1773642307	1773642307	0		443		30280	TCP	https	9635	2424	CON	8993	35435,91	0	0	29	16,75	16889,94	301,9667	
31	2026-03-16 08:25:07	Normal	1773642308	1773642307	0		44465		443	TCP	https	449	126	CON	509	110477,2	0	0	4	105,5	53910,91	115	
32	2026-03-16 08:25:07	Normal	1773642308	1773642308	0		11333		443	TCP	https	190	55	CON	255	0	0	0	3	44081,5	2029,375	85	
33	2026-03-16 08:25:07	Normal	1773642307	1773642307	0		443		30280	TCP	https	9635	2424	CON	8993	35435,91	0	0	29	16,75	16889,94	301,9667	
34	2026-03-16 08:25:07	Normal	1773642308	1773642307	0		445		44466	TCP	-	885	1449	CON	2268	1375,198	0	0	8	144,25	17193,3	259,3333	28
35	2026-03-16 08:25:07	Normal	1773642308	1773642307	0		445		60121	TCP	-	1938	2060	CON	4003	0	0	0	15	253,5	11886,53	266,8667	258

Εικόνα 263 Παράδειγμα εξαγόμενου αρχείου Excel από την εφαρμογή

7.8. Έλεγχος και διαχείριση IP (IP Control)

Η ενότητα IP Control του Shield IDS (την αρχική οθόνη της οποίας βλέπουμε στην Εικόνα 264) αποτελεί μια σημαντική λειτουργία καθώς συμβάλλει στην αξιοπιστία της εφαρμογής. Ένα NIDS παρακολουθεί τη ροή της δικτυακής κίνησης με στόχο τον εντοπισμό επιθέσεων ενώ στα συστήματα που βασίζονται στην ανίχνευση ανωμαλιών (anomaly based) η έννοια της φυσιολογικής συμπεριφοράς είναι ιδιαίτερα κρίσιμη γιατί κάθε απόκλιση από αυτή μπορεί να θεωρηθεί ύποπτη [8]. Επίσης, στη σχετική βιβλιογραφία επισημαίνεται ότι τα συστήματα ανίχνευσης που βασίζονται σε AI/ML, παρότι μπορούν να επιτύχουν υψηλή ακρίβεια συνεχίζουν να παράγουν ψευδώς θετικές ειδοποιήσεις, οι οποίες αυξάνουν τον φόρτο επαλήθευσης για τους τεχνικούς ασφαλείας [21]. Για αυτό τον λόγο, είναι χρήσιμο να υπάρχει και ένας μηχανισμός διαχείρισης εμπιστευόμενων ή ύποπτων IP διευθύνσεων στην λειτουργία του συστήματος.

Μέσω του IP Control το σύστημα συνδυάζει την αυτοματοποιημένη ανίχνευση των μοντέλων μηχανικής μάθησης με πολιτικές εμπιστοσύνης και αποκλεισμού που ορίζονται από τον τεχνικό ασφαλείας. Με αυτό τον τρόπο επιτυγχάνεται η καλύτερη προσαρμογή στις ιδιαιτερότητες του πραγματικού περιβάλλοντος δικτύου, ενώ παράλληλα ενισχύεται η αξιοπιστία της ανίχνευσης αλλά και η χρηστικότητα της εφαρμογής, ειδικότερα σε περιπτώσεις όπου απαιτείται μείωση ψευδώς θετικών ειδοποιήσεων για γνωστούς κόμβους.



Εικόνα 264 Διεπαφή χρήστη IP Control

Στην εφαρμογή η καρτέλα IP Control υλοποιεί αυτή τη λογική. Ο χρήστης μπορεί να ορίζει ποιες διευθύνσεις IP θα θεωρούνται αξιόπιστες και ποιες θα επισημαίνονται ως υψηλού κινδύνου ανεξάρτητα από την έξοδο των μοντέλων μηχανικής μάθησης. Έτσι το σύστημα δεν περιορίζεται μόνο στην πρόβλεψη αλλά ενσωματώνει και γνώση του πραγματικού περιβάλλοντος δικτύου από τον χρήστη. Η υλοποίηση αυτή βασίζεται στην βιβλιογραφία σύμφωνα με την οποία ακόμη και μοντέλα με υψηλή ακρίβεια (~96%) συνεχίζουν να παράγουν σημαντικό αριθμό ψευδώς θετικών ειδοποιήσεων ο οποίος μπορεί να φτάσει τις 1.000 περιπτώσεις σε πραγματικά δεδομένα. Αυτό κάνει αναγκαία την ύπαρξη μηχανισμών φιλτραρίσματος, προκειμένου να διατηρηθεί η πρακτική χρηστικότητα ενός IDS [21]. Στη βιβλιογραφία διαφαίνεται επίσης ότι οι μηχανισμοί λευκής και μαύρης λίστας δεν αποτελούν μόνο βοηθητικά στοιχεία διαχείρισης, αλλά μπορούν να ενσωματώνονται άμεσα στον μηχανισμό ελέγχου της επικοινωνίας, επηρεάζοντας την τελική απόφαση προώθησης ή απόρριψης πακέτων [51]. Η χρήση λευκής/μαύρης λίστας και μηχανισμών αυτόματης εκμάθησης (Learning Mode), αποτελεί μια πρακτική λύση για τη μείωση του φόρτου επαλήθευσης, λειτουργώντας συμπληρωματικά προς τα μοντέλα ανίχνευσης.

Στην εφαρμογή η καρτέλα IP Control περιλαμβάνει δύο βασικές λειτουργίες. Η πρώτη αφορά την εκμάθηση μέσω της οποίας το σύστημα παρακολουθεί την κίνηση και δημιουργεί σταδιακά λευκή λίστα από τις μη τοπικές διευθύνσεις IP που εμφανίζονται στην επικοινωνία. Η λογική αυτή συνδέεται άμεσα με την αρχή των anomaly-based NIDS, σύμφωνα με την οποία η δημιουργία προφίλ φυσιολογικής συμπεριφοράς αποτελεί βασική προϋπόθεση για τον αξιόπιστο εντοπισμό αποκλίσεων [8]. Η δεύτερη λειτουργία σχετίζεται με τον χρήστη, ο οποίος μπορεί χειροκίνητα να προσθέτει, να αφαιρεί και να αναζητά διευθύνσεις IP στη λευκή και μαύρη λίστα. Η επιλογή αυτή ενισχύεται και από βιβλιογραφικά παραδείγματα, όπου η μαύρη λίστα χρησιμοποιείται για τον αποκλεισμό μη εξουσιοδοτημένων κόμβων, ενώ η λευκή λίστα λειτουργεί ως μηχανισμός επαλήθευσης καταγεγραμμένων ή εξουσιοδοτημένων συσκευών, αποτρέποντας τη συμμετοχή άγνωστων κόμβων στην επικοινωνία και περιορίζοντας επιθέσεις [51]. Έτσι, επιβάλλεται πολιτική κανονικής ή κακόβουλης αντιμετώπισης για συγκεκριμένους κόμβους, παρακάμπτοντας το μοντέλο όταν η απόφαση είναι ήδη γνωστή από τον διαχειριστή.

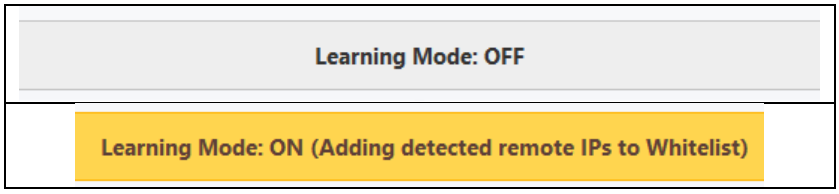
7.8.1. Learning Mode: Αυτόματη δημιουργία whitelist μέσω παρακολούθησης της κίνησης

Η λειτουργία εκμάθησης της εφαρμογής σχεδιάστηκε με στόχο την ημιαυτόματη δημιουργία λευκής λίστας μέσα από την παρακολούθηση της δικτυακής κίνησης. Αντί ο χρήστης να εισάγει χειροκίνητα όλες τις διευθύνσεις IP που θεωρούνται αξιόπιστες, το σύστημα καταγράφει τις απομακρυσμένες διευθύνσεις που εμφανίζονται στις ροές επικοινωνίας και τις προσθέτει σταδιακά στη λευκή λίστα. Με τον τρόπο αυτό μειώνεται η ανάγκη αρχικής χειροκίνητης παραμετροποίησης και διευκολύνεται η προσαρμογή της εφαρμογής στο πραγματικό περιβάλλον λειτουργίας του δικτύου.

Στο Shield IDS η λειτουργία ενεργοποιείται από την καρτέλα IP Control μέσω ειδικού πλήκτρου εναλλαγής (Εικόνα 265), του οποίου η ετικέτα μεταβάλλεται δυναμικά (από "OFF" σε "ON") παρέχοντας άμεση οπτική επιβεβαίωση της κατάστασης λειτουργίας στον χρήστη. Κατά την ενεργοποίηση αλλάζει η ένδειξη του πλήκτρου, ώστε να δηλώνεται ότι οι απομακρυσμένες IP προστίθενται στη λευκή λίστα. Όταν το κουμπί Learning Mode είναι ενεργό, κατά την επεξεργασία κάθε νέου στιγμιότυπου ροών, το σύστημα εξετάζει τις διευθύνσεις IP πηγής και προορισμού που εμφανίζονται στα δεδομένα. Για κάθε ροή, εφόσον η διεύθυνση δεν ανήκει στις τοπικές IP του ίδιου του υπολογιστή, προστίθεται αυτόματα στη λευκή λίστα. Αυτό εφαρμόζεται, τόσο για τη διεύθυνση πηγής όσο και για τη διεύθυνση προορισμού, ώστε να διαμορφώνεται σταδιακά μια λίστα από μη τοπικούς κόμβους με τους οποίους επικοινωνεί το σύστημα.

Για να εξαιρέσει η εφαρμογή τις τοπικές διευθύνσεις εντοπίζει κατά την εκκίνηση όλες τις IPv4 διευθύνσεις, που αντιστοιχούν στις τοπικές διεπαφές δικτύου, καθώς και τις ειδικές διευθύνσεις επαναφοράς τοπικού βρόχου (loopback) και μηδενικής δρομολόγησης. Έτσι αποφεύγεται η άσκοπη προσθήκη των τοπικών διευθύνσεων στη λευκή λίστα και το Learning Mode εστιάζει μόνο στους απομακρυσμένους κόμβους που παρατηρούνται στην επικοινωνία.

Επίσης όταν μια νέα διεύθυνση προστίθεται αυτόματα στη λευκή λίστα αφαιρείται ταυτόχρονα από τη μαύρη λίστα εφόσον υπήρχε ήδη εκεί. Αυτό γίνεται για να διασφαλιστεί ότι δεν θα υπάρχουν αντιφατικές εγγραφές για την ίδια IP σε διαφορετικές λίστες και ότι η τελική πολιτική αντιμετώπισης θα είναι συνεπής. Η προσθήκη μιας διεύθυνσης στη λευκή λίστα σημαίνει ότι κάθε νέα ροή που σχετίζεται με αυτήν την IP ταξινομείται αυτόματα ως κανονική (Normal), ενημερώνοντας άμεσα τα στατιστικά στοιχεία και τα γραφήματα του



Εικόνα 265 Απεικόνιση λειτουργίας εκμάθησης όταν είναι ανενεργή και όταν είναι ενεργοποιημένη

Dashboard. Μετά την ενημέρωση της λευκής λίστας, οι εγγραφές αποθηκεύονται μόνιμα στο τοπικό αρχείο ρυθμίσεων της εφαρμογής ενώ παράλληλα ανανεώνονται άμεσα και στη διεπαφή χρήστη ώστε οι αλλαγές να είναι ορατές.

Η λειτουργία εκμάθησης είναι ιδιαίτερα χρήσιμη σε περιβάλλοντα όπου απαιτείται σταδιακή καταγραφή των απομακρυσμένων διευθύνσεων IP που συμμετέχουν στην φυσιολογική επικοινωνία. Με αυτόν τον τρόπο η λευκή λίστα δημιουργείται σταδιακά από την δραστηριότητα κίνησης και μπορεί στη συνέχεια να αξιοποιηθεί ως μηχανισμός μείωσης των ψευδώς θετικών ειδοποιήσεων και βελτίωσης της χρηστικότητας της εφαρμογής.

7.8.2. Διαχείριση Μαύρης/Λευκής Λίστας (blacklist/whitelist) για έλεγχο πρόσβασης IP

Η λειτουργία Μαύρης/Λευκής Λίστας αφορά ένα σημαντικό μηχανισμό πολιτικής ασφαλείας ο οποίος συμπληρώνει τη λειτουργία της ανίχνευσης εισβολών. Η λευκή λίστα, αφορά ένα σύνολο διευθύνσεων IP που έχουν χαρακτηριστεί ως αξιόπιστες, ενώ στην μαύρη λίστα καταχωρούνται διευθύνσεις που έχουν αναγνωριστεί ως ύποπτες ή επιβεβαιωμένα κακόβουλες. Αυτές οι δύο λίστες επιτρέπουν την εφαρμογή σταθερών πολιτικών αξιοπιστίας και αποκλεισμού, οι οποίες πολιτικές επηρεάζουν άμεσα την τελική αντιμετώπιση της κυκλοφορίας.

Η λευκή λίστα έχει ως κύριο στόχο τη μείωση των ψευδώς θετικών αποτελεσμάτων. Σε πολλά δίκτυα υπάρχουν εξυπηρετητές, εσωτερικές υπηρεσίες, συσκευές και απομακρυσμένα συστήματα με τα οποία το δίκτυο επικοινωνεί συστηματικά και νόμιμα. Εφόσον αυτές οι διευθύνσεις είναι γνωστές, η προσθήκη τους στη λευκή λίστα επιτρέπει στο σύστημα να τις αντιμετωπίζει ως επιτρεπτές, ακόμη και αν η κυκλοφορία τους παρουσιάζει ορισμένα χαρακτηριστικά που θα μπορούσαν να εκληφθούν ως ύποπτα. Έτσι, η λευκή λίστα λειτουργεί ως μηχανισμός προσαρμογής της εφαρμογής στο πραγματικό περιβάλλον, περιορίζοντας περιττές ειδοποιήσεις.

Η μαύρη λίστα λειτουργεί ως μηχανισμός αυστηρού αποκλεισμού. Όταν μια διεύθυνση IP καταχωρηθεί σε αυτή οποιαδήποτε σχετική επικοινωνία αντιμετωπίζεται ως κακόβουλη ανεξάρτητα από το αν η συμπεριφορά της συγκεκριμένης ροής έχει ήδη επιβεβαιωθεί από τα μοντέλα. Η λογική της μαύρης λίστας έγκειται στο ότι ορισμένες απειλές είναι γνωστές εκ των προτέρων και δεν χρειάζεται να επαναξιολογούνται κάθε φορά. Αυτό είναι ιδιαίτερα σημαντικό σε περιπτώσεις επαναλαμβανόμενων επιθέσεων, γνωστών πηγών κακόβουλης δραστηριότητας ή IP που έχουν εντοπιστεί σε προηγούμενα περιστατικά ασφαλείας.

Η καταχώρηση μιας διεύθυνσης IP σε μια από τις δύο λίστες (Εικόνα 266) πραγματοποιείται χειροκίνητα από τον διαχειριστή ενώ παράλληλα παρέχεται η δυνατότητα αφαίρεσης ήδη καταχωρημένων διευθύνσεων (Εικόνα 268) έτσι ώστε η πολιτική να παραμένει και να προσαρμόζεται στις εξελισσόμενες ανάγκες του δικτύου. Επίσης η παρουσία ξεχωριστής απεικόνισης για τις δύο λίστες βοηθάει στην σωστή εποπτεία του συνόλου των κανόνων που εφαρμόζονται εκείνη τη στιγμή.

Για την καλύτερη διαχείριση των καταχωρήσεων περιλαμβάνεται λειτουργία αναζήτησης (Εικόνα 267). Αυτή η δυνατότητα είναι ιδιαίτερα σημαντική όταν οι λίστες περιλαμβάνουν μεγάλο αριθμό εγγραφών, καθώς επιτρέπει τον γρήγορο εντοπισμό συγκεκριμένων IP χωρίς να απαιτείται χειροκίνητη περιήγηση σε ολόκληρο τον κατάλογο. Με αυτό τον τρόπο η διεπαφή της εφαρμογής παραμένει λειτουργική ακόμη και σε περιπτώσεις μακροχρόνιας χρήσης όπου οι δύο λίστες ενδέχεται να αυξηθούν σημαντικά.

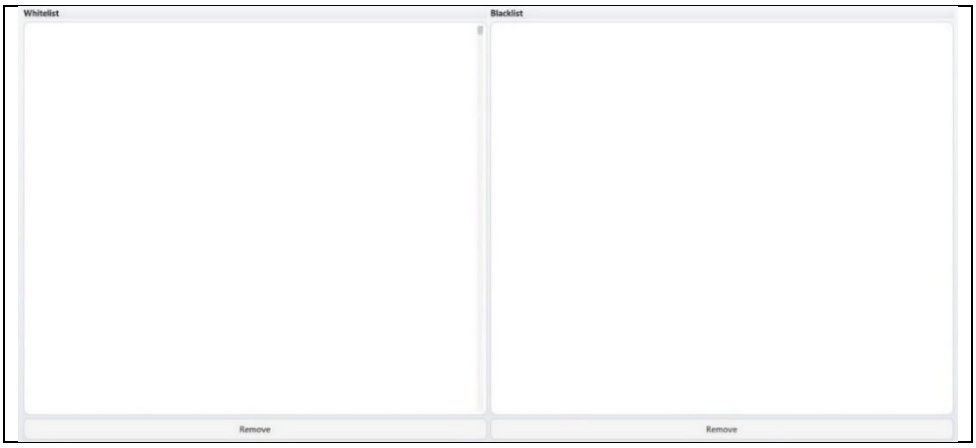
Μια ακόμα σημαντική προσθήκη είναι η μόνιμη αποθήκευση των λιστών. Οι καταχωρήσεις δεν ισχύουν μόνο για τη διάρκεια μιας εκτέλεσης της εφαρμογής, αλλά διατηρούνται ώστε να επαναφορτώνονται κατά την επόμενη εκκίνηση. Έτσι εξασφαλίζεται η συνέχεια στην πολιτική ασφάλειας και αποτρέπεται η ανάγκη επανακαθορισμού των ίδιων IP κάθε φορά που το σύστημα τίθεται σε λειτουργία.

Ο συνδυασμός των δύο λιστών μαζί με τους αλγορίθμους ανίχνευσης δημιουργεί ένα υβριδικό μοντέλο λειτουργίας, στο οποίο συνυπάρχουν η αυτόματη ανάλυση συμπεριφοράς και οι κανόνες ασφαλείας που θέτει ο χρήστης. Η λειτουργία της λευκής και της μαύρης λίστας ουσιαστικά συμβάλλει στη βελτίωση της αξιοπιστίας του συστήματος ανίχνευσης. Ειδικότερα, η λευκή λίστα συμβάλλει στη μείωση των ψευδώς θετικών ανιχνεύσεων, που αποτελούν ένα από τα σημαντικότερα προβλήματα των IDS, ενώ η μαύρη λίστα λειτουργεί ως μηχανισμός άμεσου αποκλεισμού γνωστών κακόβουλων ή ανεπιθύμητων διευθύνσεων.

Μέσω της δυνατότητας δήλωσης έμπιστων ή αποκλεισμένων διευθύνσεων, ο τεχνικός ασφαλείας μπορεί να επηρεάζει ελεγχόμενα την τελική συμπεριφορά του συστήματος, βελτιώνοντας την ακρίβεια της ανίχνευσης και μειώνοντας τον φόρτο επαλήθευσης. Τέλος εξαιτίας του προσεκτικού σχεδιασμού της καρτέλας IP Control ενισχύεται σημαντικά η διαφάνεια και ο έλεγχος του συστήματος.

Ο τεχνικός ασφαλείας παρακολουθεί τα αποτελέσματα ανίχνευσης και μπορεί να τα επηρεάσει με ελεγχόμενο τρόπο. Αυτό είναι σημαντικό σε πραγματικά περιβάλλοντα όπου η πλήρης αυτοματοποίηση δεν είναι πάντοτε επιθυμητή και η ανθρώπινη παρέμβαση παραμένει αναγκαία ειδικά όταν υπάρχει διαφοροποίηση μεταξύ γνωστής νόμιμης δραστηριότητας και πραγματικής απειλής.

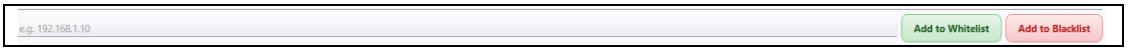
Σε αυτό το κεφάλαιο παρουσιάστηκαν οι βασικές λειτουργικές ενότητες του Shield IDS και ο τρόπος με τον οποίο αυτές υποστηρίζουν την παρακολούθηση, την ανάλυση και την ερμηνεία της δικτυακής δραστηριότητας. Μέσα από την περιγραφή των επιμέρους καρτελών και λειτουργιών αναδείχθηκε η λειτουργική οργάνωση της εφαρμογής και η εμπειρία χρήσης που προσφέρει. Στο επόμενο κεφάλαιο παρουσιάζεται η αρχιτεκτονική του συστήματος και αναλύεται ο τρόπος υλοποίησης των μηχανισμών που υποστηρίζουν τη λειτουργία του.



Εικόνα 266 Απεικόνιση της λευκής και μαύρης λίστας καθώς και τον κουμπιών αφαίρεσης IP διευθύνσεων



Εικόνα 267 Μπάρα αναζήτησης καταχωρημένων διεύθυνσεων IP



Εικόνα 268 Κουμπιά καταχώρησης μια διεύθυνσης IP είτε στην λευκή λίστα είτε στην μαύρη λίστα

8. Αρχιτεκτονική και ανάπτυξη εφαρμογής

Η εφαρμογή στηρίζεται σε μονολιθική αρχιτεκτονική και είναι βασισμένη στην PyQt5, βιβλιοθήκη της python κατάλληλη για τη δημιουργία γραφικών παραθύρων, κουμπιών, πινάκων, γραφημάτων κλπ. Πρόκειται δηλαδή για μια ενιαία εκτελέσιμη εφαρμογή που ενσωματώνει το γραφικό περιβάλλον (frontend), τη λογική της εφαρμογής (backend) και την επεξεργασία δεδομένων (data processing) σε ένα ενιαίο σύστημα χωρίς βάσεις δεδομένων ή εξωτερικές υπηρεσίες. Όλα τα δεδομένα, πλήν των αρχείων logs, διαχειρίζονται τοπικά, in-memory (δεν γράφονται συνεχώς σε κάποιο αρχείο ή βάση) φορτώνονται στη μνήμη RAM και αποθηκεύονται σε Data Frames (σε μορφή πινάκων) μέσω της ανοιχτού κώδικα βιβλιοθήκης Pandas, που παρέχει ισχυρά εργαλεία για το χειρισμό και την αποτελεσματική ανάλυση δομημένων δεδομένων. Έτσι, αποφεύγεται το συνεχές γράψιμο / διάβασμα από τον δίσκο (disk I/O) επιτρέποντας την υψηλή ταχύτητα κατά τη λειτουργία σε πραγματικό χρόνο (real time).

Οι βασικές ενότητες και υπηρεσίες χωρίζονται σύμφωνα με τις τρείς κύριες λειτουργίες (λογικός διαχωρισμός σε επίπεδα), όπως προαναφέρθηκαν: frontend, δεδομένων και backend. Στο frontend επίπεδο υπάρχει το βασικό παράθυρο της εφαρμογής με το οποίο έρχεται αντιμέτωπος ο χρήστης, αντικείμενο της κλάσης QWidget και το οποίο με τη σειρά του οργανώνει το περιεχόμενο της διεπαφής μέσω ενός QStackedWidget και πλευρικού μενού πλοήγησης, προκειμένου να υλοποιηθούν οι εφτά διαφορετικές ενότητες της εφαρμογής όπως παρουσιάστηκαν στο κεφάλαιο 8. Για την υλοποίησή τους χρησιμοποιήθηκαν από τη βιβλιοθήκη PyQt5 τα γραφικά στοιχεία (PyQt5.QtWidgets) QTableView για εμφάνιση των δεδομένων σε πίνακα και για τα logs, QPushButton για τις ενέργειες και Matplotlib για τα γραφήματα.

Στο επίπεδο δεδομένων γίνεται η αποθήκευση, οργάνωση κι ενημέρωση των δεδομένων όπως επίσης και η προώθηση τους προς το γραφικό περιβάλλον. Η υλοποίηση του επιπέδου αυτού επιτυγχάνεται μέσω της προσαρμοσμένης (custom) κλάσης PandasModel, υποκλάσης της QAbstractTableModel, που είναι υπεύθυνο για τη σύνδεση ενός DataFrame και ενός QTableView, μετατρέπει δηλαδή ένα DataFrame σε εμφανίσιμο πίνακα στο UI. Οι μέθοδοι που χρησιμοποιούνται για να επιτευχθεί αυτό είναι κυρίως η data() για την επιστροφή της τιμής ενός συγκεκριμένου κελιού, η rowCount() για το πλήθος γραμμών του πίνακα ή η dataChanged(), ένα «σήμα» δηλαδή που αποστέλλεται προς το DataFrame και

σημαίνει ότι τα δεδομένα άλλαξαν και απαιτείται ανανέωση του UI (αυτόματη ενημέρωση). Τα δεδομένα των DataFrames περιλαμβάνουν δικτυακές ροές (live network flows) και χαρακτηριστικά των συνόλων δεδομένων CIC-IDS2017 και UNSW-NB15, προβλέψεις ML και logs της εφαρμογής.

Το Backend επίπεδο περιλαμβάνει το FastSniffThread (κεφάλαιο 9.1.2) για την εκτέλεση του κώδικα σε ξεχωριστά νήματα (theads) και την βιβλιοθήκη pcapy που επιτρέπει την αλληλεπίδραση της βιβλιοθήκης libpcap για τη σύλληψη και ανάλυση πακέτων δικτύου σε πραγματικό χρόνο. Περιλαμβάνει ακόμα τον flow aggregator (κεφάλαιο 9.2), τη μονάδα αντιστοίχισης και προετοιμασίας χαρακτηριστικών σύμφωνα με τα CIC/UNSW (feature mapper, κεφάλαιο 9.2.2), το μοντέλο XGBoost το οποίο δέχεται τα χαρακτηριστικά και προβλέπει τη κατηγορία κίνησης (κανονική ή επίθεση), όπως επίσης και τα XAI συστατικά (components) για την παροχή επεξηγησιμότητας, τα οποία εκτελούνται σε διαφορετικά (background) νήματα.

Τρόπος επικοινωνίας:

Τα διαφορετικά μέρη που απαρτίζουν την εφαρμογή επικοινωνούν με άμεσες κλήσεις συναρτήσεων αλλά και με event-driven τρόπο, δηλαδή αντιδρούν σε γεγονότα που εκπέμπονται όταν συμβαίνει κάτι π.χ. όταν ολοκληρωθεί μια σύλληψη πακέτων. Αυτό, υλοποιείται μέσω του μηχανισμού signals / slots του Qt (μέρος του framework που περιλαμβάνει και το QThread) κατά το οποίο ένα module στέλνει ένα signal ενώ ένα άλλο έχει δηλώσει ένα slot για να το λάβει και να αντιδράσει. Η επικοινωνία αυτή είναι thread-safe, δηλαδή ασφαλής μεταξύ των διαφορετικών νημάτων εκτέλεσης. Το main UI thread, που χειρίζεται το γραφικό περιβάλλον και τα background threads για τη σύλληψη πακέτων, την εξαγωγή χαρακτηριστικών και τις προβλέψεις, μπορούν να ανταλλάζουν δεδομένα χωρίς να προκύπτουν καταστάσεις όπου δυο threads προσπαθούν να τροποποιήσουν ταυτόχρονα τα ίδια δεδομένα με αποτέλεσμα να δημιουργείται ασυνέπεια.

8.1. Συλλογή κι επεξεργασία δεδομένων

Η ροή δεδομένων ακολουθεί τρία βασικά επίπεδα:

1. UI → Controller: Στο επίπεδο του γραφικού περιβάλλοντος η αλληλεπίδραση ξεκινάει από τον χρήστη, όπου όταν πατήσει ένα κουμπί (QPushButton), ενεργοποιείται το clicked() signal το οποίο συνδέεται μέσω της συνάρτησης connect() με ένα slot του κεντρικού ελεγκτή.

2. Controller → Data Layer: Εδώ, ο Controller επικοινωνεί με το επίπεδο δεδομένων μέσω του background FastSniffThread. Το thread αυτό εκτελεί "βαριές" διεργασίες (όπως επεξεργασία δεδομένων), αλλά δεν μπλοκάρει το main UI thread, ενώ όταν ολοκληρώσει την επεξεργασία ενημερώνεται το κεντρικό νήμα (που διαβάζει περιοδικά μέσω Qtimer και on_snapshot), προκειμένου να ενημερωθούν τα DataFrames.

3. Data Layer → View: Σε αυτό το στάδιο το επίπεδο δεδομένων ενημερώνει το οπτικό περιβάλλον. Πιο συγκεκριμένα, όταν ενημερωθούν τα δεδομένα δημιουργείται ή ανανεώνεται το κατάλληλο PandasModel και συνδέεται με το QTableView, όπως επίσης και τα γραφήματα (matplotlib) ώστε να αντικατοπτρίζουν άμεσα τα νέα δεδομένα.

Η ακολουθία των παραπάνω επιπέδων μπορεί να συνοψιστεί ως εξής:

- Ο χρήστης πατά το κουμπί Start Capture
- Εκτελείται η μέθοδος η clicked()
- Καλείται η start_capture()
- Εκκινούνται τα FastSniffThread νήματα για τις διαθέσιμες διεπαφές
- Τα πακέτα συλλέγονται μέσω pcapy και προωθούνται στον FlowAggregator
- Ο QTimer ενεργοποιεί περιοδικά τη μέθοδο on_snapshot()
- Πραγματοποιείται εξαγωγή χαρακτηριστικών και πρόβλεψη από τα μοντέλα
- Ενημερώνονται τα DataFrames και δημιουργείται το κατάλληλο PandasModel
- Ενημερώνονται τα QTableView και τα Matplotlib plots

Εσωτερικά, η διαδικασία επεξεργασίας ξεκινάει με τη συλλογή των πακέτων μέσω της βιβλιοθήκης pcapy η οποία παρέχει πρόσβαση στα ακατέργαστα δεδομένα δικτύου. Η λειτουργία βασίζεται σε loop διαφορετικού νήματος, ώστε να επιτρέπεται η ασύγχρονη και non-blocking επεξεργασία. Τα πακέτα συλλέγονται (μέσω polling), επεξεργάζονται και προωθούνται στον flow aggregator ώστε να ομαδοποιηθούν σε ροές (flows) βάσει συγκεκριμένων χαρακτηριστικών όπως IP διευθύνσεις, θύρες και πρωτόκολλα.

Στη συνέχεια πραγματοποιείται η εξαγωγή των χαρακτηριστικών (feature extraction) από τις ροές που παρήχθησαν. Τα χαρακτηριστικά αυτά μετατρέπονται σε αριθμητική μορφή, αρχικά ως Pandas DataFrames και όπου απαιτείται ως δομή NumPy arrays έτσι ώστε να είναι συμβατά με το μοντέλο μηχανικής μάθησης. Η πρόβλεψη εκτελείται μέσω της

συνάρτησης XGBoost.predict(), η οποία δέχεται ως είσοδο τα αριθμητικά διανύσματα χαρακτηριστικών και επιστρέφει στο σύστημα μέσω thread signals το αποτέλεσμα ταξινόμησης ή πρόβλεψης. Τόσο τα αποτελέσματα αυτά όσο και τα συσχετιζόμενα μεταδεδομένα καταγράφονται σε in-memory δομή Pandas Dataframe (λειτουργεί ως προσωρινός χώρος αποθήκευσης), ενώ όταν απαιτείται εξαγωγή των δεδομένων χρησιμοποιείται η pandas.to_csv() για αρχεία CSV ή μέσω κατάλληλης διαδικασίας μετατροπής για παραγωγή αναφορών σε μορφή PDF. Η εξαγωγή πραγματοποιείται κατόπιν εντολής και όχι σε πραγματικό χρόνο, διατηρώντας έτσι τη βέλτιστη απόδοση του συστήματος.

Όπως προαναφέρθηκε, δεν χρησιμοποιούνται εξωτερικά APIs ή βάσεις δεδομένων, όλη η διαχείριση κι επικοινωνία μεταξύ των επιμέρους τμημάτων επιτυγχάνεται μέσω των memory buffers, Qt signals / slots, άμεσων κλήσεων μεθόδων και του threading synchronization (locks). Με αυτό το τρόπο το σύστημα παραμένει αυτόνομο, ελαφρύ και βελτιστοποιημένο για επεξεργασία σε πραγματικό χρόνο.

Αρχιτεκτονικά πρότυπα:

Το σύστημα βασίζεται στο αρχιτεκτονικό πρότυπο Model View Controller (MVC), σύμφωνα με το οποίο μια εφαρμογή χωρίζεται σε τρία διακριτά μέρη. Κατά το πρότυπο τα δεδομένα, σε ότι αφορά τη παρουσίασή τους και τη λογική ελέγχου, διαχωρίζονται με σαφήνεια ενώ παράλληλα επιτρέπεται η καλύτερη οργάνωση, η επεκτασιμότητα και η συντήρηση του λογισμικού. Η αντιστοίχιση του συστήματος στο MVC γίνεται ως εξής: Το Model αντιστοιχεί στο PandasModel και στα υποκείμενα DataFrames, τα οποία διαχειρίζονται και αποθηκεύουν τα δεδομένα. Η View υλοποιείται μέσω στοιχείων όπως το QTableView και τα επιμέρους tabs της εφαρμογής, τα οποία είναι υπεύθυνα για τη παρουσίαση των πληροφοριών στο χρήστη. Δεν περιέχει επιχειρησιακή λογική ούτε επεξεργάζεται άμεσα τα δεδομένα, αντίθετα, ενημερώνεται μέσω ανανέωσης του κατάλληλου model και των μηχανισμών γεγονότων του Qt όταν υπάρξει μεταβολή. Τέλος, ο Controller υλοποιείται μέσω της κεντρικής κλάσης (main class) και των επιμέρους threads. Είναι υπεύθυνος για τη διαχείριση των events και της επιχειρησιακής λογικής, λαμβάνει όλα τα ερεθίσματα από το UI, ενεργοποιεί τις κατάλληλες διεργασίες, ενημερώνει το Model και συντονίζει τη συνολική ροή της εφαρμογής.

Η διάκριση αυτή (σύμφωνη με το MVC) επιτρέπει ουσιαστικά την ύπαρξη πολλαπλών views που μπορούν να προβάλλουν ταυτόχρονα τα ίδια δεδομένα, ενώ παράλληλα υποστηρίζει και την εύκολη αντικατάσταση ή επέκταση των επιμέρους στοιχείων χωρίς να επηρεάζεται η συνολική αρχιτεκτονική.

8.1.1. Ανάλυση βασικών βιβλιοθηκών

α) Η βιβλιοθήκη PyQt5, που αποτελεί ένα από τα σημαντικότερα εργαλεία της Python σε ότι έχει να κάνει με τη δημιουργία γραφικών διεπαφών χρήστη (GUI), βασίζεται στο Qt, ένα πλήρες και ιδιαίτερα διαδεδομένο πλαίσιο (framework) ανεπτυγμένο σε C++ και λειτουργεί ως γέφυρα (python binding). Επιτρέπει δηλαδή τη συγγραφή του κώδικα σε python αξιοποιώντας όμως την υποδομή και τις κλάσεις του Qt. Σε αρχιτεκτονικό επίπεδο υποστηρίζει το πρότυπο MVC μέσω της λεγόμενης Model/View αρχιτεκτονικής του Qt, σύμφωνα με την οποία μια εφαρμογή διαχωρίζεται σε τρία διακριτά μέρη (model, view, controller), κυρίως για τη διαχείριση της πολυπλοκότητας. Ο διαχωρισμός αυτός επιτρέπει την δημιουργία εφαρμογών που μπορούν να επεκταθούν και να συντηρηθούν εύκολα. Για παράδειγμα, μπορούμε να αλλάξουμε τον τρόπο παρουσίασης των δεδομένων (view) χωρίς να χρειαστεί να τροποποιηθεί ο τρόπος αποθήκευσής τους (model). Επιπλέον, οι δυναμικές ενημερώσεις των δεδομένων γίνονται χωρίς να απαιτείται περαιτέρω συγγραφή κώδικα ή ανακατασκευή του UI, αφού το μοντέλο μπορεί να εκπέμπει σήματα, όπως το dataChanged(), ενημερώνοντας με αυτό το τρόπο αυτόματα τη View.

Η PyQt5 είναι cross-platform, που σημαίνει ότι η ίδια εφαρμογή μπορεί να τρέξει σε διαφορετικά λειτουργικά συστήματα (MS Windows, Linux, MacOS) χωρίς να απαιτούνται αλλαγές στον κώδικα. Αυτό επιτυγχάνεται επίσης μέσω του πλαισίου Qt και των platform plugins, που φροντίζουν να αποκτά η εφαρμογή αυθεντική εμφάνιση και αίσθηση (native look-and-feel) σε κάθε OS. Το PyInstaller είναι το εργαλείο μέσω του οποίου η εκτέλεση της εφαρμογής είναι ακόμα πιο εύκολη, αφού μετατρέπει την εφαρμογή σε αυτόνομο εκτελέσιμο αρχείο, ώστε ο τελικός χρήστης να μην χρειάζεται να εγκαταστήσει ξεχωριστά την python και τις επιπλέον βιβλιοθήκες από τις οποίες εξαρτάται η εφαρμογή. Βασικό στοιχείο της αρχιτεκτονικής Qt/PyQt5 είναι ο μηχανισμός signal-slot. Τα signals είναι γεγονότα, όπως το πάτημα ενός κουμπιού ή μια αλλαγή δεδομένων, ενώ τα slots είναι οι συναρτήσεις που εκτελούνται, αποκρινόμενες σε αυτά τα γεγονότα. Σε αυτό το μοντέλο επιτρέπεται η χαλαρή σύζευξη (loose coupling) μεταξύ των αντικειμένων, δηλαδή το αντικείμενο που εκπέμπει το signal δεν χρειάζεται να γνωρίζει ποιο αντικείμενο θα το

διαχειριστεί. Παράλληλα είναι thread-safe, παρέχοντας ασφαλή επικοινωνία μεταξύ των διαφορετικών νημάτων, γεγονός ιδιαίτερα σημαντικό, ειδικότερα στο πλαίσιο δημιουργίας ενός IDS αφού ένα νήμα, όπως το FastSniffThread, μπορεί να επεξεργάζεται πακέτα δικτύου χρησιμοποιώντας την βιβλιοθήκη pcapy (η οποία επιτρέπει packet capturing μέσω του libcap). Μόλις ολοκληρωθεί μια πρόβλεψη το thread εκπέμπει ένα prediction_ready() signal προς το κεντρικό UI thread, ώστε η ενημέρωση του UI να γίνεται με ασφαλή τρόπο, χωρίς να μπλοκάρει – παγώνει το GUI ακόμα και κατά τη διάρκεια έντονης επεξεργασίας πακέτων.

Στη βιβλιοθήκη περιέχονται πολλά έτοιμα γραφικά – δομικά στοιχεία (widgets) τα οποία μπορούν να αξιοποιηθούν, με χαρακτηριστικά παραδείγματα το κουμπί (QPushButton), τη διαχείριση διαφορετικών καρτελών (QStackedWidget), τη μπάρα προόδου (QProgressBar). Περιέχονται επίσης διαχειριστές διάταξης (layout managers), όπως οι QVBoxLayout και QGridLayout, οι οποίοι οργανώνουν τα widgets αυτόματα στο χώρο του παραθύρου επιτρέποντας στη διεπαφή να προσαρμόζεται σε αλλαγές του μεγέθους ή της ανάλυσης. Χαρακτηριστικό παράδειγμα, η IDS εφαρμογή μας, της οποίας η βασική δομή βασίζεται σε ένα QMainWindow, που λειτουργεί ως το κεντρικό παράθυρο και μέσα στο οποίο ένα QStackedWidget οργανώνει πολλαπλές λειτουργικές ενότητες σε 7 διαφορετικά tabs. Για την εκτέλεση παράλληλων εργασιών (parallel tasks) αξιοποιείται από το Qt το QThread, το οποίο επιτρέπει την εκτέλεση διαφορετικών διεργασιών παράλληλα με το κύριο νήμα, χωρίς να προκαλεί «πάγωμα» στην εφαρμογή.

Ο συνδυασμός της PyQt5 με βιβλιοθήκες όπως οι pandas, NumPy και Matplotlib (που χρησιμοποιήθηκαν) επιτρέπει ομαλή διαχείριση των δεδομένων, ορθούς αριθμητικούς υπολογισμούς και οπτικοποίηση. Έτσι μπορούν να παραχθούν σε πραγματικό χρόνο γραφήματα, όπως traffic plots, bar charts, pie charts και XAI γραφήματα στα οποία εξηγείται πως ένα μοντέλο, όπως αυτό του XGBoost, κατέληξε σε μια συγκεκριμένη πρόβλεψη – απόφαση.

β) Η βιβλιοθήκη Matplotlib θεωρείται ιδανική για την οπτικοποίηση των δεδομένων (data visualization) καθώς προσφέρει ένα πλήρες σύνολο εργαλείων για γραφήματα δύο διαστάσεων (2D) όπως τα γραμμικά διαγράμματα (line plots), τα γραφήματα ράβδων (bar charts), τα διαγράμματα διασποράς (scatter plots), οι χάρτες θερμότητας (heatmaps) και τα ιστογράμματα (histograms). Παρέχεται βασική υποστήριξη και για τριών διαστάσεων γραφικά (3D), αλλά το δυνατότερο σημείο είναι η δισδιάστατη απεικόνιση και αυτός είναι

και ο λόγος της ευρείας χρήσης της σε ανάλυση δεδομένων επιστημονικών εφαρμογών και μηχανικής μάθησης. Αρχιτεκτονικά βασίζεται σε αντικειμενοστραφές (object oriented) API το οποίο διαχωρίζει τα επιμέρους δομικά στοιχεία ενός γραφήματος που εμπεριέχονται στον κεντρικό καμβά (figure – το ανώτατο container). Εντός του κεντρικού καμβά υπάρχουν ένας ή περισσότεροι άξονες οι οποίοι αντιστοιχούν στις επιμέρους περιοχές σχεδίασης (subplots), ενώ κάθε οπτικό στοιχείο των αξόνων (όπως κείμενο, γραμμές ή γεωμετρικά σχήματα) είναι ένα artist object. Τα Backends είναι οι μηχανισμοί απόδοσης – ερμηνείας (rendering) που αναλαμβάνουν να μετατρέψουν το γράφημα σε ορατή μορφή, είτε στο ίδιο το παράθυρο ή την εφαρμογή είτε σε ξεχωριστό αρχείο (png, jpg κλπ.). Υπάρχουν δυο βασικοί τρόποι χρήσης της βιβλιοθήκης, ο βασισμένος στις καταστάσεις της διεπαφής (state-based) και ο αντικειμενοστραφής API (explicit object oriented API). Ο πρώτος τρόπος βασίζεται στο στοιχείο pyplot (matplotlib.pyplot), θεωρείται πιο απλός και καταλληλότερος για γρήγορο προγραμματισμό ενώ ο δεύτερος, όντας πολυπλοκότερος στις κλήσεις του (fig, ax=plt.subplots(); ax.plot(x, y)), προσφέρει λεπτομερέστερο έλεγχο και προτιμάται για μεγαλύτερες εφαρμογές στις οποίες απαιτείται δομημένη και επεκτάσιμη αρχιτεκτονική.

Η ενσωμάτωση της βιβλιοθήκης Matplotlib σε εφαρμογές με GUI που αναπτύσσονται με PyQt5 γίνεται μέσω του Backend FigureCanvasQTAgg, το οποίο επιτρέπει την απόδοση των γραφημάτων απευθείας μέσα στο παράθυρο της εφαρμογής, χωρίς να απαιτούνται ξεχωριστά pop-up παράθυρα, καθιστώντας έτσι τα γραφήματα οργανικό μέρος της εφαρμογής. Ο μηχανισμός σχεδίασης (renderer) λειτουργεί ανεξάρτητα από τη λογική της δημιουργίας των γραφημάτων και αυτό δίνει τη δυνατότητα εναλλαγής μεταξύ διαφορετικών backends, όπως ο Agg για την παραγωγή στατικών εικόνων υψηλής ποιότητας και ο QtAgg για διαδραστικά γραφήματα σε περιβάλλον Qt. Με αυτό το τρόπο η ίδια εφαρμογή μπορεί να εξάγει για παράδειγμα ένα διανυσματικό γράφημα υψηλής ανάλυσης και παράλληλα να υλοποιεί διαδραστικές οπτικοποιήσεις. Επιπλέον, η ενσωμάτωσή της με βιβλιοθήκες όπως το NumPy και το pandas επιτρέπει διανυσματική χαρτογράφηση μεγάλων συνόλων δεδομένων, ενώ τεχνικές όπως το blitting (βελτίωση απόδοσης κινούμενων αντικειμένων) και το animation API (FuncAnimation, κίνηση και χρονισμός) συμβάλλουν στην ομαλή και τη γρήγορη ανανέωση των γραφημάτων χωρίς οπτικό «τρεμόπαιγμα».

Σε επίπεδο υλοποίησης στην εφαρμογή μας αξιοποιείται το Qt backend ώστε τα αντικείμενα Figure να ενσωματώνονται στο περιβάλλον PyQt5 μέσω του FigureCanvasQTAgg, μετατρέποντας κάθε γράφημα σε QWidget που μπορεί να ενταχθεί σε ένα QMainWindow. Μέσω του signals /slots μηχανισμού (π.χ. timers και UI events), σε συνδυασμό με τις άμεσες ενημερώσεις δεδομένων, είναι εφικτή η δυναμική ενημέρωση των γραφημάτων, όπως τα traffic plots και confusion matrices, χωρίς την ανάγκη ανοίγματος νέων παραθύρων. Παράλληλα, το artist framework της Matplotlib (Figure → Axes → Artists) παρέχει ιεραρχικό και λεπτομερή έλεγχο σχεδίασης, με το σύστημα μετασχηματισμών να διαχειρίζεται τη χαρτογράφηση των συντεταγμένων και τα style sheets να εξασφαλίζουν ενα ομοιόμορφο αποτέλεσμα. Στη καρτέλα Dashboard, τα χαρακτηριστικά αυτά αξιοποιούνται για την απεικόνιση των KPIs και των διαγραμμάτων, με δυναμική ανανέωση μέσω εντολών όπως ax.clear(), ax.plot(new_data) και canvas.draw(), καθώς και με δυνατότητα εξαγωγής αναφορών σε μορφές SVG ή PDF.

8.1.2. Μηχανισμός multi-threading - Χρήση του FastSniffThread για τη συλλογή πακέτων

Το multi-threading βασίζεται στο framework QThread του PyQt5 (συγκεκριμένα στο PyQt5.QtCore.Qthread) και αποτελεί ένα πολύ σημαντικό στοιχείο για τη λειτουργία ανίχνευσης πραγματικού χρόνου του IDS καθώς βάση αυτού του μηχανισμού είναι εφικτή η ταυτόχρονη εκτέλεση πολλαπλών νημάτων. Έτσι οι εργασίες που απαιτούν υψηλότερη υπολογιστική ισχύ, όπως η επεξεργασία των πακέτων του δικτύου που μπορεί να είναι εκατοντάδες ή και χιλιάδες ανά δευτερόλεπτο, επιτρέπουν στο UI του χρήστη να παραμένει αποκριτικό (responsive). Το σχεδιαστικό μοτίβο που ακολουθείται για την αποσύνδεση των διεργασιών είναι αυτό του Παραγωγός – Καταναλωτής (Producer -Consumer). Το μέρος του συστήματος που λειτουργεί ως παραγωγός δεδομένων είναι ο FastSniffThread (υπό-κλάση του QThread της PyQt5), που συλλέγει τα πακέτα δικτύου μέσω pcapy και τα μεταβιβάζει στον συσσωρευτή ροών FlowAggregator, ενώ το main UI thread λειτουργεί ως καταναλωτής που διαβάζει περιοδικά (QTimer) τα δεδομένα μέσω του aggregator.snapshot_and_clear() και εκτελεί την εξαγωγή χαρακτηριστικών. Η επικοινωνία μεταξύ των νημάτων γίνεται μέσω ενός κοινόχρηστου συσσωρευτή ροών (shared flow aggregator) ο οποίος τροφοδοτείται από το νήμα συλλογής (FastSniffThread), ενώ για να εξασφαλιστεί η thread-safe λειτουργία εντός του FlowAggregator, υπάρχει μηχανισμός αμοιβαίου αποκλεισμού (self.aggregator.lock). Με αυτό το τρόπο μειώνεται σημαντικά

τόσο η πολυπλοκότητα της εφαρμογής όσο και ο κίνδυνος σφαλμάτων (π.χ. race conditions, deadlocks).

Η διαδικασία συλλογής πακέτων του FastSniffThread ξεκινά με τη σύλληψή τους (capturing) μέσω της βιβλιοθήκης pcapy και της συνάρτησης pcapy.findalldevs() ή οποία εντοπίζει όλες τις διαθέσιμες δικτυακές διεπαφές (network interfaces). Παράλληλα η pcapy.open_live() ανοίγει μια διεπαφή σε λειτουργία ζωντανής καταγραφής (live capture mode), επιτρέποντας έτσι την πρόσβαση στα ακατέργαστα πακέτα που διακινούνται στο δίκτυο. Στη συνέχεια εμπλέκεται ο flow aggregator (συσσωρευτής ροών) όπου τα μεμονωμένα πακέτα ομαδοποιούνται σε ροές δύο κατευθύνσεων μεταξύ πηγής και προορισμού (bidirectional) TCP/UDP καθώς και ICMP/OSPF όπου απαιτείται. Οι ροές αυτές συγκεντρώνονται σε χρονικά παράθυρα (time windows), για να ακολουθήσει η εξαγωγή των χαρακτηριστικών (feature extraction), όπου για κάθε ροή υπολογίζονται περισσότερα από 80 χαρακτηριστικά (όπως η διάρκεια, ο αριθμός πακέτων, ο όγκος δεδομένων κλπ.) όπως καθορίζονται από τα αντίστοιχα σύνολα δεδομένων. Το τελικό βήμα της διαδικασίας ολοκληρώνεται με τη συμμετοχή του XGBoost, που εκτελεί δυαδική ή πολυκλασική ταξινόμηση (binary, multiclass classification) χρησιμοποιώντας δέντρα αποφάσεων. Μέσα στη βασική μέθοδο run() του QThread εμπεριέχεται ένας βρόχος (while self._running loop) εντός του οποίου γίνεται αδιάλειπτα λήψη των πακέτων και άμεση προώθηση στον συσσωρευτή ροών. Το κεντρικό νήμα, που είναι υπεύθυνο για τη παραγωγή προβλέψεων και την ανάλυση, ανακτά περιοδικά τα δεδομένα από τον κοινόχρηστο συσσωρευτή ροών προκειμένου να εκτελεστεί το ML μοντέλο και να ενημερωθεί το UI.

Μόλις ο χρήστης πατήσει το κουμπί εκκίνησης καλείται η μέθοδος start_capture(), στην οποία εμπεριέχεται η pcapy.findalldevs() για τον εντοπισμό των διαθέσιμων δικτυακών διεπαφών προκειμένου να δημιουργηθεί ένα FastSniffThread νήμα για τη καθεμία διεπαφή. Κάθε FastSniffThread βρίσκεται σε ένα συνεχόμενο βρόχο (pcapy.open_live()) στον οποίο διαβάζει τα ακατέργαστα δικτυακά πακέτα, ενώ παράλληλα μεταφέρει δεδομένα απευθείας στο κοινόχρηστο συσσωρευτή ροών (aggregator.add_raw_packet(payload, ts)). Η ανάλυση κι ενημέρωση της διεπαφής λαμβάνουν χώρα μέσω του κεντρικού νήματος και με τη βοήθεια ενός QTimer (update_timer), ο οποίος περιοδικά ενεργοποιεί την on_snapshot() η οποία με τη σειρά της θα καλέσει την aggregator.snapshot_and_clear() για να πάρει ένα στιγμιότυπο (snapshot) των ροών και να υπολογίσει τα χαρακτηριστικά. Η προετοιμασία των δεδομένων (feature mapper) και οι προβλέψεις των μοντέλων πραγματοποιούνται στο

κεντρικό νήμα, στο οποίο επιπλέον παράγονται τα τελικά αποτελέσματα κι ενημερώνονται οι πίνακες, τα γραφήματα και τα alerts. Όταν ο χρήστης πατήσει «stop capture» τότε καλείται η stop_capture() μέσω της οποίας σταματά ο QTimer (update_timer.stop()), σταματούν όλα τα FastSniffThread νήματα (sniffer.stop()) και καθαρίζονται οι μετρητές και οι ροές εντός του συσσωρευτή ώστε να μην μείνουν δεδομένα σε εκκρεμότητα. Η συνολική αποδέσμευση των πόρων πραγματοποιείται με την ολοκλήρωση τερματισμού όλων των νημάτων.

8.1.3. Μοντέλο Δεδομένων: Χρήση του PandasModel για τη σύνδεση των DataFrames με τα UI Tables

Καθώς η εφαρμογή βασίζεται σε μια αρχιτεκτονική που έχει ως επίκεντρο τα δεδομένα (data-centric), τα pandas DataFrames αποτελούν κεντρικό δομικό στοιχείο. Όλη η επιπλέον πληροφορία που προκύπτει από την επεξεργασία και ανάλυση των δεδομένων των δικτυακών ροών μετατρέπεται εν τέλη σε DataFrame, το οποίο θα είναι πλέον το ενιαίο μοντέλο δεδομένων της εφαρμογής και θα αποτελεί το συνδετικό κρίκο μεταξύ του συστήματος ανάλυσης και του UI.

Η εκκίνηση της ροής δεδομένων πραγματοποιείται με τη συλλογή πακέτων μέσα από ένα ξεχωριστό νήμα καταγραφής, το οποίο είναι υπεύθυνο για τη λήψη των ακατέργαστων πακέτων από τη διεπαφή δικτύου και τη προώθηση των πακέτων αυτών στον συσσωρευτή ροών. Εκεί, γίνεται μια ομαδοποίηση των ροών σύμφωνα με τα χαρακτηριστικά τους (όπως IPs, θύρες ή πρωτόκολλο επικοινωνίας) ενώ δημιουργείται επίσης μια δομή στην οποία περιλαμβάνονται χαρακτηριστικά συμβατά με αυτά των συνόλων δεδομένων (όπως χρονικά στοιχεία, μετρητές πακέτων, στατιστικά μεγέθους ή TCP flags). Η δομή αυτή θα αντιπροσωπεύει πλέον τη ροή μέχρι και τη λήξη της, κατά την οποία ενεργοποιείται και η διαδικασία εξαγωγής στιγμιότυπου (snapshot). Σε αυτό το στάδιο, αρχικά γίνεται ένας υπολογισμός όλων των παράγωγων στατιστικών μεγεθών, έπειτα τα χαρακτηριστικά συγκεντρώνονται σε μορφή ενός λεξικού, ενώ στην συνέχεια το σύνολο των λεξικών συγκεντρώνεται σε μια λίστα. Η λίστα μετατρέπεται σε Pandas DataFrame, στο οποίο εμπεριέχονται μεταξύ άλλων η ταυτότητα της ροής (μέσω των χαρακτηριστικών της) και μεταδεδομένα ανίχνευσης και το οποίο αποτελεί πλέον το επίσημο μοντέλο δεδομένων στο οποίο θα βασιστούν η μηχανική μάθηση και η διαδικασία απεικόνισης.

Τα UI Tables συνδέονται με τα μοντέλα δεδομένων με βάση την αρχιτεκτονική Model/View του πλαισίου Qt, σύμφωνα με την οποία χρησιμοποιείται η υποκλάση (sub-class) του

QAbstractTableModel, με την ονομασία PandasModel, για την αποφυγή της απευθείας εισαγωγής δεδομένων σε ένα πίνακα. Το PandasModel ουσιαστικά λειτουργεί ως προσαρμογέας μεταξύ του pandas DataFrame και του QTableView. Το DataFrame αποθηκεύεται εσωτερικά στο μοντέλο και το Qt μεσολαβεί μέσω συγκεκριμένων μεθόδων (ορισμένες εκ των οποίων παρουσιάζει ενδεικτικά ο Πίνακας 12) ώστε να λάβει πληροφορίες για το σύνολο των γραμμών ή των στηλών όπως και για το περιεχόμενο του κάθε κελιού.

Η πληροφορία για το σύνολο των γραμμών / στηλών πηγάζει δυναμικά από το μέγεθος του DataFrame, ενώ το περιεχόμενο του κελιού επιστρέφεται από το PandasModel που αντλεί την αντίστοιχη τιμή του DataFrame, σύμφωνα με τη θέση γραμμής και στήλης. Έτσι, επιτρέπεται στο UI να προσαρμόζεται αυτόματα σε κάθε αλλαγή δεδομένων και παράλληλα μειώνεται η πολυπλοκότητα, αφού δεν πραγματοποιείται αντιγραφή δεδομένων, αλλά μια άμεση αντιστοίχιση. Τα ονόματα των στηλών του DataFrame χρησιμοποιούνται σαν επικεφαλίδες του πίνακα, οπότε και δεν απαιτείται ξεχωριστός ορισμός των επικεφαλίδων του πίνακα στο UI ενώ, ορισμένες στήλες του DataFrame μπορούν να περιέχουν πληροφορία παρουσίασης, όπως χρωματική κατηγοριοποίηση (dynamic color format) προκειμένου το μοντέλο να μπορεί να επιστρέψει διαφορετικό φόντο κελιού ανάλογα με την τιμή του. Έτσι, κατά την απεικόνιση των δεδομένων είναι άμεσα αντιληπτή η διάκριση μεταξύ κανονικής ή κακόβουλης δραστηριότητας.

Μέθοδος	**Αποτέλεσμα**
rowCount()	Επιστρέφει το σύνολο των γραμμών
columnCount()	Επιστρέφει το σύνολο των στηλών
data()	Επιστρέφει τη τιμή ενός κελιού
headerData()	Επιστρέφει του τίτλους των στηλών / γραμμών
flags()	Καθορίζει αν το κελί είναι επεξεργάσιμο / επιλέξιμο
set(data)	Επιτρέπει τροποποίηση δεδομένων από το UI

Πίνακας 12 Μέθοδοι άντλησης και τροποποίησης πληροφοριών

Η αρχιτεκτονική που χρησιμοποιείται βασίζεται στον διαχωρισμό των διεργασιών – ευθυνών. Οι διαδικασίες που αφορούν τη συλλογή ροών και την εξαγωγή των χαρακτηριστικών τους ανήκουν στο επίπεδο που αφορά την λογική και επεξεργασία δεδομένων, το PandasModel ανήκει στο επίπεδο διασύνδεσης με το GUI, ενώ το DataFrame είναι ο κοινός φορέας πληροφορίας.

8.2. Μονάδα Συλλογής και Επεξεργασίας Ροών (flow aggregator)

Η μονάδα συλλογής κι επεξεργασίας των ροών (Flow Aggregator) είναι ίσως το πιο σημαντικό στοιχείο του συστήματος, καθώς φέρει την ευθύνη για την μετατροπή των ακατέργαστων πακέτων δικτυακής κίνησης σε δομημένα χαρακτηριστικά που μπορούν να αναλυθούν και να ταξινομηθούν, μέσα από τα επαναλαμβανόμενα βήματα σταδιακής επεξεργασίας (λήψη των πακέτων, κανονικοποίηση, συσχέτιση με τη ροή, στατιστική ενημέρωση και παραγωγή του στιγμιότυπου δεδομένων).

Μόλις τα ακατέργαστα πακέτα αρχίσουν να παραλαμβάνονται, ξεκινάει παράλληλα και η ανάλυση σε επίπεδο δικτύου, όπου μόλις επιβεβαιωθεί ότι πρόκειται για IP κίνηση τότε γίνεται η εξαγωγή των βασικών πληροφοριών όπως είναι οι διευθύνσεις αποστολέα και παραλήπτη, το πρωτόκολλο επικοινωνίας που χρησιμοποιείται, οι θύρες και το μήκος του ωφέλιμου φορτίου (payload). Ειδικότερα, αν πρόκειται για το πρωτόκολλο TCP εξάγονται και αναλύονται επιπλέον τα χαρακτηριστικά σημαιών (flags), αριθμών ακολουθίας (sequence numbers), μέγεθος παραθύρου και μήκος επικεφαλίδας. Μέσω της διαδικασίας αυτής (κανονικοποίηση) το διαφορετικό περιεχόμενο των πακέτων μετατρέπεται σε ομοιόμορφη δομή. Εφόσον ολοκληρωθεί η κανονικοποίηση δημιουργείται ένα κανονικοποιημένο κλειδί ροής (canonical flow key) σύμφωνα με το ζεύγος διευθύνσεων και θυρών όπως επίσης και το πρωτόκολλο. Ο σκοπός του κλειδιού αυτού είναι να μπορούν να ενοποιηθούν τα πακέτα που ανήκουν στην ίδια αμφίδρομη σύνδεση. Εάν μια ροή υπάρχει ήδη στη μνήμη τότε γίνεται ενημέρωση των χαρακτηριστικών της (όπως ο συνολικός αριθμό πακέτων και τα χρονικά όρια), εάν δεν υπάρχει στη μνήμη τότε δημιουργείται μια νέα εγγραφή στην οποία αρχικοποιούνται οι μετρητές και οι δομές αποθήκευσης.

Η μονάδα διατηρεί καθολικούς (global) μετρητές που απαιτούνται για χαρακτηριστικά του συνόλου δεδομένων UNSW (όπως το σύνολο των συνδέσεων ανά υπηρεσία και ανά προορισμό) αλλά και ξεχωριστά στατιστικά στοιχεία για την κάθε κατεύθυνση της ροής.

Βάση αυτού καταγράφονται το σύνολο των πακέτων και των bytes ανά κατεύθυνση, το μήκος των πακέτων, οι χρόνοι άφιξης, οι τιμές του TTL και το σύνολο των ενεργών πακέτων με ωφέλιμο φορτίο. Για τις ροές TCP πραγματοποιείται αναλυτικότερη καταμέτρηση του συνόλου των flags (SYN, ACK, RST, FIN, URG, PSH) και υπολογίζονται το μέγεθος του αρχικού παραθύρου όπως επίσης και η καθυστέρηση χειραψίας (SYN – SYN/ACK latency) η οποία χρησιμοποιείται ως ένδειξη του χρόνου μετ' επιστροφής (RTT).

Για την αποδοτικότερη και σταθερότερη συμπεριφορά του συστήματος, καθώς οι ενημερώσεις γίνονται από διαφορετικό νήμα, η δομή των ροών προστατεύεται από μηχανισμό κλειδώματος (threading lock). Παράλληλα, υπάρχει περιορισμός του μέγιστου πλήθους των ενεργών ροών έτσι ώστε, όταν ξεπεραστεί το προκαθορισμένο όριο (max_flows=30000) να πραγματοποιείται διαγραφή του 10% των εγγραφών (αποφεύγοντας έτσι την υψηλή κατανάλωση μνήμης). Επιπλέον, όταν ο συνολικός αριθμός των ενεργών ροών αυξηθεί υπερβολικά, εφαρμόζεται πρόσθετος μηχανισμός εκκαθάρισης της μνήμης, ώστε να διατηρείται η σταθερότητα της εφαρμογής σε συνθήκες υψηλού φόρτου.

Τα χαρακτηριστικά εξάγονται μέσω της διαδικασίας λήψης στιγμιότυπου (snapshot), σύμφωνα με την οποία, γίνεται ο υπολογισμός στατιστικών μεγεθών όπως η διάρκεια ροής, ο ρυθμός μετάδοσης (Bytes/sec), οι μέσοι όροι και οι τυπικές αποκλίσεις των διαστημάτων άφιξης των πακέτων και των ενεργών ή των αδρανών περιόδων. Μόλις ολοκληρωθούν οι απαραίτητοι υπολογισμοί και τα δεδομένα οργανωθούν σε ενιαίο λεξικό χαρακτηριστικών, οι ροές που έχουν ξεπεράσει το χρονικό όριο αδράνειας (TIME OUT) αφαιρούνται από τη μνήμη και τα χαρακτηριστικά συγκεντρώνονται σε δομή πινάκων που στη συνέχεια μετατρέπεται στο Pandas DataFrame, το οποίο θα αποτελέσει την τελική έξοδο της μονάδας συλλογής και επεξεργασίας ροών και το οποίο θα χρησιμοποιηθεί για τα επόμενα στάδια (μετασχηματισμός χαρακτηριστικών, ML κλπ.).

8.2.1. Διαδικασία packet capturing, η χρήση της βιβλιοθήκης pcapy για απευθείας πρόσβαση στην κάρτα δικτύου

Η διαδικασία σύλληψης πακέτων υλοποιείται μέσω του νήματος FastSniffThread (κεφ. 9.1.2), το οποίο είναι υπεύθυνο για τη συνεχή συλλογή των πακέτων (από τη διεπαφή δικτύου του συστήματος) και τη προώθησή τους για επεξεργασία. Η πρόσβαση στα πακέτα του δικτύου επιτυγχάνεται μέσω της βιβλιοθήκης pcapy (Python wrapper της βιβλιοθήκης libpcap), μέσω της οποίας επιτρέπεται στην εφαρμογή η απευθείας λήψη των διερχόμενων

πακέτων, από την διεπαφή δικτύου, πριν καν αυτά τα πακέτα φτάσουν σε εφαρμογές του OS. Η σύνδεση με την διεπαφή δικτύου γίνεται με τη χρήση της pcapy.open_live (), τα ορίσματα της οποίας είναι βασικές παράμετροι της καταγραφής, όπως σε ποια διεπαφή θα γίνει η σύνδεση, ποιο θα είναι το μέγιστο μέγεθος πακέτου που θα συλλέγεται, αν θα υπάρχει διαχωρισμός των πακέτων (ανάμεσα σε αυτά που αφορούν μόνο στην συγκεκριμένη συσκευή ή σε όλα τα πακέτα) καθώς επίσης και ποιος θα είναι ο μέγιστος χρόνος αναμονής για την επιστροφή των πακέτων.

Μόλις ολοκληρωθεί η σύνδεση με τη διεπαφή ξεκινάει ένας συνεχόμενος βρόχος εκτέλεσης (while loop) στον οποίο εισέρχεται το νήμα και εντός του οποίου καλείται επαναλαμβανόμενα η μέθοδος cap.next() με τη βοήθεια της οποίας το σύστημα θα λάβει πληροφορίες σχετικές με τη κεφαλίδα και με το πραγματικό περιεχόμενο του πακέτου σε ακατέργαστη μορφή. Κάθε φορά που τα πεδία (μεταβλητές) που θα φιλοξενήσουν αυτές τις δύο πληροφορίες ενημερώνονται, σημαίνει ότι έχει ληφθεί ένα νέο κι έγκυρο πακέτο και άρα, αφού ενημερωθεί ο μετρητής των πακέτων, το πακέτο αυτό με τη χρονική στιγμή λήψης του προωθείται προς επεξεργασία. Για τη διασφάλιση της ποιότητας της καταγραφής πραγματοποιείται περιοδικά μια κλήση στα στατιστικά στοιχεία που συλλέγονται από την συνάρτηση cap.stats(). Οι πληροφορίες που επιστρέφονται από αυτήν αφορούν το σύνολο των πακέτων που έχουν ληφθεί και πόσα από αυτά έχουν απορριφθεί λόγω υπερφόρτωσης και διατηρούνται σε μετρητές έτσι ώστε ανά πάσα ώρα και στιγμή να είναι μετρήσιμη η απόδοση του μηχανισμού καταγραφής και να μπορούν να ανιχνευθούν πιθανές απώλειες πακέτων κατά τη διάρκεια λειτουργίας του συστήματος. Μόλις ολοκληρωθεί η συλλογή των ακατέργαστων bytes ενός πακέτου τότε αυτά, με τη χρήση της add_raw_packet(), στέλνονται στο συσσωρευτή ροών για περαιτέρω ανάλυση του πακέτου, εξαγωγή των βασικών πεδίων (σε επίπεδο δικτύου και μεταφοράς) καθώς και για την ενημέρωση των αντίστοιχων ροών επικοινωνίας.

```
# Κλάση – νήμα για συνεχή σύλληψη πακέτων
class FastSniffThread(QtCore.QThread):
      # Δημιουργία αντικειμένου (constructor)
      # Παράμετροι: Αντικείμενο κλάσης, διεπαφή δικτύου, FlowAggregator
      def __init__(self, iface=None, aggregator=None):
            super().__init__() # Κλήση constructor
            self.iface = iface # Αποθήκευση διεπαφής
            self.aggregator = aggregator # FlowAggregator
```

```
        self._running = False # Έλεγχος while loop (flag)


        self.total_raw_packets = 0 # Μετρητής πακέτων


    def run(self): # Μέθοδος εκκίνησης
        self._running = True # while loop flag = 1


        try: # Για σφάλματα αρχικοποίησης pcapy
            # Σύνδεση με την διεπαφή δικτύου.
            # Παράμετροι: network interface, snaplen=65536 (ολόκληρο το # πακέτο), promiscuous=1 (όλα τα πακέτα), timeout=100ms
            cap = pcapy.open_live(self.iface, 65536, 1, 100)


            while self._running: # Βρόχος εκτέλεσης
            try: # Για σφάλματα χειρισμού πακέτων
                # Σύλληψη επόμενου πακέτου
                (header, payload) = cap.next(
                    if header and payload: # Εφόσον είναι διαθέσιμα
                    self.total_raw_packets += 1 # Αύξηση μετρητή


                # Έλεγχος πακέτων που απορρίφθηκαν (ανά 50 πακέτα)
                if self.total_raw_packets % 50 == 0:
                try: # Για σφάλματα stats()
                # Στατιστικά σύλληψης. Παράμετροι: σύνολο πακέτων
                # που ελήφθησαν, πακέτων που απορρίφθηκαν από το ΛΣ
                # (libpcap/kernel) και αυτών που απορρίφθηκαν από τη
                # διεπαφή δικτύου
                    stats = cap.stats()
                        # index 1:Απορριφθέντα από libpcap/kernel
                        dropped = stats[1]
                        # Ενημέρωση aggregator
                        # Έλεγχος λειτουργικότητας
                        if self.aggregator:
```

```
                                    self.aggregator.cnt_dropped_pkts =
                                          dropped # Global μετρητής

                        except Exception:
                              pass # Συνέχεια, χωρίς στατιστικά
                  # Logging ανά 1000 πακέτα
                  if self.total_raw_packets % 1000 == 0
                        pass

                  ts = time.time() # Τρέχουσα ώρα
                  if self.aggregator: # Αν είναι διαθέσιμος
                        # Ανάλυση Bytes σε flows
                        self.aggregator.add_raw_packet(payload, ts)

                              except pcapy.PcapError: # Σφάλμα pcapy
                              continue # skip κι επόμενο πακέτο
                  except Exception: # Γενικό σφάλμα
                        continue # skip και συνέχεια

      except Exception as e: # Σφάλμα σύνδεσης με τη διεπαφή δικτύου
            print(f"Fast sniff error: {e}") # Λεπτομέρειες σφάλματος
            logging.error(f"FastSniffThread failed: {e}",
                  exc_info=True) # Logging σφάλματος


def stop(self): # Μέθοδος παύσης
      self._running = False # Έλεγχος while loop (flag)
```

8.2.2. Εξαγωγή και αντιστοίχιση χαρακτηριστικών: Πώς ο συσσωρευτής ροών μετατρέπει τα πακέτα σε χαρακτηριστικά (CIC και UNSW features) σε πραγματικό χρόνο και πως τροφοδοτούνται τα μοντέλα.

Οι διαδικασίες εξαγωγής και αντιστοίχισης χαρακτηριστικών πραγματοποιούνται από δύο διαφορετικές κλάσεις την FlowAggregator και την FeatureMapper αντίστοιχα. Η FlowAggregator λειτουργεί ως ένα ενδιάμεσο επίπεδο μεταξύ της συλλογής πακέτων και των μοντέλων μηχανικής μάθησης. Έτσι, μόλις ένα πακέτο συλληφθεί από το μηχανισμό

καταγραφής (κεφ. 9.2.1) τότε αυτό προωθείται στην μέθοδο add_raw_packet() όπου και με τη χρήση της βιβλιοθήκης dpkt γίνεται η ανάλυση του πακέτου.

```
# Ανάλυση πακέτου (Απόσπασμα μεθόδου)
# Παράμετροι: Ακατέργαστα bytes, timestamp
def add_raw_packet(self, raw_bytes, ts): # Επεξεργασία με βιβλιοθήκη dpkt
try: # Μπλοκ ανάλυσης
      # Ethernet αντικείμενο με raw Bytes
      eth = dpkt.ethernet.Ethernet(raw_bytes)
      # Αν δεν είναι IP πακέτο (και όχι π.χ. ARP)
      if not isinstance(eth.data, dpkt.ip.IP):
            return # break

      ip = eth.data # Επίπεδο IP (απο ethernet frame)
      size = len(raw_bytes) # Συνολικό πακέτο (ethernet + payload)

      # Μετατροπή IP bytes σε string (π.χ. b'\xc0\xa8\x01\x01' σε
      #'192.168.1.1')
      src_ip = socket.inet_ntoa(ip.src)
      dst_ip = socket.inet_ntoa(ip.dst)
```

Σε αυτό το στάδιο ουσιαστικά έχουμε τεμαχισμό των επιπέδων του πακέτου (δικτύου και μεταφοράς) έτσι ώστε να εξαχθούν οι βασικές πληροφορίες (οι διευθύνσεις IP προέλευσης και προορισμού, οι θύρες και το πρωτόκολλο επικοινωνίας όπως επίσης και το μέγεθος του πακέτου) και να δημιουργηθεί από αυτές ένα μοναδικό κλειδί ροής (flow key).

```
# Εξαγωγή TCP πεδίων (Απόσπασμα μεθόδου add_raw_packet)
# Αν τα δεδομένα είναι TCP πακέτο
if isinstance(ip.data, dpkt.tcp.TCP):
      tcp = ip.data # Αναφορά TCP αντικειμένου
      proto_name = "TCP" # Όνομα πρωτοκόλλου
      sport = tcp.sport # Θύρα πηγής
      dport = tcp.dport # Θύρα προορισμού
      payload_len = len(tcp.data) # Μήκος ωφέλιμου φορτίου
```

Όπως επίσης:

```
# Αν τα δεδομένα είναι UDP πακέτο
elif isinstance(ip.data, dpkt.udp.UDP):
```

```
udp = ip.data # Αναφορά UDP αντικειμένου
proto_name = "UDP" # Όνομα πρωτοκόλλου
sport = udp.sport # Θύρα πηγής
dport = udp.dport # Θύρα προορισμού
payload_len = len(udp.data) # Μήκος ωφέλιμου φορτίου
```

Το κλειδί ροής αντιστοιχεί στο συνδυασμό IP προέλευσης – Θύρα Προέλευσης – IP προορισμού – Θύρα προορισμού – Πρωτόκολλο επικοινωνίας και χρησιμοποιείται προκειμένου με αυτό το τρόπο να μπορούν να ομαδοποιηθούν τα πακέτα που ανήκουν στην ίδια σύνδεση.

```
# Δημιουργία κλειδιού (Canonical Key) - Bidirectional flow normalization
key = None # Αρχικοποίηση σε none
a = (src_ip, sport) # source flow ID (IP & θύρα πηγής)
b = (dst_ip, dport) # destination flow ID (IP & θύρα προορισμού)
if a <= b: # source flow ID <= destination flow ID
      # source-dest (canonical order)
      key = (src_ip, sport, dst_ip, dport, proto_name)
else: # Αντιστροφή για bidirectional μοναδικότητα
      key = (dst_ip, dport, src_ip, sport, proto_name)
```

Για τη κάθε ροή υπάρχει μια δομή δεδομένων στη μνήμη η οποία λαμβάνει ενημερώσεις σε πραγματικό χρόνο καθώς φτάνουν νέα πακέτα. Μέσα σε αυτή τη δομή αποθηκεύονται πληροφορίες όπως οι χρόνοι άφιξης των πακέτων, το μέγεθός τους, το σύνολο των Bytes που μετακινήθηκαν προς κάθε κατεύθυνση, και η κατάσταση των TCP σημαιών (SYN,ACK, FIN κλπ.). Καθώς οι μεταβλητές αυτές ενημερώνονται διαρκώς, διατηρείται από την FlowAggregator μια εικόνα πλήρης αναφορικά με την συμπεριφορά της ροής στο σύνολό της και η οποία εικόνα χρησιμοποιείται επίσης (σε επόμενο στάδιο) για τον υπολογισμό των στατιστικών χαρακτηριστικών. Επιπλέον, για τη περίπτωση του συνόλου δεδομένων UNSW, διατηρούνται και ορισμένοι παγκόσμιοι μετρητές όπως οι μετρήσεις συχνότητας υπηρεσιών ή συνδυασμών διευθύνσεων.

```
# Δημιουργία - Ενημέρωση ροής (Απόσπασμα μεθόδου add_raw_packet)
f = self.flows.get(key) # Ανάκτηση ροής
if f is None:  # Αν δεν υπάρχει στο λεξικό
      self.cnt_flows_created += 1 # Αύξηση μετρητή ροών
```

```
        f = { # Δημιουργία λεξικού ροής
        "first_ts": ts, # Χρονοσφραγίδα πρώτου πακέτου
        "last_ts": ts, #  Χρονοσφραγίδα τελευταίου πακέτου
        "first_detect_ts": None, # Χρονοσφραγίδα πρώτης ανίχνευσης
        "pkt_times": deque([ts]), # Ουρά χρόνων των πακέτων
        "pkt_lens": deque([size]), # Ουρά μεγεθών των πακέτων
        "ackdat_bytes": 0, # TCP ACK + data bytes
        "sloss": 0, # Μετρητής απώλειας πακέτων πηγής
        "dloss": 0, # Μετρητής απώλειας πακέτων προορισμού

              ... # Γραμμές κώδικα
        }
        self.flows[key] = f # Αποθήκευση ροής στο λεξικό
else:
        f["pkt_times"].append(ts) # Χρονοσφραγίδα
        f["pkt_lens"].append(size) # Μέγεθος πακέτου
        f["total_pkts"] += 1 # Μετρητής πακέτων
        f["total_bytes"] += size # Συνολικά Bytes
        f["last_ts"] = ts # Χρονοσφραγίδα τελευταίας δραστηριότητας

              ... # Γραμμές κώδικα

# Απόσπασμα μετρητών UNSW
# Μετρητής συνδέσεων (Dst IP + Service Port)
self.ct_srv_dst[(f["dst"], f["service"])] += 1
# Μετρητής συνδέσεων (Source IP + Service Port)
self.ct_srv_src[(f["src"], f["service"])] += 1
# Μετρητής αμφίδρομων ροών
self.ct_dst_src_ltm[(f["dst"], f["src"])] += 1
```

Ανά συγκεκριμένα χρονικά διαστήματα πραγματοποιείται η διαδικασία σύλληψης των ενεργών ροών κατά την οποία γίνεται ο υπολογισμός των χαρακτηριστικών της κάθε ροής. Κατά τη διαδικασία του υπολογισμού αρχικά προσδιορίζεται η διάρκεια της ροής, ενώ στη συνέχεια υπολογίζονται διάφορα στατιστικά μεγέθη (όπως Μ.Ο, ελάχιστα και μέγιστα μήκη πακέτων, τυπικές αποκλίσεις και χρόνοι μεταξύ των διαδοχικών πακέτων) από τις λίστες των πακέτων που έχουν συγκεντρωθεί.

```
# Μέθοδος δημιουργίας στατιστικών κι εκκαθάρισης
```

```
def snapshot_and_clear(self)

# Διάρκεια ροής σε δευτερόλεπτα (Απόσπασμα μεθόδου snapshot_and_clear)
dur_sec = max(f["last_ts"] - f["first_ts"], 1e-6)

# Forward Inter-Arrival Times (Απόσπασμα μεθόδου snapshot_and_clear)
fwd_iats = ( # Χρόνοι μεταξύ των αφίξεων
      [
            t2 - t1 # Διαφορά μεταξύ πακέτων
            for t1, t2 in zip( # Επανάληψη σε ζεύγη (zip)
                  # Όλα εκτός του τελευταίου κ όλα εκτός του πρώτου
                  fwd_times_micros[:-1], fwd_times_micros[1:]
            )
      ]
      # Αν έχει πακέτα
      if len(fwd_times_micros) > 1
      else [] # Άδειασμα λίστας
)
```

Με βάση τα μεγέθη αυτά δημιουργείται ένα ενιαίο λεξικό χαρακτηριστικών για κάθε ροή, μέσα στο οποίο περιλαμβάνονται χαρακτηριστικά που σχετίζονται με το σύνολο δεδομένων CIC-IDS2017 (όπως τα Flow Bytes/s, Packet Length Mean, Bwd Packet Length Max, Fwd IAT Std, PSH Flag Count και ACK Flag Count) αλλά και χαρακτηριστικά που αντιστοιχούν στο UNSW-NB15 (όπως τα sbytes, dbytes, state, service, tcprtt, sloss και dloss).

```
# Λεξικό χαρακτηριστικών ροών
features = {
      # UNSW Χαρακτηριστικά
      "srcip": f["src"], # Source IP address
      "dstip": f["dst"], # Destination IP address
      "proto": f["proto"], # Πρωτόκολλο μεταφοράς
      "sbytes": sbytes, # Σύνολο Bytes (πηγή - προορισμός)
      "dbytes": dbytes, # Σύνολο Bytes (προορισμός - πηγή)

            ... # Γραμμές κώδικα
```

```
        # CIC Χαρακτηριστικά
        # Μέγιστο μήκος επιστρεφόμενου πακέτου
        "Bwd Packet Length Max": _safe_max(bwd_lens),
        "PSH Flag Count": f.get("psh", 0), # Μετρητής PSH flag
        # ΜΟ μήκους όλων των πακέτων
        "Packet Length Mean": _safe_mean(all_lens),

            ... # Γραμμές κώδικα
```

Τα χαρακτηριστικά αυτά συγκεντρώνονται συνολικά σε μια λίστα εγγραφών που στη συνέχεια μετατρέπεται στο Pandas DataFrame, το οποίο θα αποτελέσει την τελική έξοδο της μονάδας συλλογής και επεξεργασίας ροών και το οποίο θα χρησιμοποιηθεί για τα επόμενα στάδια ανάλυσης.

```
# Προσθήκη λεξικού χαρακτηριστικών
rows.append(features)

# Δημιουργία DataFrame από λίστα λεξικών
return pd.DataFrame(rows)
```

Τα χαρακτηριστικά που εξάγονται θα πρέπει στην συνέχεια να αντιστοιχηθούν στη μορφή που απαιτούν τα εκπαιδευμένα μοντέλα. Σε αυτό το στάδιο της διαδικασίας επεμβαίνει η FeatureMapper η οποία ελέγχει (μέσω της ιδιότητας feature_names_in_) ποια είναι τα χαρακτηριστικά που απαιτούνται από το μοντέλο, προκειμένου να πραγματοποιήσει καθαρισμό των δεδομένων, διατηρώντας όλα τα χαρακτηριστικά που μπορούν να αποτελέσουν είσοδο στο μοντέλο και αφαιρώντας εκείνα που δεν αντιστοιχίζονται.

```
# Αποθήκευση της λίστας με τα ονόματα χαρακτηριστικών
# (Απόσπασμα κλάσης FeatureMapper)
feats_raw = (
        # Λήψη από XGBoost, αν δεν υπάρχει θέσε none
        getattr(model, "feature_names_in_", None) if model is not None
        else None
        )
        # Μετατροπή σε λίστα συμβολοσειράς (για συμβατότητα),
        # αν δεν υπάρχουν ονόματα χαρακτηριστικών θέσε κενή λίστα
        self.model_feats = [str(f) for f in feats_raw] if feats_raw is not
        None else []
```

```
      ... # Γραμμές κώδικα

# Καθαρισμός δεδομένων
df_clean = df.drop( # Νέο DataFrame
      columns=[ # Λίστα στηλών - χαρ/κών
            "srcip", # IP πηγής
            "dstip", # IP προορισμού
            "sport", # Θύρα πηγής
            "dsport", # Θύρα προορισμού
            "proto", # Πρωτόκολλο επικοινωνίας
            "Timestamp", # Χρονοσφραγίδα
            ],
            errors="ignore", # Αν κάποια στήλη δεν υπάρχει, αγνόησε
)
```

Παράλληλα, εάν υπάρχουν κατηγορικά χαρακτηριστικά (όπως για παράδειγμα proto, service ή state), τότε αυτά κωδικοποιούνται, μέσω των αντίστοιχων encoders, σε αριθμητική μορφή για να μπορέσουν να αξιοποιηθούν από τα μοντέλα μηχανικής μάθησης.

```
# Κωδικοποίηση κατηγορικών χαρακτηριστικών
def _encode_objects(self, X):
      # Αναζήτηση στηλών τύπου object (e.g. strings)
      for c in X.select_dtypes(include=["object"]).columns:
            X[c] = pd.factorize(X[c])[0] # Μετατροπή σε ακέραιο
      return X # Επιστροφή DataFrame
```

Υπάρχουν δύο τύποι μοντέλων ταξινόμησης που υποστηρίζονται από τον σύστημα και δεν είναι άλλοι από αυτόν που αφορά μοντέλα που βασίζονται στο σύνολο χαρακτηριστικών του CIC, όπως και αυτόν που αφορά μοντέλα που βασίζονται στο σύνολο χαρακτηριστικών του UNSW. Τα προ – επεξεργασμένα χαρακτηριστικά τροφοδοτούνται στα μοντέλα ταξινόμησης όπου, ανάλογα με τα χαρακτηριστικά που απαιτεί το καθένα, δημιουργείται το αντίστοιχο σύνολο εισόδου και πραγματοποιείται πρόβλεψη για κάθε ροή. Τα αποτελέσματα που θα προκύψουν από τα δύο μοντέλα μπορούν να λειτουργήσουν συνδυαστικά ώστε να παραχθεί μια τελική ένδειξη κανονικής ή κακόβουλης δραστηριότητας, η οποία ένδειξη στη συνέχεια χρησιμοποιείται για την ενημέρωση του συστήματος ανίχνευσης και καταγραφής συμβάντων.

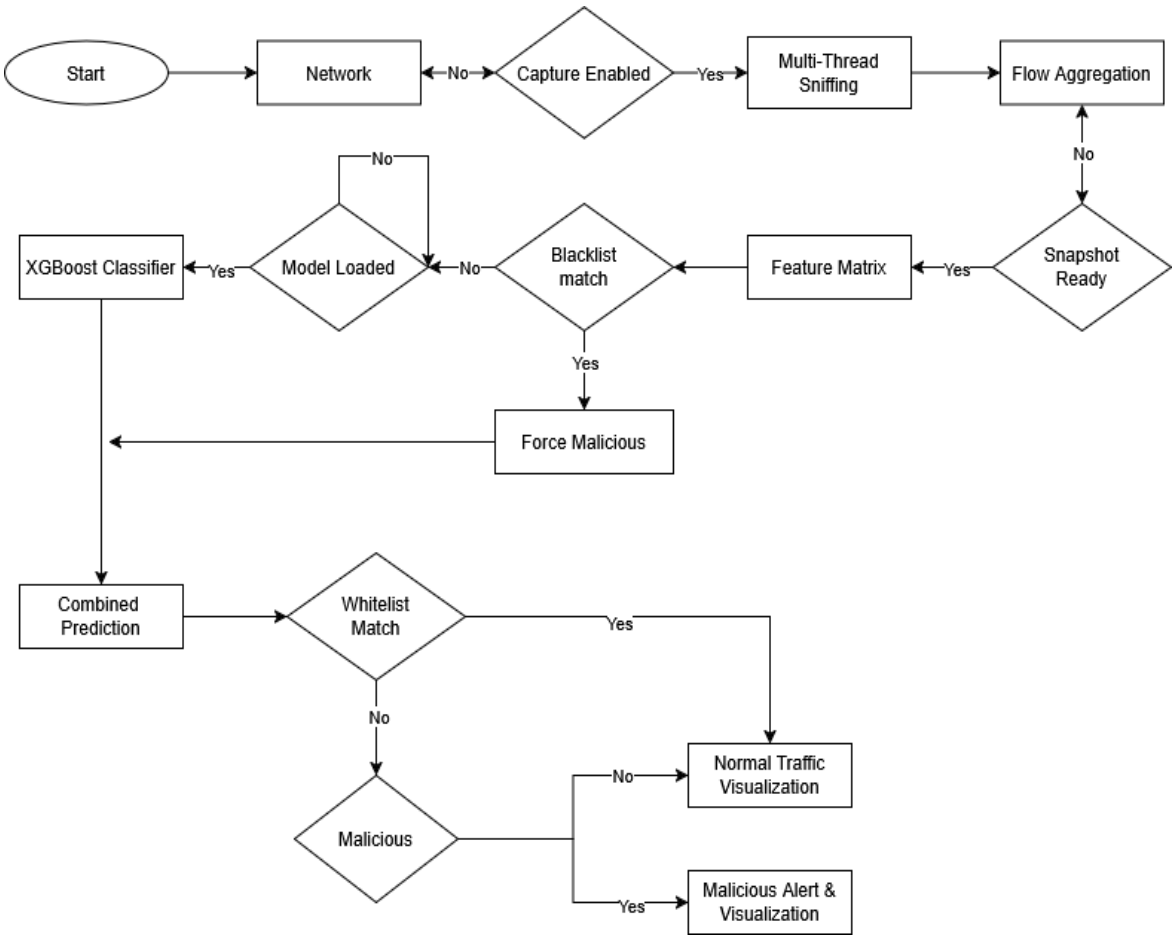
```
# Δημιουργία νέου DataFrame (με ίδιο δείκτη αντιστοίχισης όπως το αρχικό)
X_final = pd.DataFrame(index=df.index)
# Για κάθε όνομα στήλης
for col in self.model_feats:
      # Αν η στήλη υπάρχει στο df_clean παίρνει τις τιμές, αλλιώς 0
      X_final[col] = df_clean[col] if col in df_clean.columns else 0

... # Γραμμές κώδικα

# Πρόβλεψη επιθέσεων με το μοντέλο UNSW
y_pred_unsw = self.model_unsw.predict(X_unsw)
```

8.2.3. Διάγραμμα ροής δεδομένων

Το παρακάτω διάγραμμα [Σχήμα 1] οπτικοποιεί το τρόπο με τον οποίο γίνεται η επεξεργασία της κίνησης του δικτύου από τη στιγμή που συλλαμβάνονται τα πακέτα, μέχρι και την τελική απόφαση – κατηγοριοποίηση για το αν η ροή είναι κανονική ή κακόβουλη. Η ροή των δεδομένων αποτυπώνεται μέσα από επιμέρους στάδια και φαίνεται πως, ανάλογα με τη κατάσταση των συνθηκών, προκύπτει η αντίστοιχη έξοδος.



Σχήμα 1 Διάγραμμα ροής επεξεργασίας δικτυακής κίνησης για ανίχνευση κακόβουλων ροών

Αναλυτικότερα, κατά την εκκίνηση του συστήματος πραγματοποιείται η φόρτωση των μοντέλων μηχανικής μάθησης. Η παρακολούθηση της κίνησης του δικτύου ενεργοποιείται έπειτα από ενέργεια του χρήστη (start capturing) μέσω του GUI, εξασφαλίζοντας με αυτό το τρόπο ότι η διαδικασία σύλληψης πακέτων πραγματοποιείται μόνο κατ' επιλογήν. Η διαδικασία φόρτωσης των μοντέλων μηχανικής μάθησης πραγματοποιείται επίσης από το χρήστη μέσω του GUI.

Όταν η λειτουργία σύλληψης είναι ενεργή, τότε δημιουργείται ένα νήμα καταγραφής για κάθε διαθέσιμη δικτυακή διεπαφή (multi-thread sniffing), τα οποία συλλέγουν τα ακατέργαστα πακέτα από το σύνολο των δικτυακών διεπαφών που εντοπίζονται στο σύστημα. Στη συνέχεια, τα πακέτα αυτά μεταφέρονται στον συσσωρευτή ροών (Flow Aggregator) προκειμένου να μετατραπούν σε αμφίδρομες ροές και να υπολογιστούν τα στατιστικά χαρακτηριστικά.

Περιοδικά, σε τακτά χρονικά διαστήματα ο συσσωρευτής ροών δημιουργεί ένα στιγμιότυπο (snapshot) της τρέχουσας κίνησης του δικτύου, έπειτα το στιγμιότυπο αυτό μετατρέπεται σε δομή δεδομένων τύπου DataFrame και στη συνέχεια σε Feature Matrix, το οποίο αποτελεί τη δομημένη μορφή που αναμένουν τα μοντέλα μηχανικής μάθησης.

Πριν από την διαδικασία ταξινόμησης εφαρμόζεται έλεγχος προκειμένου να διαπιστωθεί εάν η διεύθυνση IP αποστολέα ή παραλήπτη είναι καταχωρημένη στη λίστα αποκλεισμού (Blacklist). Αν η διεύθυνση εμπεριέχεται στη λίστα η ροή χαρακτηρίζεται αυτόματα κακόβουλη χωρίς να απαιτείται επιπλέον πρόβλεψη από το μοντέλο.

Εφόσον τα μοντέλα είναι διαθέσιμα, το Feature Matrix προωθείται για ταξινόμηση στα δύο XGBoost μοντέλα έτσι ώστε να αποφασιστεί μέσω κοινής συνάρτησης (combined prediction) εάν η ροή είναι κανονική ή όχι (αν ένα μοντέλο την ταξινομήσει ως κακόβουλη, αυτό θα είναι και το τελικό αποτέλεσμα ανεξάρτητα από την απόφαση του δεύτερου). Εάν κάποιο μοντέλο δεν είναι διαθέσιμο τότε το αντίστοιχο στάδιο πρόβλεψης παραλείπεται, χωρίς να πραγματοποιηθεί επαναφόρτωση ή επανεκτέλεση της ταξινόμησης.

Μετά την πρόβλεψη εφαρμόζεται έλεγχος προκειμένου να διαπιστωθεί εάν η διεύθυνση IP αποστολέα ή παραλήπτη είναι καταχωρημένη στη λίστα επιτρεπόμενων διευθύνσεων (Whitelist). Αν εμπεριέχεται στη λίστα η ροή χαρακτηρίζεται αυτόματα ως κανονική ανεξάρτητα από το αποτέλεσμα του μοντέλου.

Τέλος, τα αποτελέσματα των προβλέψεων χρησιμοποιούνται για την κατηγοριοποίηση της κίνησης. Οι φυσιολογικές ροές απεικονίζονται στο γραφικό περιβάλλον ως «Normal Traffic», ενώ σε περίπτωση εντοπισμού κακόβουλης δραστηριότητας ενεργοποιούνται μηχανισμοί ειδοποίησης και ενημερώνονται τα σχετικά στοιχεία του dashboard με λεπτομέρειες όπως χρονική στιγμή, διευθύνσεις IP και θύρες επικοινωνίας.

8.3. Πρόβλεψη & διάγνωση: Αναγνώριση μοτίβων

Η διαδικασία της πρόβλεψης πραγματοποιείται εφόσον ολοκληρωθεί η εξαγωγή των χαρακτηριστικών των ροών, η δημιουργία του DataFrame, η εκκαθάριση δεδομένων, η μετατροπή κατηγορικών τιμών και η δημιουργία της τελικής δομής, που αποτελεί είσοδο των μοντέλων. Από το σύστημα υποστηρίζονται δύο είδη μοντέλων για τα οποία μπορεί να χρησιμοποιείται είτε μία σύνθεση περισσότερων υπό μοντέλων, όπως στην περίπτωση του συνόλου δεδομένων CIC-IDS2017, είτε ένα ενιαίο μοντέλο, όπως στην περίπτωση του συνόλου δεδομένων UNSW-NB15.

Για τη πρώτη περίπτωση έχουμε την πρόβλεψη με το μοντέλο που βασίζεται στο σύνολο δεδομένων CIC, κατά την οποία ο FeatureMapper δημιουργεί το υποσύνολο χαρακτηριστικών για να πραγματοποιηθεί η πρόβλεψη μέσω της predict.

```
# Υποσύνολο χαρακτηριστικών (Απόσπασμα μεθόδου on_snapshot)
# Λίστα με τα ακριβή ονόματα χαρακτηριστικών που περιμένει από το μοντέλο
expected_features = [
      # Δέχεται ονόματα χαρακτηριστικών, καθαρίζει τα κενά
      f.strip() for f in model.feature_names_in_
]

# Νέο DataFrame με τα χαρακτηριστικά του μοντέλου
X_specific = X_cic_raw.reindex(
      # Αν λείπει στήλη, γεμίζει με 0
      columns=expected_features, fill_value=0
)
```

Τα αποτελέσματα του CIC-IDS2017 μετατρέπονται κυρίως σε ετικέτες όπως BENIGN, DoS, Web Attack ή PortScan, ανάλογα με το αντίστοιχο υπό-μοντέλο που χρησιμοποιείται. Διαφορετικά μοντέλα είναι εκπαιδευμένα για την ανίχνευση συγκεκριμένων κατηγοριών

επιθέσεων όπως άρνησης υπηρεσιών (DoS) ή σάρωσης θυρών (PortScan). Για κάθε υπό μοντέλο δημιουργείται και από ένα υποσύνολο χαρακτηριστικών ώστε να εκτελεστεί η πρόβλεψη μέσω της predict. Τα αποτελέσματα που παράγονται μετατρέπονται σε ονομασίες επιθέσεων μέσω των αντίστοιχων LabelEncoder, έτσι ώστε να είναι κατανοητά σε κατηγορίες δραστηριότητας. Για τη δεύτερη περίπτωση, δηλαδή για το ενιαίο μοντέλο που βασίζεται στο σύνολο δεδομένων UNSW-NB15, τα αποτελέσματα της πρόβλεψης μετατρέπονται σε ετικέτες κατηγοριών όπως Normal, Generic, Exploits ή άλλες επιθέσεις που περιγράφονται από το σύνολο δεδομένων.

```
# Πρόβλεψη επιθέσεων για το μοντέλο CIC-IDS2017.
# Επιστρέφει 0 για κανονικό, 1 για επίθεση, 2 για DoS κλπ
y_pred = model.predict(X_specific.values)

if encoder: # Αν υπάρχει encoder
      # Μετατροπή αριθμών σε ονόματα (0=Benign, 1=DDoS κλπ.)
      names = encoder.inverse_transform(y_pred.astype(int))
else: # Αν δεν υπάρχει encoder
      names = y_pred.astype(str) # Κρατάει τους αριθμούς ως συμβολοσειρά

# Αναζήτηση γραμμών που δεν είναι κανονικές (Benign)
mask = names != "BENIGN"
if np.any(mask): # Αν βρει τέτοια γραμμή
      # γράφει το όνομα της επίθεσης στη στήλη attack_cic
      df.loc[mask, "attack_cic"] = names[mask]

      ... # Γραμμές κώδικα

# Προετοιμασία χαρακτηριστικών unsw
X_unsw, _ = self.mapper_unsw.prepare(df)
# Πρόβλεψη για το μοντέλο UNSW-NB15
y_pred_unsw = self.model_unsw.predict(X_unsw)
# Αποθήκευση προβλέψεων ως string στη νέα στήλη pred_unsw
df["pred_unsw"] = [str(p) for p in y_pred_unsw]
```

Μόλις ολοκληρωθεί η διαδικασία πρόβλεψης δημιουργούνται νέες στήλες στο DataFrame μέσω των οποίων αποθηκεύονται τα αποτελέσματα (ετικέτες επιθέσεων) των μοντέλων. Στη συνέχεια, προκειμένου να γίνει ο χαρακτηρισμός της ροής, υπολογίζεται μια τελική

ένδειξη κακόβουλης δραστηριότητας βασισμένη στην συνδυαστική αξιολόγηση των αποτελεσμάτων των δύο μοντέλων, σύμφωνα με την οποία εφόσον ένα από τα δύο μοντέλα επιστρέψει ετικέτα διαφορετική της κανονικής λειτουργίας (benign ή normal) η ροή χαρακτηρίζεται ως κακόβουλη και η τελική (δυαδική) πρόβλεψη ορίζεται ως ένδειξη επίθεσης.

```
# Αποθήκευση ετικετών
# Μετατροπή αριθμών (0,1...) σε ονόματα (Fuzzers, DoS...)
raw_names = self.le_unsw.inverse_transform(
# UNSW-NB15 LabelEncoder – Αντιστοίχιση ονομάτων
y_pred_unsw.astype(int)
)
# Αποθήκευση στη στήλη attack_unsw με κεφαλαίο το πρώτο γράμμα
df["attack_unsw"] = [str(x).capitalize() for x in raw_names]

      ... # Γραμμές κώδικα

# Τελική δυαδική πρόβλεψη. Συνδυάζει τις προβλέψεις των 2 μοντέλων
def combined_pred(row):
      # Δέχεται την πρόβλεψη του CIC, μετατροπή σε κεφαλαία
      cic = str(row.get("attack_cic", "")).upper()
      # Δέχεται την πρόβλεψη του UNSW, μετατροπή σε κεφαλαία
      unsw = str(row.get("attack_unsw", "")).upper()

      # Αν η CIC πρόβλεψη δεν είναι κανονική, κενή ή άγνωστη
      if cic not in ("BENIGN", "", "UNKNOWN"):
            return "1" # Επίθεση
      # Αν η UNSW πρόβλεψη δεν είναι κανονική, κενή ή άγνωστη
      if unsw not in ("NORMAL", "", "UNKNOWN"):
            return "1" # Επίθεση
      return "0" # Αλλιώς κανονική ροή
# Εφαρμογή αποτελεσμάτων στη στήλη prediction
df["prediction"] = df.apply(combined_pred, axis=1)
```

Παράλληλα, δημιουργείται και μια ενοποιημένη περιγραφή επίθεσης, η οποία συνδυάζει τις πληροφορίες των δύο μοντέλων ώστε να παρέχεται πληρέστερη εικόνα της συμπεριφοράς της ροής. Έτσι, εφόσον γίνει ο απαραίτητος έλεγχος ως προς το περιεχόμενο του κάθε

μοντέλου (προκειμένου να επιβεβαιωθεί αν όντως υπάρχει πρόβλεψη, αλλά και τι είδους πρόβλεψη) επιστρέφεται είτε μια λίστα με ένα ή περισσότερα ήδη επιθέσεων, είτε κατάσταση (status) κανονικής ροής.

```
# Ενοποίηση προβλέψεων
def merge_attacks(row):
      cic = row.get("attack_cic", "") # Πρόβλεψη CIC, κενό αν δεν υπάρχει
      unsw = row.get("attack_unsw", "") # Πρόβλεψη UNSW, κενό αν δεν υπάρχει
      parts = [] # Λίστα (κενή) για τα ονόματα επιθέσεων

      # Αν η CIC πρόβλεψη δεν είναι κανονική, άγνωστη ή κενή
      if cic not in ("BENIGN", "Unknown", "", None):
            parts.append(f"CIC → {cic}") # Προσθέτει CIC πρόβλεψη
      # Αν η UNSW πρόβλεψη δεν είναι κανονική, άγνωστη ή κενή
      if unsw not in ("Normal", "Unknown", "", None):
            parts.append(f"UNSW → {unsw}") # Προσθέτει UNSW πρόβλεψη

      if parts: # Εάν βρέθηκαν επιθέσεις
            return " | ".join(parts) # επιστροφή επιθέσεων
      else: # Αλλιώς
            return "Normal" # Επιστροφή normal
# Εφαρμογή αποτελεσμάτων στη στήλη attack_merged
df["attack_merged"] = df.apply(merge_attacks, axis=1)
```

Αν η διεύθυνση μιας ροής εμπεριέχεται στην καρτέλα IP Control της εφαρμογής, τότε η συμπεριφορά είναι διαφορετική. Πιο συγκεκριμένα, εάν μια ροή προέρχεται από αποστολέα ή παραλήπτη η IP διεύθυνση του οποίου βρίσκεται στην Blacklist, τότε το σύστημα χαρακτηρίζει τη ροή κακόβουλη, ανεξάρτητα από τα αποτελέσματα των μοντέλων ML. Αντίστοιχα, εάν μια ροή προέρχεται από αποστολέα ή παραλήπτη η IP διεύθυνση του οποίου βρίσκεται στην Whitelist, τότε το σύστημα χαρακτηρίζει τη ροή ως κανονική.

```
# Αναζήτηση γραμμών η IP όπου η srcip ή η dstip είναι στη whitelist
mask = df["srcip"].isin(self.whitelist) |
df["dstip"].isin(self.whitelist)
if mask.any(): # Αν υπάρχουν τέτοιες γραμμές
      df.loc[mask, "prediction"] = 0 # Prediction = 0
      df.loc[mask, "attack_cic"] = "BENIGN" # Ετικέτα Benign
```

```
        df.loc[mask, "attack_unsw"] = "Normal" # Ετικέτα Normal
        df.loc[mask, "attack_merged"] = "Normal" # Συνδυαστική ετικέτα
Normal
        df.loc[mask, "color_tag"] = "green" # Πράσινο χρώμα ροής
```

Πέραν της δυαδικής ταξινόμησης, υπάρχει ακόμα ένας μηχανισμός διάγνωσης κι ερμηνείας των αποτελεσμάτων, ο οποίος βασίζεται στην αναγνώριση προτύπων συμπεριφοράς. Για το σκοπό αυτό, υπάρχει μια διαδικασία αντιστοίχισης αποτελεσμάτων, η οποία εξετάζει τις ετικέτες που έχουν προκύψει από τα μοντέλα και τις μετατρέπει σε κατηγορίες επιθετικής συμπεριφοράς όπως DoS, PortScan κλπ.

```
# Αναγνώριση προτύπων - καθορισμός επίθεσης από συμπεριφορά
def _infer_attack_family(self, row):
        # Δέχεται CIC πρόβλεψη, αφαιρεί κενά
        cic = str(row.get("attack_cic", "")).strip()
        # Δέχεται UNSW πρόβλεψη, αφαιρεί κενά
        unsw = str(row.get("attack_unsw", "")).strip()

        if cic: # Εάν υπάρχει CIC πρόβλεψη
                cic_l = cic.lower() # Μετατροπή σε πεζά για ομοιόμορφη
        σύγκριση
        if cic_l in ( # Σύγκριση με οικογένεια επιθέσεων
                "dos hulk",
                "dos goldeneye",
                "dos slowloris",
                "dos slowhttptest",
                "heartbleed",
        ):
                return "dos" # Οικογένεια επιθέσεων DoS
        if "web attack" in cic_l: # Σύγκριση με web attack
                return "web" # Οικογένεια επιθέσεων Web
        if cic_l == "portscan":
                return "portscan"
```

Για κάθε κατηγορία επίθεσης έχει οριστεί συγκεκριμένο προφίλ συμπεριφοράς, στο οποίο περιλαμβάνονται χαρακτηριστικά που θεωρούνται ενδεικτικά για τη κάθε δραστηριότητα. Έτσι, για παράδειγμα, σε επιθέσεις άρνησης υπηρεσιών ενδεχομένως να παρατηρείται

υψηλός αριθμός πακέτων σε μικρό χρονικό διάστημα ή υψηλές τιμές στο χαρακτηριστικό του ρυθμού μετάδοσης δεδομένων. Το σύστημα συγκρίνει τις τιμές των χαρακτηριστικών της ροής με τα χαρακτηριστικά του κάθε προφίλ και δημιουργεί μια λίστα με ενδείξεις που εξηγούν γιατί η συγκεκριμένη ροή θεωρήθηκε κακόβουλη.

```
# Προφίλ επεξηγησιμότητας ανά οικογένεια επιθέσεων
def _xai_profile_for_family(self, family):
      profiles = { # Λεξικό προφίλ
            "dos": { # Προφίλ για DOS
                  "title": "DoS-like pattern", # Τίτλος για dashboard
                  "features": [ # Χαρακτηριστικά που δικαιολογούν το
            προφίλ
                        "Flow Bytes/s", # Πολύ υψηλή κίνηση
                        # Υψηλό σύνολο πακέτων προς server
                        "Total Fwd Packets",
                        "Total Backward Packets", # Πολύ λίγα acks απο
                  server
                        "SYN Flag Count", # Πλημμύρα SYN
                        "ACK Flag Count", # Χαμηλός ρυθμός ACK
                        "Packet Length Mean", # Μικρά πακέτα
                  ],
            },
            "fuzzers": { # Προφίλ για fuzzers


      ... # Γραμμές κώδικα


# Ενδείξεις που εξηγούν γιατί η ροή κρίθηκε κακόβουλη
reasons = [] # Δημιουργία λίστας
# Για κάθε Χαρακτηριστικό του προφίλ επίθεσης
for feat in prof["features"]:
      if feat in row and pd.notna(row[feat]): # Εφόσον υπάρχει στις
γραμμές
            val = row[feat] # Τιμή χαρακτηριστικού
            if isinstance(val, (int, float)): # Εαν είναι ακέραιο ή
      δεκαδικό
                  if abs(val) > 0: # Εάν είναι μεγαλύτερο του 0
                        # Πρόσθεσε "χαρακτηριστικό= xxx.xx"
                        reasons.append(f"{feat} = {val:.2f}")
            else: # Συμβολοσειρά
                  reasons.append(f"{feat} = {val}")
```

8.4. Υλοποίηση Λειτουργιών Εφαρμογής

Όπως παρουσιάστηκε στο κεφάλαιο 8, η διεπαφή χρήστη του Shield IDS αποτελείται από ξεχωριστές καρτέλες μέσα από τις οποίες παρέχεται ένας συνολικός έλεγχος του συστήματος, επιτρέποντας παράλληλα στον χρήστη να παρακολουθεί σε πραγματικό χρόνο τις ροές του δικτύου, να ερμηνεύει τις προβλέψεις των μοντέλων ML και να παρεμβαίνει άμεσα σε περιπτώσεις όπου υπάρχει πιθανός κίνδυνος.

Κάθε μια από τις καρτέλες εστιάζει σε συγκεκριμένο κομμάτι. Το κεντρικό Dashboard προσφέρει μια συνοπτική επισκόπηση με τη βοήθεια ημικυκλικών gauges που παρουσιάζουν ποσοστά κανονικής και κακόβουλης κίνησης, λίστα top remote hosts όπως επίσης και στατιστικά alerts, ενώ το Executive Dashboard περιλαμβάνει στρατηγικούς δείκτες απόδοσης (KPIs), οι οποίοι αποτυπώνουν συνολικά την κατάσταση ασφάλειας του συστήματος σε μορφή κατανοητή και κατάλληλη για στελέχη διοίκησης. Η καρτέλα Live Detection παρουσιάζει τις ροές σε πραγματικό χρόνο, με προβλέψεις από τα μοντέλα CIC και UNSW, ενώ το Explainability (XAI) παρέχει λεπτομερέστερη ανάλυση, που βασίζεται σε προφίλ μοτίβων συμπεριφοράς, συνεισφορά συγκεκριμένων χαρακτηριστικών καθώς επίσης και σε περιεχόμενο του MITRE ATT&CK. Στη καρτέλα Reports εμφανίζεται το ιστορικό των alerts με στατιστικά ανά τύπο επίθεσης και στο Log Analysis επιτρέπεται το φιλτράρισμα των logs σε συγκεκριμένα χρονικά εύρη όπως και η παρουσίαση προχωρημένων στατιστικών ανάλυσης, ενώ στη καρτέλα IP Control επιτρέπεται η διαχείριση Whitelist και Blacklist έτσι ώστε οι ροές να χαρακτηρίζονται ως κανονικές ή κακόβουλες σύμφωνα με την IP διεύθυνσή τους.

Η πλοήγηση γίνεται μέσω πλευρικής μπάρας, ενώ παράλληλα οι διαφορετικές λειτουργίες, όπως τα κουμπιά διακοπής – ενεργοποίησης δικτυακών διεπαφών ή οι επιλογές φόρτωσης μοντέλων, που ενσωματώνονται διευκολύνουν το χειρισμό και τη λήψη αποφάσεων. Σε αυτήν την ενότητα θα αναλυθεί ο τρόπος με τον οποίο υλοποιήθηκαν οι σημαντικότερες λειτουργίες για την κάθε καρτέλα.

8.4.1. Σχεδιασμός και Υλοποίηση Μενού

Η υλοποίηση των λειτουργιών του μενού (όπως αυτές περιγράφονται στο κεφάλαιο 8.1) πραγματοποιείται μέσω ενός συνδυασμού βιβλιοθηκών διεπαφής χρήστη, διαχείρισης αρχείων, φόρτωσης μοντέλων και πολυνηματικής επεξεργασίας. Στο επάνω μέρος της

εφαρμογής δημιουργείται ένα βασικό μενού λειτουργιών μέσω της κλάσης QMenuBar, στο οποίο προστίθενται τα επιμέρους μενού “Model” και “Capture”. Κάθε επιλογή του μενού υλοποιείται ως ένα αντικείμενο QAction, το οποίο συνδέεται με συγκεκριμένες συναρτήσεις μέσω σημάτων (signals), επιτρέποντας έτσι στον χρήστη να ενεργοποιεί τις λειτουργίες φόρτωσης μοντέλων ή έναρξης και διακοπής της καταγραφής της δικτυακής κίνησης. Με τον τρόπο αυτό υλοποιείται ο βασικός μηχανισμός παραμετροποίησης και ελέγχου της εφαρμογής.

```
# Απόσπασμα init_ui()
def init_ui(self):
      # Προσθήκη υπό μενού Model στη κύρια γραμμή μενού
      model_menu = menubar.addMenu("Model")

      # Δημιουργία υπό μενού για CIC-IDS2017 μοντέλα
      cic_menu = model_menu.addMenu("CIC-IDS2017")
      # Δημιουργία υπό μενού για UNSW-NB15 μοντέλα
      unsw_menu = model_menu.addMenu("UNSW-NB15")

      # Δημιουργία ενέργειας για φόρτωση cic μοντέλων
      load_cic = QAction("Load CIC Model", self)
      # Σύνδεση ενέργειας με κλικ ποντικιού
      load_cic.triggered.connect(self.load_cic_model)

      # Δημιουργία ενέργειας για φόρτωση cic μοντέλων
      load_unsw = QAction("Load UNSW Model", self)
      # Σύνδεση ενέργειας με κλικ ποντικιού
      load_unsw.triggered.connect(self.load_unsw_model)

      # Δημιουργία ενέργειας για έναρξη σύλληψης πακέτων
      start_action = QAction("Start Capture", self)
      # Δημιουργία ενέργειας για διακοπή σύλληψης πακέτων
      stop_action = QAction("Stop Capture", self)
      # Σύνδεση επιλογής (on click) με μέθοδο
      start_action.triggered.connect(self.start_capture)
      stop_action.triggered.connect(self.stop_capture)
```

Η φόρτωση των μοντέλων οργανώνεται σε ιεραρχικό μενού (submenus), στο οποίο οι κατηγορίες “CIC-IDS2017” και “UNSW-NB15” αντιστοιχούν σε διαφορετικά σύνολα δεδομένων. Για την περίπτωση του CIC-IDS2017, υποστηρίζεται από την εφαρμογή η φόρτωση πολλαπλών επιμέρους μοντέλων, τα οποία αντιστοιχούν σε διαφορετικές κατηγορίες επιθέσεων, όπως DoS, Web και PortScan. Η διαδικασία αυτή υλοποιείται με επαναλαμβανόμενη χρήση παραθύρων επιλογής αρχείων (QFileDialog), μέσω των οποίων ο χρήστης επιλέγει τα κατάλληλα αρχεία τύπου .pkl. Τα μοντέλα φορτώνονται στη μνήμη με χρήση της βιβλιοθήκης pickle, ενώ οι αντίστοιχοι label encoders φορτώνονται με τη βοήθεια της joblib και αποθηκεύονται σε δομές λεξικών (dictionaries), ώστε να είναι δυνατή η αντιστοίχιση κάθε μοντέλου με τον αντίστοιχο encoder του.

```
# Φόρτωση CIC μοντέλων – Απόσπασμα συνάρτησης load_cic_model()
def load_cic_model(self):
      dos, _ = QFileDialog.getOpenFileName( # Παράθυρο διαλόγου, επιλογή αρχείου
            # Τίτλος παραθύρου, φάκελος, κατάληξη αρχείου
            self, "Load CIC DoS Model", "", "Pickle Files (*.pkl)"
      )
      # Επιλογή αρχείου για web attack
      web, _ = QFileDialog.getOpenFileName(
            self, "Load CIC Web Attack Model", "", "Pickle Files (*.pkl)"
      )
      # Επιλογή αρχείου για portscan
      port, _ = QFileDialog.getOpenFileName(
            self, "Load CIC PortScan Model", "", "Pickle Files (*.pkl)"
      )
      # Φόρτωση μοντέλων από τα αρχεία pickle
      self.model_cic = {
            "dos": pickle.load(open(dos, "rb")), # DoS Attack
            "web": pickle.load(open(web, "rb")), # Web Attack
            "port": pickle.load(open(port, "rb")), # PortScan
      }
      # Φόρτωση των label encoders
      self.le_cic = {
            # Αν επιλέχθηκε αρχείο αντιστοιχεί στο DoS, αλλιώς καμία αντιστοίχιση
            "dos": joblib.load(dos_le) if dos_le else None,
            "web": joblib.load(web_le) if web_le else None, # Web Attack
```

```
        "port": joblib.load(port_le) if port_le else None, #
    PortScan
    }
```

Για το σύνολο δεδομένων UNSW-NB15, η υλοποίηση διαφοροποιείται, καθώς χρησιμοποιείται ένα ενιαίο πολυταξικό μοντέλο αντί των πολλαπλών επιμέρους μοντέλων. Το μοντέλο αυτό φορτώνεται επίσης μέσω pickle, ενώ ο label encoder φορτώνεται μέσω joblib. Επιπλέον, προβλέπεται η φόρτωση εξειδικευμένων encoders για κατηγορικά χαρακτηριστικά όπως τα proto, service και state. Οι encoders αυτοί αποθηκεύονται σε αντίστοιχες μεταβλητές της εφαρμογής και συνδέονται με τον μηχανισμό προεπεξεργασίαςς (mapper), εξασφαλίζοντας ότι τα δεδομένα εισόδου μετατρέπονται σωστά πριν από την εκτέλεση των προβλέψεων. Όλα τα φορτωμένα μοντέλα και οι σχετικοί μετασχηματιστές αποθηκεύονται ως μεταβλητές της κλάσης (instance attributes), ώστε να είναι άμεσα διαθέσιμα κατά τη διαδικασία ανίχνευσης.

```
# Φόρτωση πολυκλασικού μοντέλου unsw από αρχείο pickle
self.model_unsw = pickle.load(open(model_file, "rb"))
# Φόρτωση unsw label encoder αν υπάρχει
self.le_unsw = joblib.load(label_file) if label_file else None

# Encoder πρωτοκόλλων (για κωδικοποίηση TCP/UDP)
self.proto_enc = joblib.load(enc_proto) if enc_proto else None
# Encoder υπηρεσιών (eg HTTP)
self.service_enc = joblib.load(enc_service) if enc_service else None
# Encoder καταστάσεων (tcp states FIN/SYN etc.)
self.state_enc = joblib.load(enc_state) if enc_state else None

# FeatureMapper για προεπεξεργασίας UNSW features
self.mapper_unsw = FeatureMapper(self.model_unsw)
# Ρύθμιση των encoders για αυτόματη κωδικοποίηση κατηγορικών
χαρακτηριστικών
self.mapper_unsw.set_encoders(
      proto=self.proto_enc, # Protocol encoder για στήλη proto
      service=self.service_enc, # Service encoder για στήλη service
      state=self.state_enc, # State encoder για στήλη state
)
```

Το δεύτερο βασικό τμήμα του μενού αφορά τον έλεγχο της σύλληψης της δικτυακής κίνησης. Μέσω των επιλογών “Start Capture” και “Stop Capture”, ο χρήστης μπορεί να ξεκινήσει ή να διακόψει την παρακολούθηση του δικτύου. Κατά την έναρξη της διαδικασίας αυτής, η εφαρμογή εντοπίζει τις διαθέσιμες δικτυακές διεπαφές χρησιμοποιώντας τη συνάρτηση pcapy.findalldevs() και δημιουργεί ένα ξεχωριστό νήμα εκτέλεσης (thread) για κάθε διεπαφή, μέσω της κλάσης FastSniffThread. Η πολυνηματική αυτή προσέγγιση επιτρέπει την ταυτόχρονη παρακολούθηση πολλαπλών διεπαφών, αυξάνοντας την αποδοτικότητα και την κάλυψη της καταγραφής.

```
# Εκκίνηση διαδικασίας σύλληψης δικτυακών πακέτων
def start_capture(self):
      # Αν υπάρχουν ήδη ενεργά sniffer threads
      if hasattr(self, "sniffers") and any(s.isRunning() for s in
      self.sniffers):
            return # Τερματισμός συνάρτησης


      self.aggregator = FlowAggregator() # Αρχικοποίηση Aggregator
      self.capture_start_ts = time.time() # Χρόνος έναρξης καταγραφής
      self.first_attack_detected_ts = None # Αρχικοποίηση ανίχνευσης
      self.first_detection_delay = None # Αρχικοποίηση πρώτης
καθυστέρησης
      self.sniffers = [] # Λίστα για τα sniffer threads


      try: # Μπλοκ δικτυακών διεπαφών
            interfaces = pcapy.findalldevs() # Ανάκτηση διαθέσιμων
      διεπαφών
      except Exception as e: # Αν αποτύχει η ανάκτηση
            # Καταγραφή του σφάλματος στο log
            logging.error(f"Error", f"Could not list interfaces: {e}",
      exc_info=True)
            # Εμφάνιση προειδοποιητικού μηνύματος στον χρήστη
            QMessageBox.warning(self, "Error", f"Could not list
      interfaces: {e}")
            return # Τερματισμός διαδικασίας


      if not interfaces: # Αν η λίστα διεπαφών είναι άδεια
            QMessageBox.warning( # Εμφάνιση ενημερωτικού μηνύματος σε
      χρήστη
                  self, "No interfaces", "No network interfaces were
            found for capture."
```

```
            )
            return # Τερματισμός διαδικασίας

      active_sniffers = 0 # Μετρητής των threads

      for iface in interfaces: # Για κάθε διεπαφή της λίστας
            try: # Δημιουργία thread
            # Sniffer threads για τη συγκεκριμένη διεπαφή
                  sniffer = FastSniffThread(iface=iface,
                        aggregator=self.aggregator)
                  sniffer.start() # Έναρξη σύλληψης
                  # Αποθήκευση του thread στη λίστα
                  self.sniffers.append(sniffer)
                  active_sniffers += 1 # Ενημέρωση μετρητή
            except Exception as e: # Αποτυχία εκκίνησης διεπαφής
                  # Εμφάνιση ενημερωτικού μηνύματος στη κονσόλα
                  print(f"Could not start capture on {iface}: {e}")
                  # Καταγραφή του σφάλματος στο αρχείο log
                  logging.error(f"Could not start capture on {iface}:
                  {e}",
                        exc_info=True)

      # Αν δεν ξεκίνησε κάποιο sniffer thread
      if active_sniffers == 0:
            # Εμφάνιση προειδοποιητικού μηνύματος σε χρήστη
            QMessageBox.warning(
                  self,
                  "Capture Error", # Όνομα παραθύρου
                  # Μήνυμα ενημέρωσης, έλεγχος δικαιωμάτων
                  "Failed to start capture on any interface.\nCheck
                  permissions (Run as Admin).",
      )
      return # Διακοπή της διαδικασίας

      # Περιοδική ενημέρωση - Έναρξη QTimer με ανανέωση ανά 1
δευτερόλεπτο
      self.update_timer.start(1000)
      # Μετατροπή διαστήματος σε δευτερόλεπτα
      self.snapshot_interval = self.update_timer.interval() / 1000.0
```

```
        # Απενεργοποίηση κουμπιού Start Capture
        self.start_action.setEnabled(False)
        # Ενεργοποίηση κουμπιού Stop Capture
        self.stop_action.setEnabled(True)
```

Τα πακέτα που συλλέγονται από τα επιμέρους νήματα δεν επεξεργάζονται ανεξάρτητα, αλλά διοχετεύονται σε έναν κοινό μηχανισμό συγκέντρωσης και ομαδοποίησης, ο οποίος υλοποιείται μέσω της κλάσης FlowAggregator (ενότητα 9.2 Μονάδα Συλλογής και Επεξεργασίας Ροών). Ο μηχανισμός αυτός αναλαμβάνει τη δημιουργία και τη διαχείριση των δικτυακών ροών (network flows), συγκεντρώνοντας τα δεδομένα από όλες τις διεπαφές σε μία ενιαία δομή. Με αυτόν τον τρόπο είναι εφικτή η μεταγενέστερη εξαγωγή των χαρακτηριστικών όπως επίσης και η εφαρμογή των μοντέλων μηχανικής μάθησης, σε επίπεδο ροής και όχι σε επίπεδο μεμονωμένων πακέτων (Ανάλυση σχετικού κώδικα στην ενότητα 9.2 Μονάδα Συλλογής και Επεξεργασίας Ροών).

Όταν ο χρήστης επιλέξει τη διακοπή της σύλληψης, αποστέλλεται ένα σήμα διακοπής σε κάθε νήμα σύλληψης και τερματίζονται από την εφαρμογή όλοι οι ενεργοί μηχανισμοί καταγραφής. Παράλληλα, σταματά και ο χρονιστής ενημέρωσης (timer) της εφαρμογής, ενώ καθαρίζονται οι εσωτερικές δομές δεδομένων του FlowAggregator, όπως οι λίστες ροών και οι βοηθητικοί πίνακες καταμέτρησης. Η διαδικασία αυτή εξασφαλίζει ότι δεν θα συνεχιστεί η επεξεργασία πακέτων μετά τη διακοπή, αποτρέποντας ασυνέπειες ή σφάλματα.

```
# Διακοπή διαδικασίας σύλληψης δικτυακών πακέτων
def stop_capture(self):
        self.update_timer.stop() # Διακοπή Qtimer (Ενημέρωσης UI ανά sec)
        try: # Μπλόκ διαχείρισης threads & aggregator
                if hasattr(self, "sniffers"): # Αν υπάρχουν ενεργά threads
                        for sniffer in self.sniffers: # Για κάθε thread της
        λίστας
                                sniffer.stop() # Διακοπή του thread
                                sniffer.wait(500) # Αναμονή για ομαλή διακοπή
                                # Εκκαθάριση λίστας sniffer threads
                                self.sniffers = []

                if self.aggregator: # Αν υπάρχει (αντικείμενο) aggregator
                        # Αποκλεισμός thread για ασφαλή πρόσβαση στα δεδομένα
                        with self.aggregator.lock:
```

```
                    # Διαγραφή όλων των ενεργών δικτυακών ροών
                    self.aggregator.flows.clear()
                    # Εκκαθάριση μετρητών
                    # servers ανά προορισμό
                    self.aggregator.ct_srv_dst.clear()
                    # servers ανά πηγή
                    self.aggregator.ct_srv_src.clear()
                    # Καταστάσεις TCP (μέσω TTL)
                    self.aggregator.ct_state_ttl.clear()

        except Exception as e: # Σφάλμα κατά την προσπάθεια διακοπής
              # Εκτύπωση σφάλματος στη κονσόλα
              print(f"Error stopping capture: {e}")
              # Αναλυτική καταγραφή στο log file
              logging.error(f"Error stopping capture: {e}", exc_info=True)

        finally:
              # Επανεργοποίηση κουμπιού Start Capture
              self.start_action.setEnabled(True)
              # Απενεργοποίηση κουμπιού Stop Capture
              self.stop_action.setEnabled(False)
```

Τέλος, η εφαρμογή διαχειρίζεται δυναμικά την κατάσταση των επιλογών του μενού, προκειμένου να αποτρέψει λανθασμένες ενέργειες από τον χρήστη. Ειδικότερα, κατά την έναρξη της σύλληψης απενεργοποιείται η επιλογή “Start Capture” και ενεργοποιείται η “Stop Capture”, ενώ μετά τη διακοπή συμβαίνει το αντίστροφο. Επιπλέον, πριν την εκκίνηση ελέγχεται αν υπάρχουν ήδη ενεργά νήματα σύλληψης, ώστε να αποφευχθεί η πολλαπλή εκτέλεση της ίδιας διαδικασίας. Με αυτόν τον τρόπο διασφαλίζεται η ορθή λειτουργία της εφαρμογής και η αποφυγή αντικρουόμενων ενεργειών.

```
self.start_action.setEnabled(False) # Απενεργοποίηση κουμπιού Start
Capture
self.stop_action.setEnabled(True) # Ενεργοποίηση κουμπιού Stop Capture


stop_action.setEnabled(False) # Default state (UI init)


# Αν υπάρχουν ήδη ενεργά sniffer threads
```

```
if hasattr(self, "sniffers") and any(s.isRunning() for s in
self.sniffers):
      return # Τερματισμός συνάρτησης
```

8.4.2. Executive Dashboard

Η καρτέλα Executive Dashboard αποτελεί ένα σύνθετο ταμπλό (dashboard) εντός του οποίου συνδυάζονται στοιχεία διεπαφής χρήστη (UI), επεξεργασία δεδομένων και οπτικοποίηση των αποτελεσμάτων, για να μπορεί να παρέχει μια υψηλού επιπέδου συνολική εικόνα της ασφάλειας. Σε τεχνικό επίπεδο η καρτέλα βασίζεται στην κλάση ExecutiveDashboard (που είναι τύπου QWidget) η οποία λειτουργεί ως δοχείο (container) όλων των επιμέρους στοιχείων. Η ενημέρωση πραγματοποιείται δυναμικά μέσω των δεδομένων που προέρχονται από το σύστημα IDS.

```
# GUI Executive Dashboard – κληρονομεί από Qwidget
class ExecutiveDashboard(QWidget):
      # Constructor της κλάσης
      def __init__(self, parent=None): # Δέχεται προαιρετικά γονικό
widget
            super().__init__(parent) # Κλήση Constructor
            self.parent_app = parent # Αναφορά σε parent
```

Η κεφαλίδα (header) της καρτέλας φιλοξενεί την γενικότερη κατάσταση του συστήματος, τη δυνατότητα επιλογής ορισμένου χρονικού διαστήματος και τα βασικά κουμπιά της καρτέλας. Πιο συγκεκριμένα, περιλαμβάνει την ετικέτα (label) στην οποία αποτυπώνεται το επίπεδο ασφάλειας, δυο πεδία για επιλογή ημερομηνίας (από / έως) από τα οποία φιλτράρεται η χρονική περίοδος τα αποτελέσματα της οποίας αποτυπώνονται στη συνέχεια, καθώς και δύο κουμπιά ενεργειών. Το κουμπί ενημέρωσης «Update Dashboard» ενεργοποιεί τη μέθοδο request_update, μέσα από την οποία συγχρονίζονται τα δεδομένα από τον buffer του πίσω μέρους (backend) της εφαρμογής, ενώ στη συνέχεια ενεργοποιείται η διαδικασία επανυπολογισμού και ενημέρωσης των δεικτών της καρτέλας μέσω της μεθόδου ανάλυσης update_executive_data. Το κουμπί εξαγωγής «Export C-Level PDF» πραγματοποιεί κλήση στην handle_pdf_export η οποία δημιουργεί μια αναφορά μέσω της βιβλιοθήκης reportlab.

```
header = QHBoxLayout() # Header – Οριζόντια διάταξη
# Ετικέτα κατάστασης ασφαλείας (πράσινη για OPTIMAL)
```

```
self.risk_label = QLabel("Enterprise Security Posture: OPTIMAL")
self.risk_label.setStyleSheet( # CSS styling: 18pt, bold, πράσινα γράμματα
      "font-size: 18pt; font-weight: bold; color: #2ecc71;"
)
header.addWidget(self.risk_label) # Προσθήκη risk label
header.addStretch() # Ώθηση αντικειμένων προς τα δεξιά

self.date_from = QDateEdit(QtCore.QDate.currentDate()) # Date picker από
self.date_to = QDateEdit(QtCore.QDate.currentDate()) # Date picker έως
for d in [self.date_from, self.date_to]: # Για τις δύο ημερομηνίες
      d.setCalendarPopup(True) # Θέσε pop up ημερολόγιο

header.addWidget(QLabel("Period:")) # Ετικέτα Period
header.addWidget(self.date_from) # Date picker από
header.addWidget(self.date_to) # Date picker έως

apply_btn = QPushButton("Update Dashboard") # Κουμπί ενημέρωσης
apply_btn.setStyleSheet( # Μπλε φόντο, Λευκά έντονα γράμματα, 8px
      "background-color: #3498db; color: white; padding: 8px; font-
      weight: bold; border-radius: 5px;"
)
# Σύνδεση (on click) με μέθοδο request_update
apply_btn.clicked.connect(self.request_update)
header.addWidget(apply_btn) # Προσθήκη κουμπιού στο header

self.pdf_btn = QPushButton("Export C-Level PDF") # Κουμπί εξαγωγής PDF
self.pdf_btn.setStyleSheet( # Κόκκινο φόντο, λευκά έντονα γράμματα, 8px
      "background-color: #e74c3c; color: white; padding: 8px; font-
      weight:
            bold; border-radius: 5px;"
)
# Σύνδεση (on click) με μέθοδο handle_pdf_export
self.pdf_btn.clicked.connect(self.handle_pdf_export)
header.addWidget(self.pdf_btn) # Προσθήκη κουμπιού στο header
self.layout.addLayout(header) # Προσθήκη header στο layout

      ... # Γραμμές κώδικα

# Λειτουργία ενημέρωσης του Dashboard όταν πατηθεί το Update
```

```
def request_update(self):
      btn = self.sender() # Το κουμπί που έστειλε το σήμα
      if isinstance(btn, QPushButton): # Αν είναι QPushButton
            btn.setText("Updating...") # Αλλαγή κειμένου σε Updating...
            # Απενεργοποίηση κουμπιού (για αποφυγή spam clicks)
            btn.setEnabled(False)

      # Κλήση sync για νέα δεδομένα από τον buffer
      # Αν υπάρχει μέθοδος sync_data_buffer
      if hasattr(self.parent_app, "sync_data_buffer"):
            self.parent_app.sync_data_buffer() # Ενημέρωση δεδομένων
      if hasattr(self.parent_app, "all_logs"): # Αν υπάρχουν Logs
            # Ενημερώνει τα γραφήματα
            self.update_executive_data(self.parent_app.all_logs)
      if isinstance(btn, QPushButton): # Επαναφορά του κουμπιού
            QtCore.QTimer.singleShot( # Αρχική κατάσταση μετά απο 300ms
                  300, lambda: (btn.setText("Update Dashboard"),
            btn.setEnabled(True))
            )
```

Για την υλοποίηση των KPI καρτών χρησιμοποιήθηκε η μέθοδος create_kpi_card μέσα από την οποία δημιουργούνται επαναχρησιμοποιήσιμα UI στοιχεία με τίτλο, τιμή μέτρησης και χρώμα. Κάθε κάρτα αντιστοιχεί σε συγκεκριμένο metric (π.χ Cyber Resilience, Mean Detection Time κλπ.) κι ενημερώνεται δυναμικά μέσω αναζήτησης του QLabel με ObjectName “kpi_value”. Η τιμή που απεικονίζεται κάθε φορά, όπως το ποσοστό επιθέσεων ή ο μέσος χρόνος ανίχνευσης, δεν είναι στατική, ενημερώνεται έπειτα από σχετικούς υπολογισμούς στο update_executive_data. Επιπλέον, κάθε κάρτα περιλαμβάνει ένα info button, που ενεργοποιεί ένα επεξηγηματικό pop up μήνυμα μέσω mapping λεξικού (info_mapping).

```
# Δημιουργία κάρτας KPI
def create_kpi_card(self, title, value, color):
      card = QFrame() # Πλαίσιο κάρτας
      v = QLabel(value) # Ετικέτα κύριας τιμής
      # Μορφοποίηση
      v.setStyleSheet(f"color: {color}; font-size: 18pt; font-weight:
            bold;")
      v.setObjectName("kpi_value") # Όνομα αντικειμένου για css styling
```

```
        # Σύνδεση του κουμπιού info με μέθοδο show_metric_info
        info_btn.clicked.connect(lambda: self.show_metric_info(title))

def show_metric_info(self, title): # Μέθοδος pop up επεξήγησης μετρικών
        # Λήψη εξήγησης από λεξικό (default μήνυμα)
        explanation = self.info_mapping.get(title,
                "Πληροφορίες για αυτόν τον δείκτη.")
        # Pop Up με τίτλο κι επεξήγηση
        QMessageBox.information(self,
                f"Executive Insight: {title}", explanation)

        ... # Γραμμές κώδικα

self.info_mapping = { # Λεξικό επεξηγήσεων για C-Level
        # Επεξήγηση CRI
        "Cyber Resilience Index": "This is the overall health score of
        your security. 100% means the system is healthy and safe. If the
        score drops, it means the system is currently under stress from
        attacks.",
        # Endpoint Risk
        "Endpoint Risk": "Visualizes the density and intensity of attacks.
        A lower score indicates high 'Risk' from critical threats.",
        # Endpoint Threat Exposure – Επεξήγηση σε Multiline
        "Endpoint Threat Exposure": (
        "An indicator of the real-time risk level for this specific
host.\n\n"
        "• LOW: The system is secure. Network traffic patterns appear
        normal and non-threatening.\n"
        "• HIGH: The endpoint is experiencing targeted offensive activity
        (e.g., Exploits, DoS). "
        "There is an immediate risk to system integrity and resource
        availability."
        ),
        "Mean Detection Time": "The average time (MTTD) from the initial
        malicious flow appearance to detection. Critical for minimizing
        response time.", # MTTD metric
        "7-Day Risk Forecast": "Predictive trend analysis. Warns of an
        expected increase in offensive activity based on recent
        patterns.", # Forecasting
        "Critical Events Detected": "The total count of malicious events
        detected during the selected time period.", # Ειδοποιήσεις υψηλού
        επιπέδου
        "Visual Analysis": ( # Επεξηγήσεις Gauge
```

```
        "• Left Gauge: Measures how much 'clean' traffic we have. "
        "\n\n"
        "• Right Gauge: Measures how dangerous the attacks are. "
        ),
}
```

Το τμήμα στο οποίο περιλαμβάνονται τα γραφήματα (situational awareness) είναι βασισμένο σε ενσωμάτωση του Matplotlib μέσα στο PyQt μέσω του FigureCanvas. Για το σχεδιασμό των ημικυκλικών gauges χρησιμοποιείται η μέθοδος draw_gauge μέσα από την οποία δημιουργούνται δύο άξονες και γίνεται η μετατροπή των τιμών σε ποσοστά ώστε να απεικονίζονται ως pie charts με προσαρμοσμένα όρια και χρώματα. Παράλληλα, πραγματοποιείται κι έλεγχος της εγκυρότητας των τιμών για αποφυγή σφαλμάτων (πχ NaN, Infinity) όπως επίσης και δυναμική επιλογή χρώματος ανάλογα με το επίπεδο του ρίσκου, ενώ κάθε φορά που ενημερώνονται τα δεδομένα (μέσω canvas.draw_idle) γίνεται και η σχεδίαση εξασφαλίζοντας με αυτό το τρόπο οπτικοποίηση σε πραγματικό χρόνο (self.canvas.draw_idle()).

```
# Canvas Matplotlib 10x3.5 ιντσών, λευκό φόντο
self.canvas = FigureCanvas(Figure(figsize=(10, 3.5), facecolor="white"))
# 1ο γράφημα αριστερά (θέση 1 από 2)
self.ax_cyber = self.canvas.figure.add_subplot(121)
# 2ο γράφημα (θέση 2 απο 2)
self.ax_phys = self.canvas.figure.add_subplot(122)

# Ρυθμίσεις margins του Matplotlib για ολόκληρη εμφάνιση
self.canvas.figure.subplots_adjust(
# Αριστερά 5%, δεξιά 5%, πάνω 15%, κάτω 10%, απόσταση 20%
      left=0.05, right=0.95, top=0.85, bottom=0.1, wspace=0.2
)
# Προσθήκη canvas στο PyQt Layout
charts_main_layout.addWidget(self.canvas)
# Προσθήκη canvas στο κεντρικό Layout
self.layout.addWidget(charts_container)

      ... # Γραμμές κώδικα

# Συνάρτηση gauge meter (εμφάνιση ως ταχύμετρο)
```

```
def draw_gauge(self, ax, title, value):
ax.pie( # Matplotlib pie chart για gauge effect
      [safe_value, 100 - safe_value],
      colors=[color, "#ecf0f1"], # Χρώματα: [επικίνδυνο_χρώμα,
ουδέτερο_γκρι]
      startangle=180, # Ξεκινάει από κάτω (180° => 6 ώρα)
      counterclock=False, # δεξιόστροφη κατεύθυνση (ρολογιού)
)


      ... # Γραμμές κώδικα


try: # Προσπάθεια ελέγχου / καθορισμού τιμής
      # Αν είναι None / NaN
      if value is None or math.isnan(float(value)) or
            math.isinf(float(value)):
            value = 0.0 # Θέσε float 0.0
      # Αν η μετατροπή αποτύχει (π.χ. επειδή έχει string)
      except (ValueError, TypeError):
            value = 0.0  # Θέσε float 0.0
```

Ο υπολογισμός αποτελεσμάτων προκύπτει από την επεξεργασία των log αρχείων, η επεξεργασία των οποίων πραγματοποιείται με τη βοήθεια της μεθόδου update_executive_data. Εφόσον ο χρήστης επιλέξει στα σχετικά πεδία την ημερομηνία για την οποία ενδιαφέρεται (από / έως) τα δεδομένα φιλτράρονται και στη συνέχεια υπολογίζονται οι βασικοί δείκτες. Ο αριθμός των κακόβουλων ροών προκύπτει έπειτα από φιλτράρισμα συγκεκριμένων labels, ενώ το Cyber Resilience υπολογίζεται ως ποσοστό κανονικής δραστηριότητας προς το σύνολο των ροών. Παράλληλα, υπολογίζεται το endpoint heat score με βάση τη σοβαρότητα των επιθέσεων, όπου διαφορετικοί τύποι επιθέσεων έχουν διαφορετικό βάρος (critical vs reconnaissance).

```
# Απόσπασμα συνάρτησης update_executive_data.
# Ενημέρωση των KPI's - γραφημάτων με βάση τα logs
def update_executive_data(self, logs_df):
      # Αν δεν υπάρχουν logs ή το DatFrame είναι άδειο
      if logs_df is None or logs_df.empty:
                  return # Επιστροφή
      df = logs_df.copy() # Δημιουργία αντιγράφου δεδομένων
      # Μετατροπή στήλης timestamp σε pandas datetime (συμβατότητα)
```

```
df["Timestamp"] = pd.to_datetime(df["Timestamp"])
from_dt = datetime.combine(  # "Από Ημερομηνία" (00:00:00)
      self.date_from.date().toPyDate(), datetime.min.time()
)
# "Εως Ημερομηνία"
to_dt = datetime.combine(self.date_to.date().toPyDate(),
      datetime.max.time())
# Μάσκα (boolean) - Επιλογή logs εντός του ορισμένου
# χρονικού διαστήματος
mask = (df["Timestamp"] >= from_dt) & (df["Timestamp"] <= to_dt)
# Φιλτράρισμα του DataFrame με βάση τη μάσκα
filtered_df = df.loc[mask]

if filtered_df.empty: # Αν δεν υπάρχουν logs με αυτά τα φίλτρα
      # Επαναφορά dashboard σε αρχική κατάσταση
      self._reset_dashboard()
      return # Επιστροφή

#Detection Metrics (Endpoint Focus)
# Νέο DataFrame - Φιλτράρισμα μόνο των κακόβουλων εγγραφών
malicious = filtered_df[
# Κρατάει τις γραμμές στις οποίες περιέχεται το CIC ή το UNSW
      filtered_df["attack_merged"].str.contains("CIC|UNSW",
na=False)
]
threat_count = len(malicious)  # Καταμέτρηση κακόβουλων εγγραφών
self.total_detected = threat_count

# Cyber Resilience: Υπολογισμός δείκτη ανθεκτικότητας (0 έως 100)
self.cyber_resilience = max(
      # 100% αν δεν υπάρχουν επιθέσεις (Αύξηση
      # επιθέσεων - Μείωση ποσοστού)
      0.0, 100.0 * (1 - (threat_count / len(filtered_df)))
)

# Υπολογισμός κρισιμότητας βάσει της επικινδυνότητας
# των κατηγοριών CIC/UNSW
critical_heat = (
      malicious["attack_merged"] # Στήλη με τα ονόματα των
επιθέσεων
```

```
            # Αναζήτηση, αν περιέχονται τα παρακάτω (Critical Heat)
            .str.contains(
                  "DoS|Exploits|Bot|Infiltration|Backdoors|Worms|Shellco
                  de|
                        Web Attack",
                  na=False, # NaN τιμές αγνοούνται
            )
            .sum() # Πλήθος κρίσιμων επιθέσεων
      )

      # PortScans και Fuzzers προκαλούν "Medium Heat"
      recon_heat = (
            malicious["attack_merged"] # Στήλη με τα ονόματα των
      επιθέσεων
            # Αναζήτηση, αν περιέχονται τα παρακάτω (Medium Heat)
            .str.contains(
                  "PortScan|Fuzzers|Reconnaissance|Analysis|Generic",
            na=False
            )
             .sum() # Πλήθος medium επιθέσεων
      )

      # Υπολογισμός: Μικρότερο ανάμεσα σε 95
      # και (critical*2.5)+(medium*0.2)
      heat_penalty = min(95.0, (critical_heat * 2.5) +
            (recon_heat * 0.2))
      # Τελικό σκορ: 100 - heat_penalty
      self.endpoint_heat_score = 100.0 - heat_penalty
```

Η διαδικασία υπολογισμού πρόβλεψης κινδύνου (risk forecast), πραγματοποιείται με μια χρονική ανάλυση των τελευταίων 24 ωρών, όπου ουσιαστικά συγκρίνεται η πρόσφατη δραστηριότητα με το σύνολο των επιθέσεων. Αν η πρόσφατη δραστηριότητα ξεπερνάει το κατώφλι, τότε η κατάσταση λαμβάνει τον χαρακτηρισμό «INCREASING», διαφορετικά χαρακτηρίζεται ως «STABLE». Την ίδια στιγμή γίνεται και ο υπολογισμός του μέσου όρου ανίχνευσης (MTTD) από τη στήλη detection_delay. Όλα τα αποτελέσματα των μετρικών που προκύπτουν ενημερώνουν απευθείας το UI έτσι ώστε αφενός να υπάρχει αμεσότητα και αφετέρου να μπορεί ο χρήστης να έχει την εικόνα του ιστορικού και της πρόβλεψης.

```
# Risk Forecast (Απόσπασμα συνάρτησης update_executive_data())
# παράθυρο 7 ημερών με σημείο αναφοράς την επιλεγμένη τελική ημερομηνία
forecast_window_start = to_dt - timedelta(days=7)
# Φιλτράρισμα των εγγραφών που βρίσκονται μέσα στις τελευταίες 7 μέρες
forecast_df = df[(df["Timestamp"] >= forecast_window_start) &
(df["Timestamp"] <= to_dt)]
forecast_malicious = forecast_df[ # Υποσύνολο κακόβουλων εγγραφών
      # Μόνο όπου το attack_merged περιέχει CIC ή UNSW
      forecast_df["attack_merged"].str.contains("CIC|UNSW", na=False)
]
# Αρχή του πρόσφατου 24ώρου
recent_window_start = to_dt - timedelta(hours=24)
recent = len( # Σύνολο κακόβουλων εγγραφών
      forecast_malicious[ # Φιλτράρισμα κακόβουλων εγγραφών
            # Εγγραφές μετά την έναρξη του 24ώρου
            (forecast_malicious["Timestamp"] >= recent_window_start)
            # & έως τη τρέχουσα επιλογή
            & (forecast_malicious["Timestamp"] <= to_dt)
            ]
      )
# Μέσος ημερήσιος αριθμός κακόβουλων συμβάντων στο 7ήμερο
daily_avg = len(forecast_malicious) / 7.0
self.risk_forecast = ( # Δείκτης πρόβλεψης ρίσκου
      "STABLE" # Σταθερή κατάσταση
      # Αν το υποσύνολο κακόβουλων είναι άδειο
      if forecast_malicious.empty
      # Αλλιώς αύξηση, αν το πρόσφατο 24ωρο έχει περισσότερα απο το MO
      else ("INCREASING" if recent > daily_avg else "STABLE")
)
# Κόκκινο για αυξητική τάση, μωβ για σταθερή
f_color = "#e74c3c" if self.risk_forecast == "INCREASING" else "#9b59b6"

      ... # Γραμμές κώδικα

# Υπολογισμός μέσου χρόνου ανίχνευσης (MTTD)
if "detection_delay" in malicious.columns and not malicious.empty:
      # Αν υπάρχουν κακόβουλα συμβάντα υπολογίζει τον μέσο όρο
      self.mttd_val = malicious["detection_delay"].mean()
else:
```

```
      self.mttd_val = 0.0 # Αλλιώς δεν υπάρχουν δεδομένα (θέσε 0.0)
```

Το Decision Support Panel είναι ουσιαστικά το επίπεδο στο οποίο πραγματοποιείται μετάφραση των αριθμών σε επιχειρησιακή πληροφορία. Το κείμενο που παρουσιάζεται (intel_context) δημιουργείται κάθε φορά δυναμικά σύμφωνα με τα αποτελέσματα της ανάλυσης και βασίζεται ουσιαστικά στο συνολικό πλήθος των ανιχνευμένων απειλών, στον επικρατέστερο τύπο επίθεσης και στην τιμή του δείκτη έκθεσης Endpoint. Έτσι, για παράδειγμα, αν υπάρχουν επιθέσεις τότε εμφανίζεται κείμενο που εκφράζει πως θα πρέπει να υπάρξει άμεση παρέμβαση ενώ, αν δεν υπάρχουν επιθέσεις τότε εμφανίζεται κείμενο που εκφράζει κανονική λειτουργία.

```
 if threat_count > 0: # Εάν υπάρχουν επιθέσεις
      # Το πιο συχνό είδος επίθεσης
      top_type = malicious["attack_merged"].mode().iloc[0]
      self.intel_context = ( # Ενημέρωση (HTML Format)
      # Σύνολο και είδος επίθεσης
      f"<b>Risk Alert:</b> {critical_heat} critical attacks (
            {top_type}) detected. "
      # Ποσοστό heat score
      f"Endpoint risk exposure: {self.endpoint_heat_score:.0f}%. "
      "<b>Recommend immediate review and isolation</b>."
)
else: # Αν δεν υπάρχουν επιθέσεις, σχετικό μήνυμα
      self.intel_context = "<b>Executive Dashboard:</b> The endpoint is
            operating normally. No threats detected."

# Δυναμική ενημέρωση
# Ενημέρωση ετικέτας με objectName kpi_value (στο cyber card)
self.card_cyber.findChild(QLabel, "kpi_value").setText(
      # Εμφανίζει το Cyber Resilience score (με 1 δεκαδικό)
      f"{self.cyber_resilience:.1f}"
)
# Ενημέρωση ετικέτας με objectName kpi_value (στο cyber card)
self.card_mttd.findChild(QLabel, "kpi_value").setText(
      # Εμφανίζει το MTTD (με 3 δεκαδικά συν το s)
      f"{self.mttd_val:.3f}s"
)
```

Η διαδικασία εξαγωγής του PDF αποτελεί μια πλήρη αναφορά καθώς συλλέγει τα τρέχοντα αποτελέσματα μετρικών, μετατρέπει τα γραφήματα σε εικόνα (μέσω του buffer) και τα ενσωματώνει σε ένα PDF έγγραφο μαζί με μια λίστα KPI και ένα περιγραφικό κείμενο. Επιπλέον, προκειμένου η αναφορά να λειτουργεί ως ένα πλήρες εργαλείο παρακολούθησης αλλά και επικοινωνίας προς το διοικητικό επίπεδο, περιλαμβάνονται επίσης μεταδεδομένα όπως η ημερομηνία δημιουργίας ή το χρονικό διάστημα ανάλυσης.

```
# Συνάρτηση διαχείρισης εξαγωγής executive report σε PDF
def handle_pdf_export(self):
      # Παράθυρο διαλόγου "Αποθήκευση ως" για επιλογή path
      file_path, _ = QFileDialog.getSaveFileName(
            # Τίτλος παραθύρου, φίλτρο για αρχεία pdf
            self,
            "Save Executive Report",
            "",
            "PDF Files (*.pdf)",
      )

      if file_path:  # Εφόσον ορίστηκε το όνομα αρχείου κ η τοποθεσία
      # Εάν ο χρήστης δεν πρόσθεσε το .pdf στο τέλος
            if not file_path.endswith(".pdf"):
                  file_path += ".pdf"  # Προστίθεται αυτόματα

            try:  # Προσπάθεια δημιουργίας του PDF
                  # Συνάρτηση που δημιουργεί το PDF στο επιλεγμένο path
                  self.export_to_pdf(file_path)

                  # Μήνυμα επιτυχίας ολοκλήρωσης (popup)
                  QMessageBox.information(
                        self,
                        "Export Successful",
                        f"The Executive Report was successfully saved
                  to:\n\n{file_path}",
            )

            except Exception as e:  # Αν υπάρξει σφάλμα κατά την εξαγωγή
                  # Αναλυτική καταγραφή σφάλματος στο αρχείο log
                  logging.error(f"PDF Export failed: {e}",
            exc_info=True)
```

```
                # Μήνυμα σφάλματος κατά την εξαγωγή
                QMessageBox.critical(
                        self,
                        "Export Failed",
                        f"Could not save the PDF report.\n\nError
                details:\n{str(e)}",
                )

# Συνάρτηση που δημιουργεί το PDF στο επιλεγμένο path
def export_to_pdf(self, file_path):
        # Δημιουργία PDF canvas (σε μέγεθος A4)
        c = canvas.Canvas(file_path, pagesize=A4)
        w, h = A4

        # Τίτλος
        c.setFillColorRGB(0.1, 0.2, 0.4) # Σκούρο μπλε για το κείμενο
        c.setFont("Helvetica-Bold", 18) # Bold Helvetica, μέγεθος 18pt
        # Σημείο στη σελίδα: 50px από αριστερά, 50px από πάνω
        c.drawString(50, h - 50, "Executive Security Briefing")

        # Υπότιτλος με Ημερομηνία Δημιουργίας (Πότε πατήθηκε το κουμπί)
        c.setFillColorRGB(0.5, 0.5, 0.5) # Γκρι χρώμα
        c.setFont("Helvetica", 10) # Bold Helvetica, μέγεθος 10pt
        c.drawString( # Ημερομηνία/Ώρα δημιουργίας (Μεταδεδομένα)
        50, h - 70, f"Generated on: {datetime.now().strftime(
                '%d/%m/%Y %H:%M')}"
        )

        ... # Γραμμές κώδικα

        # Αποθήκευση γραφημάτων σε εικόνα
        img_buffer = BytesIO() # Εσωτερικός buffer
        self.canvas.figure.savefig( # Αποθήκευση matplotlib figure σαν
εικόνα
                # png, 150dpi, οριοθετημένα περιθώρεια, λευκό φόντο
                img_buffer, format="png", dpi=150, bbox_inches="tight",
                        facecolor="white"
        )
```

```
# Λίστα KPI Αποθήκευση
metrics = [  # Λίστα bullet με KPI's
      # Δείκτης υγείας του συστήματος (με 1 δεκαδικό)
      f"• Cyber Resilience (Health): {self.cyber_resilience:.1f}%",
      # Επίπεδο κινδύνου
      f"• Endpoint Threat Exposure: {'HIGH'
            if self.endpoint_heat_score < 50 else 'LOW'}",
      # Χρόνος ανίχνευσης (με 4 δεκαδικά)
      f"• Mean Time to Detect (MTTD): {self.mttd_val:.4f} sec",
      # Συνολικές επιθέσεις
      f"• Total Critical Events: {self.total_detected}",
      # Πρόβλεψη τάσης των επιθέσεων
      f"• 7-Day Forecast: {self.risk_forecast}",
      ]
```

8.4.3. Κεντρικό Dashboard

Η καρτέλα Dashboard υλοποιείται τεχνικά ως μία διεπαφή παρακολούθησης σε πραγματικό χρόνο που συνδυάζει δεδομένα από το IDS και παρέχει οπτικοποίηση και άμεση αλληλεπίδραση χρήστη. Σε αντίθεση με το Executive Dashboard που στοχεύει σε στρατηγικό επίπεδο, το Dashboard εστιάζει περισσότερο στη τακτική επίγνωση (situational awareness). Βασισμένο σε συνεχή ροή logs και στη γρήγορη επεξεργασία τους, αποσκοπεί στην προβολή της τρέχουσας κατάστασης (live) και στην απεικόνιση των δεδομένων με τέτοιο τρόπο ώστε και να είναι δυνατή η άμεση κατανόησή τους, αλλά και να παρέχεται (μέσα από όλη τη πληροφορία) η δυνατότητα πρόβλεψης του άμεσου μέλλοντος.

Αποτελεί το βασικό επιχειρησιακό επίπεδο του συστήματος Shield IDS και υλοποιείται με αρχιτεκτονική event-driven. Σε τεχνικό επίπεδο, βασίζεται στον κεντρικό μηχανισμό συγκέντρωσης δεδομένων (Aggregator), ο οποίος λαμβάνει συνεχόμενα ροές από τη διεπαφή δικτύου, τις επεξεργάζεται και διοχετεύει τις πληροφορίες στα επιμέρους στοιχεία του UI. Μέσω αυτής της προσέγγισης είναι εφικτή η συνεχής ενημέρωση των δεδομένων και η άμεση απεικόνισή τους, χωρίς να απαιτείται χειροκίνητη ανανέωση, δημιουργώντας έτσι ένα περιβάλλον πραγματικού χρόνου όπου κάθε αλλαγή στη κίνηση δικτύου παρουσιάζεται άμεσα στο Dashboard.

Η ενότητα Recent Alerts (Πρόσφατες Ειδοποιήσεις) υλοποιείται μέσω της κλάσης RecentAlertsCard και λειτουργεί ως υπεύθυνη αποθήκευσης και άντλησης των πιο

πρόσφατων συμβάντων. Τα δεδομένα διατηρούνται σε ένα pandas DataFrame (alerts_history), το οποίο ενημερώνεται συνεχώς με νέες εγγραφές. Κατά τη διάρκεια εισαγωγής των νέων δεδομένων, εφαρμόζεται φιλτράρισμα με βάση το Timestamp και τις IP διευθύνσεις (srcip, dstip), για την απομάκρυνση διπλότυπων ειδοποιήσεων, ενώ τα δεδομένα ταξινομούνται σε φθίνουσα σειρά ώστε τα πιο πρόσφατα να εμφανίζονται πρώτα. Το UI προσαρμόζεται δυναμικά στο μέγεθος του παραθύρου, εμφανίζοντας διαφορετικό αριθμό ειδοποιήσεων (για παράδειγμα 3 ή 7), εξασφαλίζοντας με αυτό το τρόπο τη καλύτερη αξιοποίηση του διαθέσιμου χώρου. Η κάθε ειδοποίηση αποτελεί ένα διαδραστικό στοιχείο (QWidget), όπου επιλέγοντάς το (on click) ενεργοποιείται ένα αναλυτικό παράθυρο (AlertDetailsDialog) που παρουσιάζει όλα τα δεδομένα της συγκεκριμένης ροής, προσφέροντας άμεση πρόσβαση σε σημαντικές λεπτομέρειες.

```
# Απόσπασμα κλάσης RecentAlertsCard
# Προσθήκη νέων εγγραφών στο ιστορικό
self.alerts_history = pd.concat( # Ένωση παλαιότερων alerts με νέα
      # Αγνοεί παλιά indexes, δημιουργία νέων - Συνεχόμενη αρίθμηση
      [self.alerts_history, alerts_df], ignore_index=True
)
# Διαγραφή διπλότυπων alerts απο το ιστορικό
self.alerts_history.drop_duplicates(
      # Διατήρηση τελευταίας εγγραφής (μοναδικότητα χρόνου, Ip)
      subset=["Timestamp", "srcip", "dstip"], keep="last", inplace=True
)
# Έλεγχος παράθυρου εφαρμογής
if self.window().height() < 1000:
      max_alerts = 3 # Αν είναι μικρότερο, εμφάνισε 3 Alerts
else: # Αλλιώς εμφάνισε 7
      max_alerts = 7
# Ταξινόμιση των alert ανά ημερομηνία
self.alerts_history = self.alerts_history.sort_values(
      "Timestamp", ascending=False # Πιο πρόσφατα πρώτα (Descending)
).head(max_alerts)

      ... # Γραμμές κώδικα

      # --- Αναλυτικό παράθυρο (on click) για κάθε γραμμή
      def show_details(event, row=row):
```

```
        # Αν η γραμμή έχει το to_dict (μέθοδος του Panda series)
        if hasattr(row, "to_dict"):
              # Μετατροπή όλων των στηλών σε dictionary
              alert_dict = row.to_dict()
        else: # Αλλιώς
              # Είναι ήδη dictionary ή απλά δομή
              alert_dict = dict(row)
        # Pop up με alert (αναφορά στο γονικό παράθυρο)
        dlg = AlertDetailsDialog(alert_dict, self)
        dlg.exec_() # Παράθυρο διαλόγου (OK / Cancel)
```

Η ενότητα Explainability Preview (XAI) προσθέτει ένα ιδιαίτερα σημαντικό επίπεδο κατανόησης των αποφάσεων του συστήματος. Δημιουργείται αρχικά μέσω της create_dashboard_page() ως μια απλή κάρτα με placeholder μήνυμα, ενώ η βασική υλοποίηση της συνοπτικής προεπισκόπησης γίνεται μέσω της update_xai_preview(), που εμφανίζει συνοπτική πληροφορία.

```
# Απόσπασμα συνάρτησης create_dashboard_page()
# Δημιουργία κενής κάρτας XAI
card3 = self.make_card("Explainability Preview", "")
# Ετικέτα placeholder
self.xai_label = QLabel("Waiting for model activity...")
# Μικρό, γκρί κείμενο
self.xai_label.setStyleSheet("font-size: 10pt; color: #555;")
card3.layout().addWidget(self.xai_label) # Προσθήκη ετικέτας
layout.addWidget(card3, 1, 0) # Θέση [1, 0] πλέγματος

      ... # Γραμμές κώδικα

# Ενημέρωση XAI preview κάρτας με εξήγηση πρόβλεψης του μοντέλου
def update_xai_preview(self, prediction, top_features=None):
      # Αν η πρόβλεψη είναι κακόβουλη
      if prediction in ("malicious", "attack", 1):
            text = ( # HTML => κόκκινη προειδοποίηση - Bold
                  "<b style='color:#D32F2F;'>Malicious traffic
                        detected</b><br>"
                  "The model flagged this flow as suspicious.<br>"
            )
```

```
            if top_features: # Αν υπάρχουν Top χαρακτηριστικά από SHAP
                  # Τα πρώτα 3 χαρακτηριστικά
                  text += f"Key contributing features: {',
                        '.join(top_features[:3])}"
      else: # Αν η πρόβλεψη δεν είναι κακόβουλη
            text = ( # HTML => πράσινη ενημέρωση - Bold
                  "<b style='color:#2E7D32;'>Normal traffic</b><br>"
                  "No significant anomalies detected."
            )

      self.last_xai_text = text # Αποθήκευση τελευταίου κειμένου
      self.xai_label.setText(text) # Ενημέρωση ετικέτας
```

Στη συνέχεια επεκτείνεται μέσω της update_xai_preview_from_row() που ενσωματώνει πραγματική επεξηγησιμότητα, χρησιμοποιώντας SHAP κι επιλέγοντας τον κατάλληλο explainer ανάλογα με το ενεργό μοντέλο (CIC ή UNSW), υπολογίζοντας έτσι τις τιμές SHAP για τη συγκεκριμένη ροή κι εξάγοντας τα σημαντικότερα χαρακτηριστικά. Κατά τη διαδικασία αυτή διατηρούνται μόνο τα χαρακτηριστικά με θετική συνεισφορά στην πρόβλεψη, ταξινομούνται κατά σημαντικότητα κι επιστρέφονται, μέσω της βοηθητικής συνάρτησης extract_top_shap_for_row(), τα top 3, τα οποία ενσωματώνονται σε ένα ενιαίο κείμενο, μαζί με πληροφορίες για το είδος της επίθεσης, τη ροή και το προφίλ συμπεριφοράς. Σε περίπτωση που δεν αποδοθεί έγκυρο αποτέλεσμα μέσω του SHAP, χρησιμοποιείται συγκεκριμένος μηχανισμός ο οποίος παρέχει μια ευρετική εξήγηση, ενώ η ενημέρωση του preview γίνεται είτε αυτόματα, κατά τη ροή των predictions, είτε με επιλογή της γραμμής (on click) από τον χρήστη.

```
# Επεξήγηση για κάθε γραμμή log / alert
def update_xai_preview_from_row(self, row):
      # Τύπος επίθεσης απο τις δύο πιθανές στήλες (cic & unsw)
      val = row.get("attack_merged") or row.get("Attack Type")
            or "Unclassified"
      attack_merged = str(val) # Μετατροπή σε συμβολοσειρά
      src = row.get("srcip", "?") # Source IP ή ? αν δεν διατίθεται
      dst = row.get("dstip", "?") # Destination IP ή ? αν δεν διατίθεται
      # Αν πρόκειται για κανονική κίνηση
      if attack_merged.lower() == "normal":
            text = ( # Γράψε Normal traffic με πράσινα - έντονα
                  "<b style='color:#2E7D32;'>Normal traffic</b><br>"
```

```
                "No significant anomalies detected."
            )
            self.last_xai_text = text # Αποθήκευση για
επαναχρησιμοποίηση
            self.xai_label.setText(text) # Εμφάνιση στο UI
            return # Έξοδος
# Καθορισμός οικογένειας επίθεσης (DoS, PortScan etc)
family = self._infer_attack_family(row)
# Δέχεται τα χαρακτηριστικά του προφίλ της οικογένειας
prof = self._xai_profile_for_family(family)

reasons = [] # Λίστα επεξηγήσεων
used_shap = False # Δεν χρησιμοποιείται SHAP (flag)

try: # Προσπάθεια επεξήγησης με SHAP
            # Μετατροπή σειράς σε μια σειρά ανά DataFrame
            row_df = pd.DataFrame([row.to_dict() if hasattr(row,
"to_dict")
                  else dict(row)])
            shap_pairs = [] # Λίστα χαρακτηριστικό - τιμή SHAP
            # πρόβλεψη για το μοντέλο cic
            cic = str(row.get("attack_cic") or "BENIGN").strip()
            # πρόβλεψη unsw
            unsw = str(row.get("attack_unsw") or "Normal").strip()
            # Αν το unsw βρήκε επίθεση και υπάρχει μοντέλο unsw
            if unsw.lower() != "normal" and self.model_unsw is not None:
                  # Προετοιμασία των χαρακτηριστικών για το μοντέλο unsw
                  X_unsw, _ = self.mapper_unsw.prepare(row_df)
                  # Επεξήγηση SHAP για το μοντέλο unsw
                  explainer = self.shap_explainers.get("unsw")

                  if explainer is not None: # Αν υπάρχει SHAP επεξήγηση
                        # TOP 3 χαρακτηριστικά
                        shap_pairs = self._extract_top_shap_for_row(
                              explainer=explainer, # Επεξήγηση
                              model=self.model_unsw, # Μοντέλο
                              X_row=X_unsw, # Γραμμή
                              top_n=3,
                        )
            # Αλλιώς, αν το cic βρήκε επίθεση
```

```
elif cic.lower() != "benign"
      and isinstance(self.model_cic, dict):
      # Προετοιμασία χαρακτηριστικών cic
      X_cic_raw, _ = self.mapper_cic.prepare(row_df)
      # Εκκαθάριση των ονομάτων των στηλών
      X_cic_raw.columns = [str(col).strip() for col
            in X_cic_raw.columns]
      # Αρχικοποίηση, μόνο αριθμητικές στήλες
      X_cic_raw = X_cic_raw.select_dtypes(
            include=["int64", "float64", "bool"])

      cic_l = cic.lower() # Μετατροπή σε πεζά (matching)
      model = None # Αρχικοποίηση, για cic μοντέλο
      explainer = None # Αρχικοποίηση, για SHAP επεξήγηση
      # Αν είναι dos ή heartbleed
      if "dos" in cic_l or "heartbleed" in cic_l:
            model = self.model_cic.get("dos") # Μοντέλο DoS
            # Επεξήγηση DoS
            explainer = self.shap_explainers.get("cic_dos")
      elif "web attack" in cic_l: # Αν είναι web attack
            model = self.model_cic.get("web") # Μοντέλο web
            # Επεξήγηση cic_web
            explainer = self.shap_explainers.get("cic_web")
      elif "portscan" in cic_l: # Αν είναι portscan
            model = self.model_cic.get("port") # Μοντέλο
port
            # Επεξήγηση cic_port
            explainer = self.shap_explainers.get("cic_port")
      # Αν βρέθηκε το αντίστοιχο μοντέλο
      if model is not None and explainer is not None:
            # Συγκεκριμένα χαρακτηριστικά που αναμένονται
            expected_features = [f.strip() for f in
                  model.feature_names_in_]
            # Στοίχιση στηλών, ανάλογα με τα χαρακτηριστικά
            X_row = X_cic_raw.reindex(columns=
                  expected_features, fill_value=0)
            # SHAP Ανάλυση
            shap_pairs = self._extract_top_shap_for_row(
                  explainer=explainer, # Επεξήγηση
```

```
                        model=model, # Μοντέλο
                        X_row=X_row, # Γραμμή
                        top_n=3, # TOP 3 χαρακτηριστικά
                    )

        if shap_pairs: # Αν υπήρξε SHAP επεξήγηση
            for feat, shap_val in shap_pairs: # Για κάθε
        χαρακτηριστικό
                # Προσθήκη χαρακτηριστικού, θετική τιμή 4ων
            δεκαδικών
                reasons.append(f"{feat} = +{shap_val:.4f}")
            used_shap = True # Χρησιμοποιήθηκε SHAP (flag)
    # Αν η προσπάθεια επεξήγησης με SHAP αποτύχει
    except Exception as e:
        # Καταγραφή σφάλματος - χωρίς να επιρεαστεί το UI
        logging.error(f"Preview SHAP error: {e}", exc_info=True)

    if not used_shap: # Αν δεν χρησιμοποιήθηκε SHAP
        for feat in prof["features"]: # Για τα χαρακτηριστικά του
    προφίλ
            # Αν υπάρχουν - Και δεν είναι NaN
            if feat in row and pd.notna(row[feat]):
                val = row[feat] # Τιμή χαρακτηριστικού
                    # Αν η τιμή είναι αριθμητική
                    if isinstance(val, (int, float)):
                        # Αν είναι μεγαλύτερη του 0
                        if abs(val) > 0:
                            # Προσθήκη χαρ/κού, 2 δεκαδικά
                            reasons.append(f"{feat} =
                                {val:.2f}")
                    else: # Αλλιώς, καταχώριση τιμής
                        reasons.append(f"{feat} = {val}")
                # Διατήρηση των πρώτων 3ων
                if len(reasons) == 3:
                    break

    if not reasons: # Αν δεν βρέθηκαν επεξηγήσεις
        # Default επεξήγηση
        reasons.append("abnormal traffic characteristics")
```

```
      trigger_title = ( # Διαφορετικός τίτλος ανάλογα με τη συμπεριφορά
            "Top 3 SHAP features:"
            if used_shap # Επεξήγηση SHAP
            else "Top 3 behavior indicators:"
      )

      text = ( # HTML πρότυπο επεξήγησης - Κόκκινο
            "<b style='color:#D32F2F;'>Malicious traffic
      detected</b><br>"
            # Κόκκινο (ανοιχτό), έντονο - Επίθεση
            f"<span style='color:#555;'>Type:</span>
                  <b>{attack_merged}</b><br>"
            # Κόκκινο (ανοιχτό), έντονο - Οικογένεια (προφίλ) επίθεσης
            f"<span style='color:#555;'>Behavior profile:</span>
                  {prof['title']}<br>"
            # Κόκκινο (ανοιχτό), έντονο - IPs πηγής κ' προορισμού
            f"<span style='color:#555;'>Flow:</span> {src} →
      {dst}<br><br>"
            # Κόκκινο (ανοιχτό), έντονο - Τίτλος εξήγησης
            f"<span style='color:#555;'>{trigger_title}</span><br>"
            + "<br>".join(reasons[:3]) # Top 3 λόγοι
      )

      self.last_xai_text = text # Αποθήκευση για επαναχρησιμοποίηση
      self.xai_label.setText(text) # Εμφάνιση στο UI
```

Η ενότητες Network Activity και Live Traffic επικεντρώνονται στην παρακολούθηση της ροής δεδομένων του δικτύου. Η Network Activity υλοποιείται με τη χρήση της βιβλιοθήκης psutil, η οποία συλλέγει στατιστικά στοιχεία από τις κάρτες δικτύου (io_counters). Το σύστημα υπολογίζει τη μεταβολή των bytes ανά δευτερόλεπτο και τη μετατρέπει σε MB/s, προσφέροντας έτσι μια άμεση ένδειξη της δραστηριότητας.

```
# Συνάρτηση ενημέρωσης δραστηριότητας δικτυακής κίνησης
def update_network_activity(self):
      try:
            # Πάρε ενεργά interfaces
            interfaces = psutil.net_if_stats()
            active = sum(1 for i in interfaces.values() if i.isup)
```

```
# Λήψη συνολικών bytes sent/received
counters = psutil.net_io_counters(pernic=False)
# Bytes που έχουν σταλεί
bytes_sent = counters.bytes_sent
# Bytes που έχουν ληφθεί
bytes_recv = counters.bytes_recv

# Αν είναι η πρώτη φορά (δεν έχει προηγούμενα δεδομένα)
if not hasattr(self, "_last_net"):
      # Αρχικοποίηση με τρέχουσες τιμές
      self._last_net = (bytes_sent, bytes_recv, time.time())
      return
now = time.time() # Τρέχουσα χρονική στιγμή
dt = now - last_time # Χρόνος από τυελευταία μέτρηση
if dt <= 0: # Αν πέρασε μηδενικός/αρνητικός χρόνος
      return # επιστροφή

# Υπολογισμός ρυθμού σε bytes/sec
sent_rate = (bytes_sent - last_sent) / dt # Μεταφόρτωση
recv_rate = (bytes_recv - last_recv) / dt # Λήψη
# Ενημέρωση τελευταίας μέτρησης
self._last_net = (bytes_sent, bytes_recv, now)

# Μορφοποίηση από bytes/sec σε MB/s
def fmt(bps):
      # MB συν 2 δεκαδικά
      return f"{bps / (1024*1024):.2f} MB/s"
```

Την ίδια στιγμή, το Live Traffic απεικονίζεται γραφικά μέσω της κλάσης TrafficChart, η οποία ενσωματώνει Matplotlib σε περιβάλλον PyQt. Τα δεδομένα αποθηκεύονται σε μια δομή (deque) που διατηρεί ιστορικό 2 περίπου λεπτών, επιτρέποντας τη συνεχή σχεδίαση της καμπύλης κίνησης σε πραγματικό χρόνο. Συμπληρωματικά, η λειτουργία Top Talkers αναλύει ένα λεξικό αντιστοίχισης IP διευθύνσεων και όγκου δεδομένων, ταξινομεί τις εγγραφές και εμφανίζει τους πιο ενεργούς κόμβους του δικτύου, προσφέροντας επιπλέον ορατότητα στη δραστηριότητα.

```
# Matplotlib canvas για real-time γράφημα κίνησης δικτύου
class TrafficChart(FigureCanvas):
```

```
      def __init__(self, parent=None, max_points=120): # Ιστορικό 2
λεπτών
            fig = Figure(figsize=(8, 3), dpi=100) # 8 * 3 ίντσες, 100
      dpi
            self.ax = fig.add_subplot(111) # 1 subplot σε πλέγμα 1 * 1
            super().__init__(fig) # κλήση constructor
            self.setParent(parent) # Σύνδεση με Qt parent
            self.max_points = max_points # Χρήση μέγιστου αριθμού
      σημείων
            self.x = deque(maxlen=max_points) # Άξονας x, χρόνος
            self.y = deque(maxlen=max_points) # Άξονας y, Bytes ανα sec
            self.start_ts = time.time() # Χρονοσφραγίδα εκκίνησης
            self.ax.set_xlabel("seconds") # Ετικέτα x
            self.ax.set_ylabel("bytes/s") # Ετικέτα y
            # Άδεια γραμμή plot (για αυτόματη ενημέρωση)
            (self.line,) = self.ax.plot([], [])
            fig.tight_layout(pad=2.5) # Αυτόματο padding (γέμισμα)
            # Περιθώρεια
            fig.subplots_adjust(top=1.0, bottom=0.2, left=0.15,
      right=0.95)


# Ενημέρωση Top Talkers (IP, traffic)
def update_talkers(self, talkers_dict):
      """talkers_dict = {ip: bytes, ...}""" # Λεξικό (IP: total bytes)
      # Φίλτρο για πολύ μικρές τιμές (<1 byte), μηδενισμός
      filtered = {ip: (val if val >= 1 else 0) for ip, val in
            talkers_dict.items()}
      # Κατά φθίνουσα σειρά ως προς το traffic
      sorted_items = sorted(filtered.items(), key=lambda x: x[1],
            reverse=True)[:5]
      # Για κάθε QLabel στη στήλη IPs
      for i, lbl in enumerate(self.entries):
            if i < len(sorted_items): # Αν υπάρχει talker
                  ip, val = sorted_items[i] # Καταχώριση IP, bytes
                  # Μορφοποίηση IP (15 χαρ/ρες), Bytes (6 χαρ/ρες 2
            δεκαδικά)
                  lbl.setText(f"{ip:<15} — {val/1e6:>6.2f} MB")
            else: # Αν δεν υπάρχουν
                  lbl.setText("—") # Παύλα
```

Η ενότητα Logs Management είναι ο βασικός μηχανισμός διαχείρισης και αποθήκευσης των δεδομένων. Τα logs συλλέγονται αρχικά σε μικρά τμήματα σε έναν buffer (self.logs_buffer) και στη συνέχεια συγχωνεύονται μέσω της μεθόδου sync_data_buffer σε ένα ενιαίο DataFrame (all_logs). Μετά τη συγχώνευση, ο buffer καθαρίζεται ώστε να αποφεύγεται η υπερχείλιση της μνήμης. Επιπλέον, εφαρμόζεται ο μηχανισμός log rotation σύμφωνα με τον οποίο όταν το αρχείο καταγραφής φτάσει ένα προκαθορισμένο μέγεθος (100MB), αρχειοθετείται και δημιουργείται νέο αρχείο, ώστε να διασφαλίζεται η σταθερότητα και η αποδοτικότητα του συστήματος σε μακροχρόνια λειτουργία.

```
self.logs_buffer.append(df) # Προσθήκη νέων logs στον buffer

# Απόσπασμα συνάρτησης sync_data_buffer()
# Συγχώνευση δεδομένων του buffer στο κύριο DataFrame
def sync_data_buffer(self):
      if self.logs_buffer: # Αν υπάρχουν δεδομένα στο buffer
            # Ένωση όλων των μικρών κομματιών σε ένα DataFrame
            new_chunk = pd.concat(self.logs_buffer, ignore_index=True)
            # Προσθήκη στο συνολικό ιστορικό
            self.all_logs = pd.concat([self.all_logs, new_chunk],
                  ignore_index=True)
            # Άδειασμα buffer μετά τη συγχώνευση
            self.logs_buffer = []
            # Ενημέρωση στατιστικών βάση όλων των δεδομένων
            self.update_attack_stats()

# Απόσπασμα συνάρτησης append_logs() – Διαχείριση Logs
def append_logs(self, df):
      if os.path.exists(self.logs_file): # Αν υπάρχει το log αρχείο
            # Όριο μεγέθους 100 MB (100 * 1024 * 1024 bytes)
            if os.path.getsize(self.logs_file) >= 100 * 1024 * 1024:
                  # Χρονοσφραγίδα για ονομασία
                  timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
                  # Φάκελος του αρχικού αρχείου
                  base_folder = os.path.dirname(self.logs_file)
                  # Υποφάκελος αρχειοθέτησης
                  archive_folder = os.path.join(base_folder, "logs")

                  # Αν δεν υπάρχει ο υποφάκελος
```

```
                if not os.path.exists(archive_folder):
                        os.makedirs(archive_folder) # Δημιουργία


                new_filename = f"logs_{timestamp}.csv" # Όνομα νέου
        αρχείου
                # Πλήρες μονοπάτι
                new_path = os.path.join(archive_folder, new_filename)
                # Μετακίνηση αρχείου στον υποφάκελο
                os.rename(self.logs_file, new_path)

# Αν έγινε rotation, το αρχείο δεν υπάρχει, άρα φτιάχνεται νέο με
headers
df.to_csv( # Αποθήκευση του Dataframe σε csv log file
      self.logs_file, # Όνομα αρχείου logs
      mode="a", # Λειτουργία προσάρτησης (προσθήκη, όχι overwrite)
      # Headers μόνο αν είναι νέο αρχείο
      header=not os.path.exists(self.logs_file),
      index=False, # Παραγωγή καθαρού csv, χωρίς την αριθμητική στήλη
      )
```

Η ενότητα System Usage παρέχει άμεση εικόνα της κατανάλωσης των πόρων του συστήματος. Υλοποιείται με τη βοήθεια τριών QProgressBar που αντιστοιχούν στη CPU, τη RAM και τον δίσκο. Οι τιμές ενημερώνονται περιοδικά (με χρήση του sys_timer, ο οποίος ενεργοποιεί τη σχετική διαδικασία ανά ένα δευτερόλεπτο) μέσω της συνάρτησης update_system_led() αλλά και των συναρτήσεων της βιβλιοθήκης psutil, όπως cpu_percent(), virtual_memory().percent και disk_usage().percent. Ειδικότερα για τη χρήση του δίσκου, υπολογίζεται κυρίως η δραστηριότητα I/O, η οποία αποτυπώνεται στη μεταβλητή disk_busy. Με αυτόν τον τρόπο, ο χρήστης μπορεί να αξιολογεί σε πραγματικό χρόνο αν η λειτουργία του IDS και η επεξεργασία των δεδομένων επιβαρύνουν το σύστημα.

```
# Απόσπασμα συνάρτησης update_system_leds()
# Ενημέρωση ενδείξεων (labels) cpu, ram, disk
def update_system_leds(self):
      # Υπολογισμός τρέχουσας χρήσης cpu & ram (ποσοστό)
      cpu = psutil.cpu_percent(interval=None)
      ram = psutil.virtual_memory().percent
      # Τρέχοντα στατιστικά δίσκου
      current_io = psutil.disk_io_counters()
```

```
current_time = time.time() # τρέχουσα χρονοσφραγίδα
# Αν δεν αποθηκεύτηκαν οι προηγούμενες τιμές
if not hasattr(self, "_last_disk_io"):
      self._last_disk_io = current_io # τρέχουσες μετρήσεις
      self._last_time = current_time # τρέχουσα χρονική στιγμή
      disk_busy = 0.0 # Αρχικοποίηση
else: # Αν αποθηκεύτηκαν οι προηγούμενες τιμές
      # Χρονικό διάστημα απο τη τελευταία μέτρηση
      delta_time = current_time - self._last_time
      if delta_time <= 0: # Αν μικρότερο / ίσο του 0
            delta_time = 0.1 # Ελάχιστος (θετικός) χρόνος
      read_rate = ( # Υπολογισμός ρυθμού ανάγνωσης:
            # Bytes ανάγνωσης προηγούμενης μέτρησης /
            # χρόνος τελευταίας μέτρησης (για bytes/sec) /
            # 1024 * 1024 (για μετατροπή σε MB/sec)
            (current_io.read_bytes -
      self._last_disk_io.read_bytes)
            / delta_time
            / (1024 * 1024)
                  )
      write_rate = ( # Υπολογισμός ρυθμού εγγραφής:
            # Bytes εγγραφής προηγούμενης μέτρησης /
            # χρόνος τελευταίας μέτρησης (για bytes/sec) /
            # 1024 * 1024 (για μετατροπή σε MB/sec)
            (current_io.write_bytes -
      self._last_disk_io.write_bytes)
            / delta_time
            / (1024 * 1024)
      )
       # Συνολικός ρυθμός ανάγνωσης και εγγραφής
      io_rate = read_rate + write_rate
      # Περιορισμός του δείκτη στο 100 (μέγιστο)
      disk_busy = min(100, io_rate)

# Ενημέρωση μετρητών με τρέχουσες τιμές
self._last_disk_io = current_io
self._last_time = current_time
# Περιορισμός τιμής σε έγκυρο εύρος
disk_busy = max(0.0, min(disk_busy, 100.0))
# Ενημέρωση UI (ακέραια ποσοστά)
```

```
        self.cpu_label.setText(f"{cpu:.0f}%") # Εμφάνιση χρήσης cpu
        self.ram_label.setText(f"{ram:.0f}%") # Εμφάνιση χρήσης ram
        self.disk_label.setText(f"{disk_busy:.0f}%") # Εμφάνιση χρήσης
δίσκου
        self.cpu_bar.setValue(int(cpu)) # Ενημέρωση progress bar CPU
        self.ram_bar.setValue(int(ram)) # Ενημέρωση progress bar RAM
        self.disk_bar.setValue(int(disk_busy)) # Ενημέρωση progress bar
δίσκου



# Δημιουργία Timer για περιοδική ενημέρωση
self.net_timer = QTimer()
# Σύνδεση με συνάρτηση ενημέρωσης
self.net_timer.timeout.connect(self.update_network_activity)
# Εκτέλεση κάθε 1 δευτερόλεπτο
self.net_timer.start(1000)
```

Η ενότητα Emergency Control (Emergency Disconnect) λειτουργεί ως μηχανισμός άμεσης παρέμβασης σε κρίσιμες καταστάσεις κι επαναφορά της σύνδεσης έπειτα από μια κατάσταση απομόνωσης. Υλοποιείται μέσω ενός κουμπιού (kill_btn) που συνδέεται με τη μέθοδο emergency_shutdown_network. Όταν ενεργοποιηθεί, εκτελούνται εντολές σε επίπεδο εφαρμογής και λειτουργικού συστήματος, οι οποίες αφορούν την διακοπή διαδικασίας σύλληψης πακέτων (για ομαλότερη λειτουργία του συστήματος) και την απενεργοποίηση των δικτυακών προσαρμογέων (network adapters), προκειμένου να απομονωθεί πλήρως το σύστημα από το δίκτυο. Η λειτουργία αυτή έχει σχεδιαστεί ως «διακόπτης έκτακτης ανάγκης» για περιπτώσεις σοβαρής απειλής και για την εκτέλεσή της απαιτούνται δικαιώματα διαχειριστή.

```
# Επείγουσα διακοπή δικτύου (Απενεργοποίηση δικτυακών διεπαφών)
# Αποσπάσματα της συνάρτησης emergency_shutdown_network()
def emergency_shutdown_network(self):
      if not self.network_disabled: # Αν δεν έχει ήδη απενεργοποιηθεί
            reply = QMessageBox.question( # Διάλογος επιβεβαίωσης
                  self,
                  "Emergency Network Kill", # Τίτλος - Προειδοποίηση
                  "WARNING: All network adapters (Wi-Fi/Ethernet)
                        will be disabled.\n"
```

```
                "Administrator privileges will be
        requested.\n\nContinue?",
                QMessageBox.Yes | QMessageBox.No, # Κουμπιά επιλογών
                QMessageBox.No, # Default το κουμπί No
        )
        if reply != QMessageBox.Yes: # Αν όχι Yes
                return # Επιστροφή (Ακύρωση διαδικασίας)

        # PowerShell, εύρεση και απενεργοποίηση των δικτυακών
    διεπαφών
        ps_command = "Get-NetAdapter | Disable-NetAdapter -
                Confirm:$false"

        # Πριν την απενεργοποίηση, διακοπή διαδικασίας σύλληψης
    πακέτων
        self.stop_capture()

        # Εκτέλεση powershell με δικαιώματα διαχειριστή
        result = ctypes.windll.shell32.ShellExecuteW(
                # Εκτέλεση με runas (UAC prompt)
                None, "runas", "powershell.exe", f'-Command
                        "{ps_command}"', None, 1
        )
```

Η ενότητα Detected Attacks (1st & 2nd Models) συνδυάζει τα αποτελέσματα, από τα δύο διαφορετικά μοντέλα ανίχνευσης, για πιο ολοκληρωμένη κάλυψη. Το πρώτο μοντέλο (CIC-IDS2017) εξειδικεύεται σε κατηγορίες όπως DoS, PortScan και Web Attacks, ενώ το δεύτερο (UNSW-NB15) εντοπίζει επιπλέον τύπους όπως Fuzzers, Backdoors και Reconnaissance. Τα αποτελέσματα συγχωνεύονται σε μια κοινή μεταβλητή (attack_merged), έτσι ώστε να είναι εφικτή η ενοποιημένη απεικόνιση των απειλών. Στη συνέχεια, υπολογίζεται η συχνότητα εμφάνισης των επιμέρους επιθέσεων και ενημερώνονται δυναμικά τα labels των στατιστικών καρτών, παρέχοντας με αυτό το τρόπο τόσο συνοπτική όσο και αναλυτική εικόνα των επιθέσεων.

```
# Αποσπάσματα της συνάρτησης update_executive_data()
# Νέο DataFrame - Φιλτράρισμα μόνο των κακόβουλων εγγραφών
malicious = filtered_df[
        # Κρατάει τις γραμμές στις οποίες περιέχεται το CIC ή το UNSW
```

```
        filtered_df["attack_merged"].str.contains("CIC|UNSW", na=False)
]

# Υπολογισμός κρισιμότητας βάσει της επικινδυνότητας των κατηγοριών CIC/UNSW
critical_heat = (
        malicious["attack_merged"]  # Στήλη με τα ονόματα των επιθέσεων
        .str.contains(  # Αναζήτηση, αν περιέχονται τα παρακάτω (Critical Heat)
                "DoS|Exploits|Bot|Infiltration|Backdoors|Worms|Shellcode|Web
                        Attack",
                na=False, # NaN τιμές αγνοούνται
        )
        .sum()  # Πλήθος κρίσιμων επιθέσεων
)
# PortScans και Fuzzers προκαλούν "Medium Heat"
recon_heat = (
        malicious["attack_merged"]  # Στήλη με τα ονόματα των επιθέσεων
                # Αναζήτηση, αν περιέχονται τα παρακάτω (Medium Heat)
                .str.contains(
                "PortScan|Fuzzers|Reconnaissance|Analysis|Generic", na=False
        )
        .sum() # Πλήθος medium επιθέσ εων
)

# Το πιο συχνό είδος επίθεσης των συγχωνευμένων αποτελεσμάτων (cic, unsw)
top_type = malicious["attack_merged"].mode().iloc[0]

# Αποσπάσματα της συνάρτησης update_alerts()
# Ενοποιημένη απεικόνιση των απειλών
attack_name = ( # Εφεδρική ετικέτα
        row.get("attack_merged") # Ετικέτα επίθεσης (συγχωνευμένη)
        or row.get("attack_type") # Είδος επίθεσης
        or row.get("prediction") # Πρόβλεψη
        or "Unclassified" # Μη ταξινομημένη (χωρίς ή άγνωστο περιεχόμενο)
)

# Υπολογισμός και ενημέρωση των στατιστικών επιθέσεων στο UI
def update_attack_stats(self):
```

```
# Αν δεν υπάρχουν logs ή το DataFrame είναι άδειο
if self.all_logs is None or self.all_logs.empty:
      # Μηδενισμός ετικετών αν δεν υπάρχουν logs
      self.attack_exploits_lbl.setText("Exploits: 0")
      self.attack_dos_lbl.setText("DoS: 0")
      self.attack_portscan_lbl.setText("PortScan: 0")
      self.attack_web_lbl.setText("Web: 0")
      self.attack_generic2_lbl.setText("Generic: 0")
      self.attack_fuzzers_lbl.setText("Fuzzers: 0")
      self.attack_backdoors_lbl.setText("Backdoors: 0")
      self.attack_recon_lbl.setText("Reconnaissance: 0")
      return # Τερματισμός (δεν υπάρχουν δεδομένα για υπολογισμό)

try: # Υπολογισμός κι ενημέρωση
      counts = (
            # Μετατροπή σε string και πεζά, καταμέτρηση τιμών
            self.all_logs["attack_merged"].astype(str).str.lower()
                  .value_counts()
      )

      def get_sum(keyword): # Άθροιση counts βάση keyword
            return sum(val for key, val in counts.items() if
                  keyword in key)
      # Σύνολο επιθέσεων που περιέχουν τη λέξη exploit
      exploits = get_sum("exploit")
      dos = get_sum("dos") # Σύνολο επιθέσεων dos
      portscan = get_sum("portscan") # Σύνολο επιθέσεων portscan
      web = get_sum("web") # Σύνολο επιθέσεων web
      fuzzers = get_sum("fuzzer") # Σύνολο επιθέσεων fuzzer
      backdoors = get_sum("backdoor") # Σύνολο επιθέσεων backdoor
      recon = get_sum("recon") # Σύνολο επιθέσεων recon
      generic = get_sum("generic") # Σύνολο επιθέσεων generic

      # Ενημέρωση ετικετών UI
      # label Exploits
      self.attack_exploits_lbl.setText(f"Exploits: {exploits}")
      self.attack_dos_lbl.setText(f"DoS: {dos}") # label dos
      # label PortScan
      self.attack_portscan_lbl.setText(f"PortScan: {portscan}")
```

```
            self.attack_web_lbl.setText(f"Web: {web}") # label Web
            # label Generic
            self.attack_generic2_lbl.setText(f"Generic: {generic}")
            # label Fuzzers
            self.attack_fuzzers_lbl.setText(f"Fuzzers: {fuzzers}")
            # label Backdoors
            self.attack_backdoors_lbl.setText(f"Backdoors: {backdoors}")
            # label Reconnaissance
            self.attack_recon_lbl.setText(f"Reconnaissance: {recon}")

      except Exception as e: # Σφάλμα κατά τον υπολογισμό - ενημέρωση
            print(f"Stats error: {e}") # Εμφάνιση σφάλματος στη κονσόλα
            # Καταγραφή στο Log file
            logging.error(f"Stats error: {e}", exc_info=True)
```

Η ενότητα Detection Latency (TTFD) μετράει την απόδοση του συστήματος ως προς τον χρόνο ανίχνευσης. Για την υλοποίησή της υπολογίζεται η διαφορά μεταξύ της χρονικής στιγμής εισόδου μιας ροής στον aggregator (first_ts) και της στιγμής παραγωγής της πρόβλεψης ως κακόβουλη (First_detect_ts, detect_ts). Το αποτέλεσμα, αποθηκεύεται ως detection_delay και μέσω του πεδίου ttfd_value_label, εμφανίζεται σε δευτερόλεπτα με υψηλή ακρίβεια (χιλιοστά του δευτερολέπτου), παρέχοντας έναν άμεσο δείκτη της ταχύτητας απόκρισης του IDS. Η μέτρηση αυτή είναι σημαντική για την αξιολόγηση της αποτελεσματικότητας του συστήματος σε περιβάλλοντα όπου ο χρόνος αντίδρασης είναι καθοριστικός.

```
# Αν υπάρχει ροή f και δεν έχει οριστεί ο χρόνος εντοπισμού
if f and f["first_detect_ts"] is None:
      # Χρόνος εντοπισμού
      f["first_detect_ts"] = row["detect_ts"]

# Χρόνος καθυστέρησης ανίχνευσης
"detection_delay": (
      # πρώτος εντοπισμός - αρχικός εντοπισμός
      f["first_detect_ts"] - f["first_ts"]
      # Αν έχει καταγραφεί (do nothing)
      if f.get("first_detect_ts") is not None
      else 0.0 # Αλλιώς 0 δευτερόλεπτα
```

```
)

# Ανάγνωση καθυστέρησης ανίχνευσης απο τρέχων alert (προεπιλογή 0.0)
raw_delay = self.current_explained_alert.get("detection_delay", 0.0)
if raw_delay < 1e-3: # Αν η καθυστέρηση είναι μικρότερη από 0.001 sec
      # Εμφάνιση τιμής για πολύ μικρή καθυστέρηση
      self.ttfd_value_label.setText("< 0.001")
else:
      # Εμφάνιση καθυστέρησης με 3 δεκαδικά
      self.ttfd_value_label.setText(f"{raw_delay:.3f}")
```

8.4.4. Live Detection και παρακολούθηση σε πραγματικό χρόνο

Η καρτέλα Live Detection λειτουργεί ως το βασικό σημείο παρακολούθησης της δικτυακής κίνησης σε πραγματικό χρόνο, συνδυάζοντας τη συλλογή πακέτων, την επεξεργασία ροών και την ανάλυση δεδομένων για την άμεση αναγνώριση πιθανών απειλών. Η λειτουργία της καρτέλας βασίζεται σε μια συνεχόμενη ροή επεξεργασίας, σύμφωνα με την οποία τα δεδομένα μετατρέπονται σταδιακά από ακατέργαστη πληροφορία σε αξιοποιήσιμα συμπεράσματα.

Η διαδικασία ξεκινά με τη συλλογή των δικτυακών πακέτων μέσω της βιβλιοθήκης pcapy, σε συνδυασμό με την αξιοποίηση του μηχανισμού πολυνηματικής επεξεργασίας, έτσι ώστε η καταγραφή της κίνησης να γίνεται χωρίς να επηρεάζεται η απόκριση του γραφικού περιβάλλοντος. Μέσω της συνάρτησης add_raw_packet(), τα πακέτα που συλλέγονται αναλύονται σε επίπεδο Ethernet, IP και μεταφοράς πρωτοκόλλων (TCP, UDP, ICMP), προκειμένου να εξαχθούν βασικές πληροφορίες όπως οι διευθύνσεις IP, οι θύρες, οι σημαίες / flags (για TCP-UDP) και το μέγεθος των δεδομένων.

```
# Αποσπάσματα της συνάρτησης add_raw_packet()
def add_raw_packet(self, raw_bytes, ts):
      """Επεξεργασία πακέτου με dpkt"""
      # Ethernet αντικείμενο με raw Bytes
      eth = dpkt.ethernet.Ethernet(raw_bytes)
      # Αν δεν είναι IP πακέτο (π.χ. ARP)
      if not isinstance(eth.data, dpkt.ip.IP):
            return  # break

      ip = eth.data  # Επίπεδο IP (απο ethernet frame)
      size = len(raw_bytes)  # Συνολικό πακέτο (ethernet + payload)
```

```
# Μετατροπή IP bytes σε string (π.χ.
# b'\xc0\xa8\x01\x01' σε '192.168.1.1')
src_ip = socket.inet_ntoa(ip.src) # IP πηγής
dst_ip = socket.inet_ntoa(ip.dst) # IP προορισμού

# Αν τα δεδομένα είναι TCP πακέτο
if isinstance(ip.data, dpkt.tcp.TCP):
      tcp = ip.data  # Αναφορά TCP αντικειμένου
      proto_name = "TCP"  # Όνομα πρωτοκόλλου
      sport = tcp.sport  # Θύρα πηγής
      dport = tcp.dport  # Θύρα προορισμού
      payload_len = len(tcp.data)  # Μήκος ωφέλιμου φορτίου

      # Μετατροπή bits σε γράμματα για να ταιριάζει με τον παλιό
κώδικα
      flags_int = tcp.flags
      if flags_int & dpkt.tcp.TH_SYN:
            tcp_flags_str += "S" # TCP SYN flag
      if flags_int & dpkt.tcp.TH_ACK:
            tcp_flags_str += "A" # TCP ACK flag
# Αν τα δεδομένα είναι UDP πακέτο
elif isinstance(ip.data, dpkt.udp.UDP):
      proto_name = "UDP"  # Όνομα πρωτοκόλλου
elif ip.p == 89:  # 89 = OSPF
      proto_name = "ospf"  # Όνομα πρωτοκόλλου
elif ip.p == 1:  # 1 = ICMP
      proto_name = "icmp"  # Όνομα πρωτοκόλλου
```

Στη συνέχεια, τα δεδομένα προωθούνται στον συσσωρευτή ροών Flow Aggregator, ο οποίος παρέχει την κεντρική ανάλυση. Εκεί, τα μεμονωμένα πακέτα ομαδοποιούνται σε αμφίδρομες ροές (flows) με βάση τον συνδυασμό διεύθυνσης IP πηγής και προορισμού, θυρών και πρωτοκόλλου, εφαρμόζοντας κανονικοποίηση των άκρων της επικοινωνίας προκειμένου κάθε σύνδεση να αναπαρίσταται ενιαία ανεξάρτητα από την κατεύθυνση. Παράλληλα, για κάθε ροή υπολογίζεται ένα εκτενές σύνολο στατιστικών χαρακτηριστικών σε πραγματικό χρόνο, όπως η διάρκεια της ροής, ο αριθμός και το μέγεθος των πακέτων, οι χρονικές αποστάσεις μεταξύ τους (inter-arrival times), καθώς επίσης και οι μετρητές για TCP flags (όπως SYN, ACK, FIN κλ.π). Η διαδικασία αυτή επιτρέπει τη μετατροπή της

χαμηλού επιπέδου δικτυακής πληροφορίας σε δομημένα δεδομένα κατάλληλα για περαιτέρω ανάλυση.

```
# Δημιουργία Κλειδιού (Canonical Key) - Bidirectional flow normalization
# (Αποσπάσματα της συνάρτησης add_raw_packet())
key = None  # Αρχικοποίηση σε none
a = (src_ip, sport)  # source flow ID (IP & Θύρα πηγής)
b = (dst_ip, dport)  # destination flow ID (IP & Θύρα προορισμού)
if a <= b:  # source flow ID <= destination flow ID
      # source-dest (canonical order)
      key = (src_ip, sport, dst_ip, dport, proto_name)
else:  # αντιστροφή για bidirectional μοναδικότητα
      key = (dst_ip, dport, src_ip, sport, proto_name)

      ... # Γραμμές κώδικα

# Δημιουργία νέας ροής
if f is None:  # Αν δεν υπάρχει στο λεξικό
      self.cnt_flows_created += 1  # Αύξηση μετρητή ροών
      f = {  # Δημιουργία λεξικού ροής
            "first_ts": ts,  # Χρονοσφραγίδα πρώτου πακέτου
            "last_ts": ts,  #  Χρονοσφραγίδα τελευταίου πακέτου
            "first_detect_ts": None,  # Χρονοσφραγίδα πρώτης ανίχνευσης
            "pkt_times": deque([ts]),  # Ουρά χρόνων των πακέτων
            "pkt_lens": deque([size]),  # Ουρά μεγεθών των πακέτων
            "ackdat_bytes": 0,  # TCP ACK + data bytes
            "sloss": 0,  # Μετρητής απώλειας πακέτων πηγής
            "dloss": 0,  # Μετρητής απώλειας πακέτων προορισμού
            "total_pkts": 1, # Σύνολο πακέτων
            "total_bytes": size, # Σύνολο bytes

# Στατιστικά χαρακτηριστικά
f["total_pkts"] += 1  # Μετρητής πακέτων
f["total_bytes"] += size  # Συνολικά Bytes
f["last_ts"] = ts  # Χρονοσφραγίδα τελευταίας δραστηριότητας

# Καταμέτρηση Flags (από το string που δημιουργήθηκε)
if "S" in tcp_flags_str: # Αν υπάρχει SYN flag στο TCP packet
      f["syn"] += 1 # Μετρητής SYN
```

```
if "A" in tcp_flags_str: # Αν υπάρχει ACK flag στο TCP packet
      f["ack"] += 1 # Μετρητής ACK
      f["ackdat_bytes"] += size # Μετρητής ACK Bytes
if "F" in tcp_flags_str: # Αν υπάρχει FIN flag στο TCP packet
      f["fin"] += 1 # Μετρητής FIN

# Αν το πακέτο ανήκει στην αρχική κατεύθυνση της σύνδεσης
if forward_dir:
      f["fwd_pkts"] += 1 # Αύξηση του μετρητή πακέτων αποστολής
      # Προσθήκη μεγέθους του τρέχοντος πακέτου στο σύνολο των Forward
      bytes
      f["fwd_bytes"] += size
# Αν το πακέτο ανήκει στην κατεύθυνση απάντησης
else:
      f["bwd_pkts"] += 1 # Αύξηση του μετρητή πακέτων επιστροφής
      # Προσθήκη μεγέθους του τρέχοντος πακέτου στο σύνολο των Backward
      bytes
      f["bwd_bytes"] += size
```

Σε τακτά χρονικά διαστήματα, ο συσσωρευτής δημιουργεί στιγμιότυπα (snapshots) των ενεργών ροών, τα οποία μετατρέπονται σε δομές τύπου DataFrame και χρησιμοποιούνται ως είσοδος για περαιτέρω επεξεργασία (τα στιγμιότυπα χειρίζεται η συνάρτηση snapshot_and_clear() όπως περιγράφεται και αναλύεται στην ενότητα 9.2.2). Τα δεδομένα αυτά μπορούν να αξιοποιηθούν από τα μοντέλα μηχανικής μάθησης για την κατηγοριοποίηση της κίνησης σε φυσιολογική ή κακόβουλη, εμπλουτίζοντας κάθε ροή με χαρακτηρισμούς που αποτυπώνουν τη φύση της δραστηριότητας.

Η απεικόνιση των ροών στην καρτέλα πραγματοποιείται δυναμικά μέσω του γραφικού περιβάλλοντος, στο οποίο εμφανίζεται περιορισμένο υποσύνολο έως 100 εγγραφών (ώστε να αποφεύγεται η υπερφόρτωση της εφαρμογής). Τα αποτελέσματα ενσωματώνονται σε πίνακες δεδομένων που ενημερώνονται συνεχώς, ενώ κάθε ροή εμφανίζεται ως γραμμή με τα βασικά χαρακτηριστικά της, όπως διευθύνσεις IP, θύρες, χρονικές ενδείξεις και τύπος δραστηριότητας, αξιοποιώντας το μοντέλο δεδομένων που βασίζεται σε DataFrame (PandasModel). Παράλληλα, εφαρμόζεται χρωματική κωδικοποίηση στις εγγραφές, ώστε να διακρίνονται άμεσα οι κατηγορίες κίνησης (κανονική, κακόβουλη ή μη ταξινομημένη), μέσω της χρήσης πεδίων όπως το color_tag που επηρεάζουν την εμφάνιση κάθε γραμμής

(PandasModel.data). Η δυναμική αυτή ενημέρωση επιτρέπει στον χρήστη να παρακολουθεί σε πραγματικό χρόνο την εξέλιξη της δικτυακής δραστηριότητας.

```
# Απεικόνιση Ροών
# Qt Model για εμφάνιση του DataFrame στο QTableView
class PandasModel(QtCore.QAbstractTableModel):
      # Constructor για αρχικοποίηση μοντέλου (άδειο DataFrame,
      # parent widget, flag dark_mode)
      def __init__(self, df=pd.DataFrame(), parent=None,
dark_mode=False):
            super().__init__(parent)  # Κλήση του constructor
            self._df = df  # Αποθήκευση DataFrame για εσωτερική χρήση
            self.dark_mode = (
                  # Αποθήκευση flag dark_mode (για προσαρμογή χρωμάτων
            UI)
                  dark_mode
      )

# Απόσπασμα συνάρτησης data(). Εμφάνιση κάθε ροής ως γραμμή
# Επιστροφή των δεδομένων για συγκεκριμένη θέση και ρόλο
def data(self, index, role=Qt.DisplayRole):
      # Αν ο ρόλος είναι DisplayRole (προεπιλογή για
      # εμφάνιση του κειμένου στα κελιά)
      if role == Qt.DisplayRole:
            # Επιστρέφει την τιμή του DataFrame στη θέση
            # [row, column] ως string
            return str(self._df.iloc[index.row(), index.column()])

      # Αν υπάρχει η στήλη 'color_tag' στο DataFrame (Χρωματική
κωδικοποίηση)
      if "color_tag" in self._df.columns:
            color = str( # Μετατροπή της τιμής color_tag σε string
                  # Πρόσβαση στη τιμή του κελιού στη στήλη color_tag
                  self._df.iloc[
                        # Γραμμή, θέση στήλης
                        index.row(),
                  self._df.columns.get_loc("color_tag")
                  ]
            ).lower() # Μετατροπή σε πεζά

            if color == "red": # Αν color_tag = "red"
```

```
            # Σκούρο κόκκινο
            return QtGui.QBrush(QtGui.QColor(140, 30, 30, 200))
        elif color == "green": # Αν color_tag = "green"
            # Σκούρο Πράσινο
            return QtGui.QBrush(QtGui.QColor("#277247"))
        elif color == "gray": # Αν color_tag = "gray"
            return QtGui.QBrush(QtGui.QColor("#2a2a2a")) # Σκούρο
        Γκρί
```

Τέλος, το σύστημα διατηρεί μια συνεχή σύνδεση μεταξύ της ανάλυσης και της οπτικοποίησης, ενημερώνοντας τα δεδομένα της διεπαφής καθώς νέες ροές δημιουργούνται ή παλαιότερες λήγουν. Με τον τρόπο αυτό, η καρτέλα Live Detection προσφέρει μια ολοκληρωμένη και άμεση εικόνα της κατάστασης του δικτύου, επιτρέποντας την ταχεία αναγνώριση και αξιολόγηση πιθανών κυβερνοαπειλών σε πραγματικό χρόνο.

8.4.5. Explainability περιβάλλον

Η καρτέλα Explainability (XAI) Analysis υλοποιείται ως ένας συνδυασμός PyQt5 για το UI, που αξιοποιεί το SHAP για την εξήγηση των μοντέλων και το Matplotlib για την οπτικοποίηση των αποτελεσμάτων. Η σελίδα δημιουργείται μέσω της συνάρτησης create_xai_page(), μέσα στην οποία ορίζονται βασικά στοιχεία της διεπαφής όπως πολλαπλά QLabel για την εμφάνιση κειμένου (summary, features, model decisions), ένα ειδικό πλαίσιο για beginner explanation και ένα FigureCanvas της Matplotlib που φιλοξενεί το γράφημα. Κάθε ένα από αυτά τα στοιχεία μορφοποιείται με Qt Style (background, padding, fonts) προκειμένου να προκύψει η «card-based» εμφάνιση.

```
# Δημιουργία καρτέλας Explainability Analysis
# Απόσπασμα συνάρτησης create_xai_page()
def create_xai_page(self):
    page = QWidget() # Βασικό widget container για τη σελίδα XAI
    layout = QVBoxLayout(page) # vertical layout για XAI components

    # Ετικέτα σύνοψη ανάλυσης alert (αρχικοποίηση)
    self.xai_summary = QLabel("No alert selected.")

    # Ετικέτα σημαντικότερων χαρακτηριστικών που συνέβαλαν στην
πρόβλεψη
    self.xai_features = QLabel("")
```

```
# Ετικέτα σύγκρισης αποφάσεων μεταξύ των CIC/UNSW
self.xai_models = QLabel("")
self.xai_models.setStyleSheet( # Εφαρμογή custom CSS styling
      """
      QLabel {
            background-color: #ffffff;
            border-radius: 10px;
            padding: 14px;
            font-size: 10pt;
            color: #333;
            border: 1px solid #e0e0e0;
      }
"""
)
layout.addWidget(self.xai_models)
# Matplotlib γράφημα για SHAP bar chart (πλάτος, ύψος σε ίντσες)
self.xai_fig = Figure(figsize=(5, 2))
# Ενσωμάτωση γραφήματος στο PyQt5 UI
self.xai_canvas = FigureCanvas(self.xai_fig)
# Άξονες του SHAP plot (1 γραμμή, 1 στήλη, subplot 1)
self.xai_ax = self.xai_fig.add_subplot(111)
# Ετικέτα οδηγιών
self.xai_beginner_txt = QLabel("Select an alert to see
      explanation.")
```

Η ενημέρωση της καρτέλας γίνεται δυναμικά όταν ο χρήστης επιλέξει ένα alert από τον πίνακα. Με την επιλογή (on click) ενεργοποιείται η συνάρτηση on_table_row_clicked, η οποία αποθηκεύει τη γραμμή δεδομένων και καλεί τη βασική συνάρτηση render_full_xai() για να δημιουργήσει όλο το περιεχόμενο της σελίδας. Στη συνέχεια, όλο το υπόλοιπο κομμάτι της καρτέλας (κείμενα, τιμές, γράφημα) παράγεται runtime με βάση τα δεδομένα της συγκεκριμένης εγγραφής. Σε αυτό το σημείο, δημιουργείται ουσιαστικά η σύνοψη του περιστατικού, γίνεται η ανάκτηση του είδους της επίθεσης, των IP διευθύνσεων (πηγής και προορισμού) και της χρονικής σήμανσης και σχηματίζεται η συνοπτική εικόνα στην κορυφή της καρτέλας.

```
# Χειρισμός του κλικ του χρήστη σε μια γραμμή του πίνακα
def on_table_row_clicked(self, index):
```

```
try: # Μπλοκ διαχείρισης σφαλμάτων
      sender_view = self.sender() # Αποστολέας σήματος
      if not sender_view: # Αν δεν βρεθεί
            return # Τερματισμός
      # Μοντέλο δεδομένων που χρησιμοποιεί το table view
      model = sender_view.model()
      row_idx = index.row() # Αριθμός γραμμής που πατήθηκε
      # Αν το μοντέλ περιέχει DataFrame (στο attribute _df)
      if hasattr(model, "_df"):
            # Ανάκτηση ολόκληρης της γραμμής
            row_data = model._df.iloc[row_idx]

            # Αποθήκευση της επιλεγμένης γραμμής
            self.current_explained_alert = row_data
            # Επιβεβαίωση χειροκίνητου alert
            self.user_selected_alert = True

            # Ενημέρωση XAI preview
            self.update_xai_preview_from_row(row_data)
            # Μετάβαση στη σελίδα XAI
            self.content_area.setCurrentWidget(self.xai_page)
            # Σχεδίαση XAI ανάλυσης
            self.render_full_xai(self.current_explained_alert)

except Exception as e: # Αν προκύψει σφάλμα
      # Εμφάνιση σφάλματος στη κονσόλα
      print(f"Error clicking row: {e}")
      # Καταγραφή του σφάλματος στο log
      logging.error(f"Error clicking row: {e}", exc_info=True)
```

Στη συνέχεια προσδιορίζεται η οικογένεια της επίθεσης μέσω της συνάρτησης infer_attack_family() και ανακτάται το αντίστοιχο προφίλ ερμηνείας μέσω της xai_profile_for_family(). Στον πυρήνα της υλοποίησης βρίσκεται η εξαγωγή SHAP values. Μέσα στην render_full_xai(), το σύστημα μετατρέπει πρώτα τη γραμμή σε DataFrame και στη συνέχεια περνά τα δεδομένα από τους αντίστοιχους mappers για τα δύο μοντέλα (UNSW και CIC). Για κάθε μοντέλο, ελέγχει αν υπάρχει διαθέσιμη επεξήγηση (SHAP

explainer) και αν υπάρχει, υπολογίζει τα πιο σημαντικά χαρακτηριστικά που υποστηρίζουν τη τελική πρόβλεψη.

```
# Μετατροπή σειράς σε μια σειρά ανά DataFrame
# Αποσπάσματα συνάρτησης (κεφ. 9.4.3) update_xai_preview_from_row()
row_df = pd.DataFrame([row.to_dict()
      if hasattr(row, "to_dict") else dict(row)])

# Προετοιμασία των χαρακτηριστικών για το μοντέλο unsw
X_unsw, _ = self.mapper_unsw.prepare(row_df)
# Επεξήγηση SHAP για το μοντέλο unsw
explainer = self.shap_explainers.get("unsw")

# Προετοιμασία χαρακτηριστικών cic
X_cic_raw, _ = self.mapper_cic.prepare(row_df)

      ... # Γραμμές κώδικα

# Αποσπάσματα συνάρτησης render_full_xai()
# Εξαγωγή top-5 SHAP τιμών για το μοντέλο UNSW-NB15
unsw_pairs = self._extract_top_shap_for_row(
      explainer=explainer_unsw, # Επεξήγηση SHAP
      model=self.model_unsw, # Πολυκλασικό (loaded) μοντέλο
      # Προεπεξεργασμένα χαρακτηριστικά (για UNSW dataset format)
      X_row=X_unsw,
      top_n=5, # Σύνολο top χαρακτηριστικών που θα επιστραφούν
      )

# Εξαγωγή top-5 SHAP τιμών για το μοντέλο CIC-IDS2017
cic_pairs = self._extract_top_shap_for_row(
      explainer=explainer_cic, # Επεξήγηση SHAP
      model=model_cic_selected,  # Μοντέλο για τη συγκεκριμένη επίθεση
      X_row=X_cic, # Προεπεξεργασμένα χαρακτηριστικά (για CIC dataset
format)
      top_n=5, # Σύνολο top χαρακτηριστικών που θα επιστραφούν
      )
```

Τα αποτελέσματα μετατρέπονται σε αναγνώσιμο κείμενο που εμφανίζεται στο UI. Για κάθε χαρακτηριστικό δημιουργείται μια συμβολοσειρά της μορφής feature = value (SHAP:

+x.xxx), η οποία στη συνέχεια προστίθεται στη λίστα επεξηγήσεων (top supporting). Παράλληλα, δημιουργείται και το block “Model decisions”, όπου εμφανίζονται οι προβλέψεις των δύο μοντέλων (CIC-IDS2017 και UNSW-NB15) και γίνεται λογικός έλεγχος ώστε να εξεταστεί αν και τα δύο μοντέλα θεωρούν τη ροή φυσιολογική, αν και τα δύο την θεωρούν κακόβουλη, αν συμφωνούν στον τύπο της επίθεσης ή αν καταλήγουν σε διαφορετική ερμηνεία της ίδιας δραστηριότητας.

```
# Αποσπάσματα συνάρτησης render_full_xai()
if unsw_pairs: # Αν το μοντέλο UNSW παρήγαγε έγκυρα SHAP pairs
      for feat, shap_val in unsw_pairs: # Για κάθε ένα απο τα
χαρακτηριστικά
            # Πραγματική τιμή των χαρακτηριστικών
            raw_val = X_unsw.iloc[0][feat] if feat in X_unsw.columns
                  else row.get(feat, "N/A")
            unsw_reasons.append( # Αναλυτική επεξήγηση (UI Text)
                  # Τιμή σε χαρακτηριστικό, επιρροή shap
                  f"{feat:<25} = {raw_val}  (SHAP: {shap_val:+.4f})"
            )
            unsw_labels.append(f"UNSW: {feat}") # Ετικέτα matplotlib
      chart
            # Shap τιμή για matplotlib chart
            unsw_values.append(float(shap_val))
      used_unsw_shap = True # Χρήση shap (flag)

      ... # Γραμμές κώδικα

if cic_pairs: # Αν το μοντέλο CIC παρήγαγε έγκυρα SHAP pairs
      for feat, shap_val in cic_pairs: # Για κάθε ένα απο τα
χαρακτηριστικά
            # Πραγματική τιμή των χαρακτηριστικών
            raw_val = X_cic.iloc[0][feat] if feat in X_cic.columns
                  else row.get(feat, "N/A")
            cic_reasons.append( # Αναλυτική επεξήγηση (UI Text)
                  # Τιμή σε χαρακτηριστικό, επιρροή shap
                  f"{feat:<25} = {raw_val}  (SHAP: {shap_val:+.4f})"
            )
            cic_labels.append(f"CIC: {feat}") # Ετικέτα matplotlib chart
            # Shap τιμή για matplotlib chart
            cic_values.append(float(shap_val))
```

```
used_cic_shap = True # Χρήση shap (flag)

... # Γραμμές κώδικα

# Σύγκριση αποτελεσμάτων μοντέλων
models_text = ( # HTML-formatted κείμενο αποφάσεων των 2 μοντέλων
      "<b>Model decisions:</b><br><br>" # Τίτλος (bold και διπλό
enter)
      f"CIC-IDS2017 → <b>{cic}</b><br>" # CIC πρόβλεψη (bold)
      f"UNSW-NB15 → <b>{unsw}</b><br><br>" # UNSW πρόβλεψη (bold)
)

cic_attack = cic.lower() != "benign" # όχι κανονική κίνηση
unsw_attack = unsw.lower() != "normal" # όχι κανονική κίνηση
# Αν και τα δύο μοντέλα εντόπισαν κακόβουλη κίνηση
if cic_attack and unsw_attack:
      # Αν συμφωνούν στο είδος επίθεσης
      if cic.lower() == unsw.lower():
            models_text += ( # Μήνυμα επιβεβαίωσης
            "<span style='color:#2E7D32;'>" # Πράσινο
            "Models agree on the attack type."
            "</span>"
      )
      else: # Αν δεν συμφωνούν στο είδος επίθεσης
            models_text += ( # Μήνυμα προειδοποίησης
            "<span style='color:#F57C00;'>" # Πορτοκαλί
            "Both models detected malicious traffic, "
            "but they disagree on the attack type."
            "</span>"
      )
# Αν και τα δύο μοντέλα εντόπισαν κανονική κίνηση
elif not cic_attack and not unsw_attack:
      models_text += ( # Μήνυμα επιβεβαίωσης
      "<span style='color:#2E7D32;'>" # Πράσινο
      "Models agree that the traffic is normal."
      "</span>"
      )
else: # Αλλιώς υπάρχει διαφωνία (normal vs malicious)
      models_text += ( # Μήνυμα προειδοποίησης
```

```
            # κόκκινο
            "<span style='color:#D32F2F;'>Models disagree.</span><br>"
            "One model detected malicious activity while the other did
      not."
      )

        # Εφαρμογή του κειμένου στο XAI models label
        self.xai_models.setText(models_text)
```

Επιπλέον, η υλοποίηση περιλαμβάνει και ένα “beginner explanation” section, το οποίο δίνει πιο κατανοητή περιγραφή της επίθεσης και μπορεί να ενσωματώνει πληροφορία από το MITRE ATT&CK, βοηθώντας έτσι επιπλέον στη γεφύρωση του χάσματος μεταξύ τεχνικής ανάλυσης και κατανόησης από μη ειδικούς. Η πληροφορία αυτή ανακτάται από προκαθορισμένη δομή αντιστοίχισης MITRE ATT&CK μέσω της συνάρτησης _mitre_context_for_family(), η οποία βασίζεται στην οικογένεια της επίθεσης και επιστρέφει τα αντίστοιχα πεδία tactic, technique, tech_id και τον σχετικό σύνδεσμο.

```
# Απόσπασμα συνάρτησης render_full_xai()
# Αν υπάρχει ετικέτα beginner explanation
if hasattr(self, "xai_beginner_txt"):
      msg = self.attack_descriptions.get( # Λήψη περιγραφής από λεξικό
            family, # Οικογένεια επίθεσης
            # Εφεδρικό (γενικό) μήνυμα
            self.attack_descriptions.get("generic", "Suspicious behavior
                  detected.")
      )

      self.xai_beginner_txt.setStyleSheet( # Κάρτα πληροφοριών
            "background-color: #E3F2FD; " # Background (Light Blue)
            "color: #0D47A1; " # Κείμενο (Blue)
            "border: 1px solid #BBDEFB; " # Περιθώριο (Light Blue)
            "padding: 14px; " # Εσωτερικό περιθώριο (Light Blue)
            "border-radius: 10px; " # Στρογγυλεμένες γωνίες
            "font-size: 11pt;" # Μέγεθος γραμματοσειράς
      )

      if mitre: # Αν υπάρχει MITRE ATT&CK mapping για αυτή την επίθεση
            msg += ( # Πρόσθεση MITRE πληροφοριών στο μήνυμα
```

```
                # Τίτλος (Bold, διπλό enter)
                "<br><br><b>MITRE ATT&CK:</b><br>"
                # Ταυτότητα, τεχνική, είδος επίθεσης
                f"{mitre['tactic']} – {mitre['technique']}
                      ({mitre['tech_id']})<br>"
                # Σύνδεσμος (επιλέξιμος)
                f"<a href='{mitre['url']}'>attack.mitre.org</a>"
          )
    # Εφαρμογή του beginner explanation μηνύματος
    self.xai_beginner_txt.setText(msg)
```

Στο κάτω μέρος δημιουργείται το γράφημα με Matplotlib. Οι SHAP τιμές που συλλέγονται αποθηκεύονται σε λίστες (labels και values) και στη συνέχεια σχεδιάζονται ως bar chart πάνω στον άξονα self.xai_ax, ο οποίος ανήκει στο FigureCanvas. Στις περιπτώσεις όπου υπάρχει διαθέσιμη επεξήγηση και από τα δύο μοντέλα, τα labels του γραφήματος φέρουν και ένδειξη προέλευσης, όπως CIC ή UNSW, ώστε να είναι σαφές από ποιο μοντέλο προέρχεται κάθε χαρακτηριστικό. Με κάθε νέα επιλογή alert, το γράφημα καθαρίζεται και επανασχεδιάζεται, αποτυπώνοντας τη συνεισφορά κάθε χαρακτηριστικού στην τελική πρόβλεψη.

```
# Απόσπασμα συνάρτησης render_full_xai()
self.xai_ax.clear() # Καθαρισμός αξόνων του matplotlib
    plot_labels = [] # Λίστα ετικετών
    plot_values = [] # Λίστα τιμών SHAP
    # Τίτλος γραφήματος
    chart_title = "Feature contribution overview"
    # Αν υπάρχουν αποτελέσματα SHAP από τα μοντέλα CIC ή UNSW
    if used_unsw_shap or used_cic_shap:
        # Προσθήκη ετικέτας UNSW στο γράφημα
        plot_labels.extend(unsw_labels)
        plot_values.extend(unsw_values) # Προσθήκη UNSW SHAP τιμής

        plot_labels.extend(cic_labels) # Προσθήκη CIC ετικέτας
        plot_values.extend(cic_values) # Προσθήκη CIC SHAP τιμής
        # Αλλαγή τίτλου γραφήματος
        chart_title = "Top supporting SHAP contributions"
    else: # Αν δεν υπάρχουν αποτελέσματα SHAP από τα μοντέλα CIC ή
UNSW
```

```
        # Πρώτα 5 χαρακτηριστικά (από raw data)
        plot_labels = fallback_labels[:5]
        plot_values = fallback_values[:5] # Αντίστοιχες τιμές
        # Εναλλακτικός τίτλος (fallback)
        chart_title = "Feature contribution overview"

    if plot_values: # Αν υπάρχουν δεδομένα για το διάγραμμα
        # Οριζόντια μπάρα με μπλε χρώμα
        bars = self.xai_ax.barh(plot_labels, plot_values,
              color="#3498db")
        self.xai_ax.set_title( # Τίτλος διαγράμματος
              chart_title, # Δυναμικός τίτλος
              fontsize=10, # Μέγεθος 10pt
              fontweight="bold", # Bold
        )
        # Μέγεθος ετικετών αξόνων (8pt)
        self.xai_ax.tick_params(axis="both", labelsize=8)
        # Αντιστροφή άξονα y (top χαρακτηριστικά πρώτα)
        self.xai_ax.invert_yaxis()

        for bar in bars: # Για κάθε bar του γραφήματος
              width = bar.get_width() # Πλάτος, τιμή SHAP
              # Προσθήκη αριθμητικής ετικέτας πάνω σε κάθε bar
              self.xai_ax.text(
                    width, # X θέση, πλάτος bar
                    # Y θέση, κέντρο bar
                    bar.get_y() + bar.get_height() / 2,
                    # 4 δεκαδικά για SHAP, 2 για σφάλμα (fallback)
                    f" {width:.4f}" if (used_unsw_shap or
              used_cic_shap)
                          else f" {width:.2f}",
                    va="center", # στοίχιση στο κέντρο
                    fontsize=8, # Μέγεθος 8pt
                    fontweight="bold" # αριθμοί Bold
              )
    else: # Αν δεν υπάρχουν δεδομένα για το διάγραμμα
        self.xai_ax.text( # Κεντρικό μήνυμα
              0.5, 0.5, # Κέντρο αξόνων
              "No feature data available", # Ενημερωτικό Μήνυμα
```

```
                ha="center", va="center", # στοίχιση στο κέντρο
                fontsize=9, # Μέγεθος 9pt
            )
        # Αυτόματη βελτιστοποίηση layout (χωρίς overlapping)
        self.xai_fig.tight_layout()
        # Ανανέωση/εμφάνιση του matplotlib
        self.xai_canvas.draw()
```

Τέλος, υπάρχει ένας επιπλέον fallback μηχανισμός, με βάση τον οποίο, αν για οποιονδήποτε λόγο αποτύχει η SHAP ανάλυση (για παράδειγμα επειδή δεν υπάρχει επεξήγηση ή αντίστοιχο μοντέλο) ή δεν έχει επαρκή δεδομένα, το σύστημα χρησιμοποιεί προκαθορισμένα χαρακτηριστικά (predefined behavioral features) από το αντίστοιχο προφίλ της επίθεσης και εμφανίζει τις τιμές τους ως εναλλακτική εξήγηση. Αυτό επιτυγχάνεται μέσω της εναλλακτικής λίστας που κατασκευάζεται (behavior profile indicators) η οποία χρησιμοποιείται τόσο στη λεκτική επεξήγηση όσο και στο γράφημα.

```
# Απόσπασμα συνάρτησης render_full_xai()
# Fallback αν δεν υπάρχουν αποτελέσματα SHAP από κανένα μοντέλο
fallback_labels = [] # Λίστα ετικετών του fallback γραφήματος
fallback_values = [] # Λίστα αριθμητικών τιμών fallback χαρακτηριστικών

if not sections: # Αν δεν υπάρχουν ενότητες εξήγησης
    fallback_reasons = [] # Λίστα εξηγήσεων
    # Για κάθε χαρακτηριστικό που ανήκει στο προφίλ συμπεριφοράς
    for feat in profile["features"]:
        val = row.get(feat) # Λήψη τιμής του χαρακτηριστικού
        if val is None: # Αν δεν βρέθηκε αντιστοίχιση
            # Δοκιμή ξανά με κομμένο όνομα
            val = row.get(feat.strip())
        # Αν υπάρχει έγκυρη τιμή
        if val is not None and pd.notna(val):
            try: # Προσπάθεια μετατροπής σε αριθμό
                num_val = float(val) # Μετατροπή σε float
                # Αν η τιμή είναι πάνω απο 0
                if abs(num_val) > 0:
                    # Προσθήκη μορφοποιημένης εξήγησης
                    fallback_reasons.append(f"{feat:<25
                        } : {num_val:.2f}")
```

```
                        # Προσθήκη ετικέτας γραφήματος
                        fallback_labels.append(feat)
                        # Προσθήκη αριθμητικής τιμής
                        fallback_values.append(num_val)
                # Αν η τιμή δεν μπορεί να γίνει float
                except (ValueError, TypeError):
                        # Αγνόησε το χαρακτηριστικό, συνέχισε με το
                επόμενο
                        continue
        # Αν βρέθηκαν fallback επεξήγησης
        if fallback_reasons:
                sections.append( # Προσθήκη στις εξηγήσεις
                        # Τίτλος ενότητας
                        "<b>Behavior profile indicators (
                                fallback explanation):</b><br>"
                                + "<br>".join(fallback_reasons[:5])
                ) # Μόνο τα πρώτα 5
        else: # Αν δεν βρέθηκε καμία fallback πληροφορία
                # Μήνυμα ότι δεν υπάρχουν δεδομένα εξήγησης
                sections.append("No SHAP explanation data available")
# Κείμενο για το XAI features label
features_text = (
        # Εμφάνιση τίτλου του behavior προφιλ
        f"<b>Behavior profile:</b> {profile['title']}<br><br>"
        # Ένωση όλων των ενοτήτων (με κενό διάστημα ανάμεσα)

        + "<br><br>".join(sections)
) # Εμφάνιση του τελικού κειμένου στο XAI UI
self.xai_features.setText(features_text)
```

8.4.6. Δημιουργία και διαχείριση αναφορών (Alerts & Reports)

Η λειτουργία της καρτέλας Alerts & Reports υλοποιείται μέσω μιας ροής επεξεργασίας δεδομένων και της δυναμικής ενημέρωσης της διεπαφής χρήστη. Το σύστημα δεν εμφανίζει απευθείας όλα τα δεδομένα που παράγονται από τη διαδικασία ανίχνευσης, καθώς πραγματοποιείται πρώτα ένας συγχρονισμός δεδομένων κατά τον οποίο τα προσωρινά δεδομένα που αποθηκεύονται στο logs_buffer ενοποιούνται με το κεντρικό ιστορικό καταγραφών all_logs. Η διαδικασία αυτή υλοποιείται μέσω της συνάρτησης

sync_data_buffer(), η οποία συγχωνεύει τα νέα δεδομένα σε μορφή DataFrame και καθαρίζει τον προσωρινό buffer, εξασφαλίζοντας ότι το σύστημα διατηρεί ένα συνεπές και ενημερωμένο σύνολο καταγραφών πριν από οποιαδήποτε προβολή.

```
# Συγχώνευση δεδομένων του buffer στο κύριο DataFrame
# Απόσπασμα συνάρτησης sync_data_buffer()
def sync_data_buffer(self):
      if self.logs_buffer: # Αν υπάρχουν δεδομένα στο buffer
            # Ένωση όλων των μικρών κομματιών σε ένα DataFrame
            new_chunk = pd.concat(self.logs_buffer, ignore_index=True)
            # Προσθήκη στο συνολικό ιστορικό
            self.all_logs = pd.concat([self.all_logs, new_chunk],
                  ignore_index=True)
            # Άδειασμα buffer μετά τη συγχώνευση
            self.logs_buffer = []
            # Ενημέρωση στατιστικών βάση όλων των δεδομένων
            self.update_attack_stats()

# Ανανέωση του πίνακα Reports (Απόσπασμα συνάρτησης)
def refresh_reports_table(self):
      self.sync_data_buffer() # Συγχρονισμός των πιο πρόσφατων δεδομένων
από buffer
```

Στη συνέχεια, κατά την ανανέωση της καρτέλας, εφαρμόζεται φιλτράρισμα πάνω στο σύνολο των καταγραφών, ώστε να διατηρηθούν αποκλειστικά οι ροές που έχουν χαρακτηριστεί ως κακόβουλες. Αυτό επιτυγχάνεται μέσω του πεδίου attack_merged, το οποίο περιέχει την ενοποιημένη απόφαση των μοντέλων ανίχνευσης, και με χρήση κατάλληλης συνθήκης επιλογής που διατηρεί μόνο τις εγγραφές που περιλαμβάνουν ένδειξη επίθεσης από τα σύνολα δεδομένων CIC-IDS2017 ή UNSW-NB15. Με τον τρόπο αυτό απορρίπτεται πλήρως η κανονική δικτυακή κίνηση, εξασφαλίζοντας ότι η καρτέλα Alerts & Reports παρουσιάζει μόνο τα σημαντικά περιστατικά ασφάλειας.

```
# Ανανέωση του πίνακα Reports (Απόσπασμα συνάρτησης)
def refresh_reports_table(self):
      # Συγχρονισμός των πιο πρόσφατων δεδομένων από buffer
      self.sync_data_buffer()
      try: # Μπλοκ διαχείρισης DataFrame
            # Αν υπάρχουν logs και δεν είναι άδειο το DataFrame
```

```
        if self.all_logs is not None and not self.all_logs.empty:
            df = self.all_logs.copy() # Αντιγραφή DataFrame
            df = df[ # Φιλτράρισμα γραμμών που περιέχουν επιθέσεις
                df["attack_merged"].str.contains( # Έλεγχος στη
            στήλη attack_merged
                    "CIC →|UNSW →", case=False, na=False
                )
            ]
```

Τα φιλτραρισμένα δεδομένα οργανώνονται σε έναν πίνακα, ο οποίος υλοποιείται με το στοιχείο QTableView και τροφοδοτείται μέσω ενός προσαρμοσμένου PandasModel. Κάθε γραμμή του πίνακα αντιστοιχεί σε μία ροή που έχει χαρακτηριστεί ως επίθεση, ενώ όλες οι διαθέσιμες στήλες του DataFrame προβάλλονται δυναμικά, χωρίς να απαιτείται χειροκίνητος ορισμός των πεδίων. Αυτό επιτρέπει την απεικόνιση ενός μεγάλου πλήθους χαρακτηριστικών, συμπεριλαμβανομένων των χρονικών πεδίων, των στοιχείων ταυτοποίησης της σύνδεσης, των στατιστικών της ροής και των εξειδικευμένων χαρακτηριστικών που προέρχονται από τα δύο σύνολα δεδομένων.

```
# Ανανέωση του πίνακα Reports (Απόσπασμα συνάρτησης)
def refresh_reports_table(self):
      # Εφαρμογή νέου μοντέλου στον πίνακα
      self.reports_table.setModel(model)
      # Αυτόματη προσαρμογή πλάτους στηλών
      self.reports_table.resizeColumnsToContents()
      # Ενημέρωση ετικέτας με αριθμό επιθέσεων
      self.attack_count_label.setText(f"Attacks Detected:
{attack_count}")
      # Επιπλέον στατιστικά επιθέσεων
      self.update_attack_stats()
```

Η οπτική επισήμανση των επιθέσεων υλοποιείται μέσω ενός μηχανισμού χρωματικής κατηγοριοποίησης, ο οποίος βασίζεται στο πεδίο color_tag. Εφόσον μια εγγραφή σχετίζεται με επίθεση, επιστρέφεται τιμή που αντιστοιχεί σε κόκκινη απόχρωση, και το PandasModel εφαρμόζει το αντίστοιχο χρώμα στο υπόβαθρο της γραμμής.

```
# Χρώμα μιας γραμμής με βάση το περιεχόμενο της στήλης attack_merged
# (Απόσπασμα συνάρτησης)
```

```
def assign_color(row):
      # Μετατροπή της τιμής της attack_merged σε string (πεζά)
      val = str(row["attack_merged"]).lower()
      # Αν εμπεριέχεται ανίχνευση από CIC ή UNSW
      if "cic →" in val or "unsw →" in val:
            return "red" # Επιστροφή κόκκινο
      elif "normal" in val: # Αν εμπεριέχεται το normal
            return "green" # Επιστροφή πράσινο
      else: # Αλλιώς
            return "gray"  # Επιστροφή γκρί
      # Εφαρμογή σε κάθε γραμμή, αποθήκευση αποτελεσμάτων στη στήλη
color_tag
      df["color_tag"] = df.apply(assign_color, axis=1)
```

Παράλληλα, το σύστημα υπολογίζει και εμφανίζει τον συνολικό αριθμό των επιθέσεων που έχουν εντοπιστεί, μέσω μιας απλής μέτρησης του πλήθους των γραμμών του φιλτραρισμένου DataFrame. Η τιμή αυτή προβάλλεται σε ειδική ετικέτα στην κορυφή της σελίδας και ενημερώνεται δυναμικά σε κάθε ανανέωση, παρέχοντας την άμεση εικόνα του όγκου των περιστατικών. Επιπλέον, πριν από την απεικόνιση, τα δεδομένα ταξινομούνται με βάση το πεδίο Timestamp σε φθίνουσα σειρά, ώστε οι πιο πρόσφατες επιθέσεις να εμφανίζονται πρώτες και να διευκολύνεται η εστίαση στα νεότερα συμβάντα.

```
# Ανανέωση του πίνακα Reports (Απόσπασμα συνάρτησης)
def refresh_reports_table(self):
      # Μετρητής αριθμού ανιχνευμένων επιθέσεων
      attack_count = len(df)

      if df.empty: # Αν δεν βρέθηκαν επιθέσεις μετά το φιλτράρισμα
            # Ενημερωτικό μήνυμα
            df = pd.DataFrame([{"Info": "No attacks were detected."}])
      else: # Αν βρέθηκαν επιθέσεις μετά το φιλτράρισμα
            if "Timestamp" in df.columns: # Αν υπάρχει στήλη Timestamp
                  # Ταξινόμηση (νεότερες πρώτες)
                  df = df.sort_values("Timestamp", ascending=False)
                        # Δημιουργία table model με τα φιλτραρισμένα
                  δεδομένα
                  model = PandasModel(
                        # Επαναφορά DataFrame
```

```
                df.reset_index(drop=True),
            dark_mode=(self.theme_choice == "Dark")
            )
            # Εφαρμογή νέου μοντέλου στον πίνακα
            self.reports_table.setModel(model)
            # Αυτόματη προσαρμογή πλάτους στηλών
            self.reports_table.resizeColumnsToContents()

            # Ενημέρωση ετικέτας με αριθμό επιθέσεων
            self.attack_count_label.setText(f"Attacks Detected:
{attack_count}")
            # Επιπλέον στατιστικά επιθέσεων
            self.update_attack_stats()
        else:
            # Αν δεν υπάρχουν logs
            # Μετρητής επιθέσεων 0
            self.attack_count_label.setText("Attacks Detected: 0")
```

Η καρτέλα υποστηρίζει επίσης διαδραστικότητα, καθώς κάθε γραμμή του πίνακα είναι επιλέξιμη. Μέσω σύνδεσης του γεγονότος επιλογής (clicked) με τη συνάρτηση χειρισμού on_table_row_clicked(), το σύστημα ανακτά τα δεδομένα της επιλεγμένης ροής, τα αποθηκεύει για ανάλυση και τα προωθεί στην καρτέλα Explainability. Εκεί ενεργοποιείται η διαδικασία ανάλυσης και ερμηνείας, επιτρέποντας την περαιτέρω διερεύνηση του περιστατικού.

```
# Δημιουργία σελίδας Reports (Απόσπασμα συνάρτησης)
def create_reports_page(self):
    page = QWidget() # Κύριο container widget για τη σελίδα Reports

    label = QLabel("Alerts & Reports") # Τίτλος σελίδας
    label.setAlignment(Qt.AlignCenter) # Κεντράρισμα κειμένου
οριζόντια

    # Πίνακας εμφάνισης επιθέσεων
    self.reports_table = QTableView() # Πίνακας για εμφάνιση logs
επιθέσεων
    # Σύνδεση επιλεξιμότητας (on click) με XAI analysis
    self.reports_table.clicked.connect(self.on_table_row_clicked)
    layout.addWidget(self.reports_table) # Προσθήκη στο layout
```

```
# Χειρισμός του κλικ του χρήστη σε μια γραμμή του πίνακα
# (Απόσπασμα συνάρτησης)
def on_table_row_clicked(self, index):
      # Ανάκτηση ολόκληρης της γραμμής
      row_data = model._df.iloc[row_idx]
      # Αποθήκευση της επιλεγμένης γραμμής
      self.current_explained_alert = row_data
      self.user_selected_alert = True # Επιβεβαίωση χειροκίνητου alert
      self.update_xai_preview_from_row(row_data) # Ενημέρωση XAI preview
      # Μετάβαση στη σελίδα XAI
      self.content_area.setCurrentWidget(self.xai_page)
```

Ένα ακόμη βασικό χαρακτηριστικό της υλοποίησης είναι η αυτόματη ανανέωση της καρτέλας. Μέσω της συνάρτησης auto_refresh_reports_page(), το σύστημα ελέγχει αν η συγκεκριμένη σελίδα είναι ενεργή και, εφόσον ισχύει αυτό, καλεί τη διαδικασία ανανέωσης του πίνακα. Η επιλογή αυτή αποτρέπει περιττές ενημερώσεις όταν ο χρήστης βρίσκεται σε άλλη καρτέλα και ταυτόχρονα διασφαλίζει ότι τα δεδομένα παραμένουν συνεχώς επικαιροποιημένα όταν η καρτέλα είναι ορατή. Συνολικά, η λειτουργία της καρτέλας βασίζεται σε έναν συνεχή κύκλο εισαγωγής νέων δεδομένων, συγχρονισμού, φιλτραρίσματος και δυναμικής απεικόνισης.

```
# Αυτόματη ανανέωση της σελίδας Reports
def auto_refresh_reports_page(self):
      try: # Μπλοκ ανανέωσης
            # Αν η σελίδα είναι ενεργή (επιλεγμένη)
            if self.content_area.currentWidget() == self.reports_page:
                  # Ανανέωση με τα πιο πρόσφατα δεδομένα
                  self.refresh_reports_table()
      except Exception as e: # Αν υπάρξει σφάλμα
            print(f"Auto refresh reports error: {e}") # Εμφάνιση στη
      κονσόλα
            # Καταγραφή στο Log file
            logging.error(f"Auto refresh reports error: {e}",
      exc_info=True)
```

8.4.7. Ανάλυση καταγραφών (Logs Analysis)

Για την υλοποίηση της καρτέλας Logs Analysis το σύστημα βασίζεται σε μια κεντρική δομή δεδομένων τύπου Pandas DataFrame (self.all_logs), στην οποία διατηρείται και το σύνολο

των καταγραφών. Σε αντίθεση με την καρτέλα Reports, όπου εφαρμόζεται φιλτράρισμα ώστε να εμφανίζονται μόνο εγγραφές που έχουν χαρακτηριστεί ως επιθέσεις, στην ενότητα Logs Analysis χρησιμοποιείται ένα πλήρες αντίγραφο των δεδομένων χωρίς αποκλεισμούς, επιτρέποντας με αυτό το τρόπο την εξέταση τόσο της κακόβουλης όσο και της φυσιολογικής ή της μη ταξινομημένης δραστηριότητας.

```
# Αποθήκευση όλων των logs για πλήρη προβολή
self.all_logs = pd.DataFrame()
# Πλήρες αντίγραφο των δεδομένων
df = self.all_logs.copy()
```

Η ενημέρωση των δεδομένων πραγματοποιείται μέσω ενός προσωρινού καταχωρητή (buffer) στη μνήμη (self.logs_buffer), στον οποίο αποθηκεύονται αρχικά όλες οι νέες εγγραφές. Έτσι, αντί να ενημερώνεται απευθείας το κύριο DataFrame με κάθε νέα καταγραφή, οι εγγραφές συγκεντρώνονται και συγχωνεύονται μαζικά μέσω της συνάρτησης pd.concat, μειώνοντας το υπολογιστικό κόστος και βελτιώνοντας την απόδοση της εφαρμογής, ιδιαίτερα σε περιπτώσεις διαχείρισης μεγάλου όγκου δεδομένων.

```
# Προσθήκη των logs στον buffer (Απόσπασμα συνάρτησης)
def append_logs(self, df):
      # Προσθήκη νέων logs στον buffer
      self.logs_buffer.append(df)


# Συγχώνευση δεδομένων του buffer στο κύριο DataFrame (Απόσπασμα συνάρτησης)
def sync_data_buffer(self):
      # Ένωση όλων των μικρών κομματιών σε ένα DataFrame
      new_chunk = pd.concat(self.logs_buffer, ignore_index=True)
      # Προσθήκη στο συνολικό ιστορικό
      self.all_logs = pd.concat([self.all_logs, new_chunk],
            ignore_index=True)
      # Άδειασμα buffer μετά τη συγχώνευση
      self.logs_buffer = []
```

Το φιλτράρισμα των καταγραφών πραγματοποιείται με βάση το χρονικό διάστημα που ορίζεται από τον χρήστη μέσω των δύο πεδίων ημερομηνίας. Οι τιμές αυτές μετατρέπονται σε αντικείμενα datetime και εφαρμόζεται λογική μάσκα (boolean mask) πάνω στη στήλη

Timestamp του DataFrame, προκειμένου να διατηρηθούν μόνο οι εγγραφές που εμπίπτουν στο επιλεγμένο εύρος. Στη συνέχεια, τα δεδομένα ταξινομούνται ως προς τον χρόνο σε φθίνουσα σειρά, διασφαλίζοντας ότι οι πιο πρόσφατες εγγραφές εμφανίζονται πρώτες στον πίνακα.

```
# Ανανέωση του πίνακα Logs (Απόσπασμα συνάρτησης)
def refresh_logs_table(self):
      # Αρχικοποίηση - Έναρξη ημέρας (00:00:00)
      start_dt = datetime.combine(
            self.log_date_from.date().toPyDate(), datetime.min.time()
      )
      # Αρχικοποίηση - Τέλος ημέρας (23:59:59)
      end_dt = datetime.combine(
            self.log_date_to.date().toPyDate(), datetime.max.time()
      )
      # Εφαρμογή φίλτρου
      # Μάσκα (boolean) - Επιλογή logs εντός ορισμένου χρονικού
διαστήματος
      mask = (df["Timestamp"] >= start_dt) & (df["Timestamp"] <= end_dt)
      df = df.loc[mask]  # Φιλτράρισμα του DataFrame με βάση τη μάσκα

      # Ταξινόμηση (από νεότερη προς παλαιότερη)
      df = df.sort_values("Timestamp", ascending=False)
```

Ο πίνακας καταγραφών υλοποιείται μέσω ενός προσαρμοσμένου (custom) μοντέλου (PandasModel) που συνδέεται με ένα QTableView, παρέχοντας τη δυνατότητα για απευθείας απεικόνιση όλων των στηλών του DataFrame. Μεταξύ αυτών περιλαμβάνονται τόσο τα χαρακτηριστικά της δικτυακής ροής όσο και τα αποτελέσματα των μοντέλων ανίχνευσης (όπως attack_cic, pred_unsw, attack_unsw, prediction, attack_type). Παράλληλα, πραγματοποιείται επιπλέον επεξεργασία στη στήλη attack_merged, η οποία μετακινείται αμέσως μετά το πεδίο Timestamp και μετονομάζεται σε "Attack Type", ώστε να αναδεικνύεται οπτικά το βασικό συμπέρασμα για κάθε εγγραφή.

```
# Ανανέωση του πίνακα Logs (Απόσπασμα συνάρτησης)
def refresh_logs_table(self):
      model = PandasModel( # Νέο table model
            # Reset index και υποστήριξη dark mode
            df.reset_index(drop=True), dark_mode=
```

```
                (self.theme_choice == "Dark")
        )
        self.logs_table.setModel(model) # Εφαρμογή μοντέλου στον πίνακα
logs
        self.logs_table.resizeColumnsToContents() # Αυτόματη προσαρμογή
στηλών


        # Αν υπάρχουν οι στήλες που θα αναδιοργανωθούν
        if "attack_merged" in df.columns and "Timestamp" in df.columns:
                cols = df.columns.tolist() # Λίστα με ονόματα στηλών
                # Αφαίρεση στήλης attack_merged από την τρέχουσα θέση
                cols.remove("attack_merged")
                # Θέση, μετά το Timestamp
                idx = cols.index("Timestamp") + 1
                # Εισαγωγή attack_merged δίπλα στο Timestamp
                cols.insert(idx, "attack_merged")
                df = df[cols] # Εφαρμογή νέας σειράς στηλών
                # Μετονομασία σε Attack Type
                df.rename(columns={"attack_merged": "Attack Type"},
        inplace=True)
```

Για τη καλύτερη αναγνωσιμότητα, εφαρμόζεται δυναμικά, κατά την απόδοση των γραμμών στον πίνακα, μηχανισμός χρωματικής κατηγοριοποίησης, σύμφωνα με τον οποίο κάθε εγγραφή λαμβάνει μια ετικέτα (color_tag) ανάλογα με τη τιμή που περιέχει η στήλη attack_merged. Έτσι, οι εγγραφές που περιέχουν ένδειξη κανονικής λειτουργίας επισημαίνονται με πράσινο, οι εγγραφές που σχετίζονται με επιθέσεις με κόκκινο και οι μη ταξινομημένες περιπτώσεις με γκρι (όπως περιγράφεται στην ενότητα 9.4.6, συνάρτηση assign_color())

```
# Δεδομένα ή ιδιότητες για κάθε κελί
def data(self, index, role=Qt.DisplayRole):
        if not index.isValid(): # Αν ο δείκτης του κελιού είναι έγκυρος
                return None # Αν δεν είναι επιστροφή
        if role == Qt.DisplayRole: # Αν ζητείται το περιεχόμενο κελιού
                # Επιστροφή τιμής του DataFrame στη θέση [row, column] ως
        string
                return str(self._df.iloc[index.row(), index.column()])
```

```
      if role == Qt.BackgroundRole: # Αν ζητειται το χρώμα φόντου του
κελιού
            try:
                  # Αν υπάρχει η στήλη color_tag
                  if "color_tag" in self._df.columns:
                        color = str( # Τιμή color_tag για την τρέχουσα
                  γραμμή
                              self._df.iloc[ # Πρόσβαση στο DataFrame
                                    # Γραμμή κελιού, θέση στήλης
                              color_tag
                                    index.row(),
                              self._df.columns.get_loc
                                          ("color_tag")
                              ]
                        ).lower() # Μετατροπή σε πεζά


                        if self.dark_mode: # Αν το UI είναι σε dark mode
                              if color == "red": # Αν το color_tag είναι
                        red
                                    # Σκούρο κόκκινο φόντο (ημιδιαφανές)
                                    return QtGui.QBrush(
                                          QtGui.QColor(140, 30, 30,
                                    200))
                              # Αν το color_tag είναι green
                              elif color == "green":
                                    # Σκούρο πράσινο φόντο
                                    return QtGui.QBrush(
                                          QtGui.QColor("#277247"))
                              # Αν το color_tag είναι gray
                              elif color == "gray":
                                    # Σκούρο γκρι φόντο
                                    return
                              QtGui.QBrush(QtGui.QColor("#2a2a2a
                        else: # Αν το UI δεν είναι σε dark mode
                              if color == "red": # Αν το color_tag είναι
                        red
                                    # Ανοιχτό κόκκινο φόντο
                                    return QtGui.QBrush(
                                          QtGui.QColor(255, 200, 200))
                              # Αν το color_tag είναι green
                              elif color == "green":
                                    # Ανοιχτό πράσινο φόντο
```

```
                                    return QtGui.QBrush(
                                          QtGui.QColor(200, 255, 200))
                                # Αν το color_tag είναι gray
                                elif color == "gray":
                                      # Ανοιχτό γκρι φόντο
                                      return QtGui.QBrush(
                                            QtGui.QColor(230, 230, 230))
            except Exception: # Σφάλμα κατά το χρωματισμό
                  pass # Συνέχισε με τα υπόλοιπα

      return None # Αν ζητείται άλλο, δεν επιστρέφει κάτι
```

Η ανανέωση του πίνακα πραγματοποιείται είτε αυτόματα είτε μέσω χειροκίνητης εντολής που καλεί τη συνάρτηση refresh_logs_table. Η διαγραφή των καταγραφών, όπως και των πληροφοριών που φέρουν τα επιμέρους στοιχεία (π.χ. μετρητής επιθέσεων, σχετικές καρτες και δείκτες του dashboard κλπ), γίνεται με μηχανισμό επιβεβαίωσης μέσω QMessageBox, ώστε να αποτρέπεται η ακούσια απώλεια δεδομένων κι επιπλέον, παρέχεται η δυνατότητα εξαγωγής των δεδομένων σε αρχείο Excel (.xlsx) το οποίο μορφοποιείται (όπως ο πίνακας του UI) με styling του Pandas (DataFrame.style).

```
# Σύνδεση κουμπιού ανανέωσης με τη συνάρτηση που ξαναγεμίζει τον πίνακα logs
refresh_btn.clicked.connect(self.refresh_logs_table)


# Διαγραφή όλων των logs και reset στο dashboard
def clear_logs(self):
      confirm = QMessageBox.question(  # Διάλογος επιβεβαίωσης στον χρήστη
            self,
            "Clear Logs", # Τίτλος παραθύρου
            # Μήνυμα επιβεβαίωσης
            "Are you sure you want to delete ALL logs and reset all
                  dashboards?",
            QMessageBox.Yes | QMessageBox.No, # Επιλογές Yes/No
            QMessageBox.No, # Προεπιλεγμένη επιλογή το No
      )
      # Αν επιλέχθηκε το Yes
      if confirm == QMessageBox.Yes:
```

```
            try: # Μπλόκ εκκαθάρισης
                  # Άδειασμα του DataFrame με όλα τα logs
                  self.all_logs = pd.DataFrame()

            if os.path.exists(self.logs_file): # Αν υπάρχει αρχείο logs
                  # Διαγραφή του αρχείου από τον δίσκο
                  os.remove(self.logs_file)
            # Τοποθέτηση άδειου model στον πίνακα
            self.logs_table.setModel(PandasModel(pd.DataFrame()))
            self.logs_label.setText("0") # Μηδενισμός ετικέτας πλήθους
      logs
            self.total_logs = 0 # Μηδενισμός εσωτερικού μετρητή logs
            # Ανανέωση reports (για να μην δείχνει παλιά δεδομένα)
            self.refresh_reports_table()
            # Αν υπάρχει η ετικέτα μετρητή επιθέσεων
            if hasattr(self, "attack_count_label"):
                  # Μηδενισμός μετρητή
                  self.attack_count_label.setText("Attacks Detected: 0")
            # Εκκαθάριση KPI gauges/summary data
            self.exec_dashboard_page.update_executive_data(pd.DataFrame(
      ))
            # Αν υπάρχει η κάρτα alerts
            if hasattr(self, "alerts_card"):
                  # Τοποθέτησε κενό DataFrame (για να αδειάσει)
                  self.alerts_card.update_alerts(pd.DataFrame())
            # Μήνυμα επιτυχίας
            QMessageBox.information(
                  self, "Success", "All logs cleared and dashboards
            reset."
            )
      except Exception as e: # Σφάλμα κατά την εκκαθάριση
      # Καταγραφή του σφάλματος στο log
      logging.error(f"Error", f"Error while clearing:{e}",
exc_info=True)
      # Εμφάνιση μηνύματος με σφάλμα
      QMessageBox.critical(self, "Error", f"Error while clearing: {e}")

# Εφαρμογή συνάρτησης μορφοποίησης σε κάθε γραμμή του DataFrame
styled_df = df.style.apply(highlight_threats, axis=1)
```

```
# Εξαγωγή των logs σε Excel με ίδια μορφοποίηση όπως του πίνακα.
# Απόσπασμα Συνάρτησης export_to_excel()
def export_to_excel(self):
      df = self.all_logs.copy() # Αντίγραφο του DataFrame logs
      if "Timestamp" in df.columns:
            # Ταξινόμηση από το πιο πρόσφατο προς το παλαιότερο
            df = df.sort_values("Timestamp", ascending=False)

      def highlight_threats(row): # Μορφοποίηση πίνακα
            # Τιμή της στήλης Attack Type (πεζά)
            attack_val = str(row.get("Attack Type", "")).lower()
            # Αν η στήλη έχει normal
            if "normal" in attack_val or "green" in attack_val:
                  # Ανοιχτό Πράσινο
                  return ["background-color: #80ed84"] * len(row)
            elif ( # Αν η στήλη έχει cic, unsw ή attack
                  "cic" in attack_val
                  or "unsw" in attack_val
                  or "attack" in attack_val
            ):
                  # Ανοιχτό Κόκκινο
                  return ["background-color: #ff808b"] * len(row)

            else: # Αλλιώς (μη ταξινομημένο ή nan)
                  return ["background-color: #e2e3e5"] * len(row) # Γκρι

      # Pandas Styler με βάση τη συνάρτηση highlight_threats
      styled_df = df.style.apply(highlight_threats, axis=1)

      # Παράθυρο αποθήκευσης αρχείου
      save_path, _ = QFileDialog.getSaveFileName(
            # Τίτλος και φίλτρο xlsx
            self, "Save Logs as Excel", "", "Excel Files (*.xlsx)"
      )
      # Εξαγωγή DataFrame σε Excel (χωρίς index)
      styled_df.to_excel(save_path, engine="openpyxl", index=False)
```

Σε επίπεδο διαχείρισης και αποθήκευσης, οι καταγραφές εγγράφονται σε αρχείο CSV στο σύστημα αρχείων της εφαρμογής, μέσω της συνάρτησης append_logs(), όπως περιγράφεται στην ενότητα 9.4.3. Όταν το μέγεθος του αρχείου υπερβεί το όριο των 100MB, ενεργοποιείται ο μηχανισμός log rotation, κατά τον οποίο το αρχείο μετονομάζεται και μετακινείται σε φάκελο αρχειοθέτησης με προσθήκη χρονικής σήμανσης στο όνομα, ενώ ταυτόχρονα δημιουργείται νέο ενεργό αρχείο για τη συνέχεια της καταγραφής. Κατά την εκκίνηση της εφαρμογής φορτώνονται, μέσω της load_logs_from_file(), τόσο το κύριο CSV αρχείο όσο και τα αρχειοθετημένα, τα οποία συγχωνεύονται σε ένα ενιαίο DataFrame, εξασφαλίζοντας ένα πλήρες ιστορικό.

```
# Φορτώνει όλα τα logs από το κύριο αρχείο και τα backups
def load_logs_from_file(self):
      try: # Μπλόκ φόρτωσης αρχείων
            all_dfs = [] # Λίστα για DataFrames των αρχείων logs
            # Αν υπάρχει το κύριο αρχείο logs.csv
            if os.path.exists(self.logs_file):
                  # Φόρτωση και προσθήκη στη λίστα
                  all_dfs.append(pd.read_csv(self.logs_file))

            # Φάκελος με το κύριο logs.csv
            base_folder = os.path.dirname(self.logs_file)
            # Υποφάκελος για τα αρχειοθετημένα
            archive_folder = os.path.join(base_folder, "logs")
            # Αν υπάρχει ο φάκελος με τα παλιά αρχεία
            if os.path.exists(archive_folder):
                  # Για κάθε αρχείο του φακέλου
                  for filename in os.listdir(archive_folder):
                        # Αν το όνομα αρχείου ξεκινάει logs_ και είναι
            .csv
                        if filename.startswith("logs_") and
                              filename.endswith(".csv"):
                              # Πλήρες μονοπάτι του αρχείου
                              file_path = os.path.join(archive_folder,
                                    filename)
                              try: # Μπλόκ φόρτωσης αρχείου
                                    # Φόρτωση του αρχειοθετημένου
                              αρχείου
                                    archive_df = pd.read_csv(file_path)
```

```
                        # Προσθήκη στη λίστα των DataFrames
                        all_dfs.append(archive_df)
                    except Exception as e: # Σφάλμα
                        # Μήνυμα για το συγκεκριμένο αρχείο
                        print(f"Error loading archive file
                            {filename}: {e}")

        if all_dfs: # Αν βρέθηκαν logs
            # Συνένωση όλων των DataFrames
            self.all_logs = pd.concat(all_dfs, ignore_index=True)
            # Αν υπάρχει στήλη Timestamp
            if not self.all_logs.empty and "Timestamp" in
                self.all_logs.columns:
                self.all_logs.drop_duplicates( # Αφαίρεση
            διπλότυπων
                    # Βάση συνένωσης χρονοσφραγίδας και IPs
                    subset=["Timestamp", "srcip", "dstip"],
                        inplace=True
                )
            # Μήνυμα στη κονσόλα με συνολικό αριθμό των logs
            print(f"Loaded total {len(self.all_logs)} logs
                (including archives).")
        else: # Αν δεν βρέθηκαν logs
            # Αρχικοποίηση (άδειου) DataFrame
            self.all_logs = pd.DataFrame()

    except Exception as e: # Αν υπάρξει σφάλμα κατά τη φόρτωση
        # Καταγραφή σφάλματος στο log
        logging.error(f"Error loading logs: {e}", exc_info=True)
        print(f"Error loading logs: {e}") # Μήνυμα σφάλματος
        # Fallback σε άδειο DataFrame (για ασφάλεια)
        self.all_logs = pd.DataFrame()
```

Η εφαρμογή, μέσω του χρονομετρητή logs_refresh_timer ο οποίος ενεργοποιείται κάθε 5 δευτερόλεπτα και καλεί τη συνάρτηση auto_refresh_logs_page(), υποστηρίζει αυτόματη ανανέωση της προβολής όταν η καρτέλα Logs είναι ενεργή, ελέγχοντας ποια σελίδα εμφανίζεται και καλώντας περιοδικά τη διαδικασία ενημέρωσης. Παράλληλα, εμφανίζεται η ένδειξη της τελευταίας χρονικής στιγμής ανανέωσης, προκειμένου να γνωρίζει ο χρήστης

πότε έγινε η τελευταία επικαιροποίηση των δεδομένων. Σε περιπτώσεις όπου το εφαρμοζόμενο φιλτράρισμα δεν επιστρέφει κάποιο αποτελέσματα, ο πίνακας παραμένει κενός (αντικαθιστά το περιεχόμενό του με ένα κενό PandasModel) και εμφανίζεται σχετικό μήνυμα ενημέρωσης.

```
# Αυτόματη ανανέωση της καρτέλας Logs
def auto_refresh_logs_page(self):
      try: # Μπλοκ ανανέωσης
            # Αν αυτή τη στιγμή η σελίδα Logs είναι ενεργή (επιλεγμένη)
            if self.content_area.currentWidget() == self.logs_page:
                  self.refresh_logs_table() # Ανανέωση του πίνακα
                  # Τρέχουσα ώρα
                  now_str = datetime.now().strftime("%H:%M:%S")
                  # Ενημέρωση ετικέτας με την ώρα ανανέωσης
                  self.logs_status_label.setText(f"Last updated:
            {now_str}")
      except Exception as e: # Σφάλμα κατά την ανανέωση
            # Καταγραφή πλήρους σφάλματος
            logging.error(f"Auto refresh logs error: {e}",
            exc_info=True)
            print(f"Auto refresh logs error: {e}") # Εμφάνιση στη
      κονσόλα


# QTimer για περιοδική ανανέωση της καρτέλας Logs
self.logs_refresh_timer = QTimer()
# Σύνδεση timeout() με τη συνάρτηση ανανέωσης
self.logs_refresh_timer.timeout.connect(self.auto_refresh_logs_page)
self.logs_refresh_timer.start(5000)  # Ανανέωση κάθε 5 δευτερόλεπτα
```

8.4.8. Έλεγχος και διαχείριση IP (IP Control)

Ο έλεγχος και η διαχείριση των IP διευθύνσεων υλοποιείται μέσω συγκεκριμένων δομών δεδομένων, μηχανισμών αποθήκευσης, διεπαφής χρήστη και λογικής επεξεργασίας των ροών. Οι βασικές δομές από τις οποίες ξεκινάει όλη η διαχείριση είναι οι self.whitelist και self.blacklist, όπου ορίζονται ως σύνολα (set) προκειμένου να είναι εφικτή η αποδοτική αναζήτηση και η αποφυγή διπλών εγγραφών. Οι δομές αυτές αποτελούν το κομμάτι του συστήματος, στο οποίο καταγράφονται οι αξιόπιστες και οι κακόβουλες διευθύνσεις IP αντίστοιχα.

```
self.whitelist = set() # Αρχικοποίηση της whitelist σε κενό σύνολο
self.blacklist = set() # Αρχικοποίηση της blacklist σε κενό σύνολο
```

Για να μπορεί να διατηρείται η πληροφορία μεταξύ των διαφορετικών εκτελέσεων της εφαρμογής, υλοποιείται ένας μηχανισμός μόνιμης αποθήκευσης μέσω των συναρτήσεων save_ip_lists() και load_ip_lists(). Η αποθήκευση σε αυτή τη περίπτωση πραγματοποιείται με τη βοήθεια της βιβλιοθήκης pickle, σύμφωνα με την οποία οι λίστες που δημιουργούνται εσωτερικά, μπορούν να μετατραπούν σε Python λίστες και να αποθηκευτούν σε εξωτερικό αρχείο (ip_lists.pkl), έτσι ώστε κατά την εκκίνηση της εφαρμογής να επαναφορτώνονται και να μετατρέπονται εκ νέου σε set (εξασφαλίζοντας έτσι τη συνέχεια της πολιτικής ασφάλειας χωρίς την ανάγκη επανακαθορισμού των IP διευθύνσεων κάθε φορά που εκτελείται η εφαρμογή).

```
# Αποθήκευση λιστών IP (whitelist και blacklist) – Απόσπασμα συνάρτησης
def save_ip_lists(self):
      # Άνοιγμα αρχείου σε binary write mode για αποθήκευση των δεδομένων
      with open(self.ip_lists_file, "wb") as f:
            pickle.dump( # Δεδομένα σε pickle και εγγραφή στο αρχείο
                  {
                        # Μετατροπή της whitelist από set σε list
                        "whitelist": list(self.whitelist),
                        # Μετατροπή της blacklist από set σε list
                        "blacklist": list(self.blacklist),
                  },
                  f, # Το αρχείο που θα γραφτούν τα δεδομένα
            )

# Φόρτωση αποθηκευμένων λιστών IP (whitelist και blacklist)
def load_ip_lists(self):
      try:  # Μπλοκ φόρτωσης λιστών
            # Αν υπάρχει το αρχείο που είναι αποθηκευμένες οι λίστες
            if os.path.exists(self.ip_lists_file):
                  # Άνοιγμα σε binary read mode
                  with open(self.ip_lists_file, "rb") as f:
                        data = pickle.load(f)  # Δεδομένα ως Python
```

```
                # Φόρτωση της whitelist ως set ή κενό set αν δεν υπάρχει
                self.whitelist = set(data.get("whitelist", []))
                # Φόρτωση της blacklist ως set ή κενό set αν δεν υπάρχει
                self.blacklist = set(data.get("blacklist", []))
        except Exception as e:  # Σφάλμα φόρτωσης λιστών
            print(f"IP list load error: {e}") # Εμφάνιση σφάλματος στη κονσόλα
            # Καταγραφή του σφάλματος στο log
            logging.error(f"IP list load error: {e}", exc_info=True)
            self.whitelist = set() # Αρχικοποίηση της whitelist σε κενό σύνολο
            self.blacklist = set() # Αρχικοποίηση της blacklist σε κενό σύνολο
```

Η διαχείριση των λιστών από τον χρήστη πραγματοποιείται εξ' ολοκλήρου μέσω του UI, μέσω συναρτήσεων όπως η add_to_whitelist() και η add_to_blacklist(), με τις οποίες πραγματοποιείται η εισαγωγή των νέων διευθύνσεων. Επιπλέον, για τη διατήρηση της συνέπειας μεταξύ των λιστών, όταν μια IP προστίθεται στη whitelist, ταυτόχρονα αφαιρείται από τη blacklist με τη χρήση της μεθόδου discard(), ενώ το ίδιο ισχύει και αντίστροφα. Η αφαίρεση των διευθύνσεων πραγματοποιείται επίσης μέσω της discard(), ενώ η λειτουργία αναζήτησης υλοποιείται με τη συνάρτηση filter_ip_lists(text), η οποία φιλτράρει δυναμικά τα στοιχεία της διεπαφής συγκρίνοντας το κείμενο αναζήτησης με τις εγγραφές των λιστών.

```
# Προσθήκη IP στη whitelist
def add_to_whitelist():
    ip = ip_input.text().strip() # Λήψη IP (Αφαίρεση κενών)
    if not ip: # Αν δεν είναι IP
        return # Τερματισμός
    self.whitelist.add(ip) # Προσθήκη στη whitelist
    self.blacklist.discard(ip) # Αφαίρεση απο τη blacklist
    self.save_ip_lists() # Αποθήκευση λίστας
    ip_input.clear() # Εκκαθάριση του input πεδίου
    self.refresh_ip_ui_lists() # Ανανέωση των πινάκων UI
```

```
# Προσθήκη IP στη blacklist
def add_to_blacklist():
	ip = ip_input.text().strip() # Λήψη IP (Αφαίρεση κενών)
	if not ip: # Αν δεν είναι IP
		return # Τερματισμός
	self.blacklist.add(ip) # Προσθήκη στη blacklist
	self.whitelist.discard(ip) # Αφαίρεση απο τη whitelist
	self.save_ip_lists() # Αποθήκευση λίστας
	ip_input.clear() # Εκκαθάριση του input πεδίου
	self.refresh_ip_ui_lists() # Ανανέωση των πινάκων UI

# Αφαίρεση IP στη whitelist
def remove_from_whitelist():
	# Λήψη επιλεγμένου στοιχείου από τον πίνακα whitelist
	item = self.wl_list.currentItem()
	if not item: # Αν δεν έχει επιλεχθεί κάποιο στοιχείο
		return # Τερματισμός
	self.whitelist.discard(item.text()) # Αφαίρεση IP από τη λίστα
	self.save_ip_lists() # Αποθήκευση λίστας
	self.refresh_ip_ui_lists() # Ανανέωση των πινάκων UI

# Αφαίρεση IP στη blacklist
def remove_from_blacklist():
	# Λήψη επιλεγμένου στοιχείου από τον πίνακα blacklist
	item = self.bl_list.currentItem()
	if not item: # Αν δεν έχει επιλεχθεί κάποιο στοιχείο
		return # Τερματισμός
	self.blacklist.discard(item.text()) # Αφαίρεση IP από τη λίστα
	self.save_ip_lists() # Αποθήκευση λίστας
	self.refresh_ip_ui_lists() # Ανανέωση των πινάκων UI

# Φιλτράρισμα των πινάκων IP με βάση κείμενο αναζήτησης (Απόσπασμα συνάρτησης)
def filter_ip_lists(text):
	# Μετατροπή του κειμένου σε πεζά, αφαίρεση κενών χαρακτήρων
	text = text.lower().strip()
	# Για κάθε στοιχείο του πίνακα whitelist
	for i in range(self.wl_list.count()):
		item = self.wl_list.item(i) # Τρέχων στοιχείο του πίνακα
```

```
            # Θέσε κρυφό αν το κείμενο δεν περιέχεται στην IP
            item.setHidden(text not in item.text().lower())
            # Για κάθε στοιχείο του πίνακα blacklist
      for i in range(self.bl_list.count()):
            item = self.bl_list.item(i) # Τρέχων στοιχείο του πίνακα
            # Θέσε κρυφό αν το κείμενο δεν περιέχεται στην IP
            item.setHidden(text not in item.text().lower())
```

Η επιλογή Learning Mode ενεργοποιείται μέσω ενός toggle κουμπιού (self.learn_btn) και ελέγχεται από τη μεταβλητή self.learning_mode. Έτσι, όταν η λειτουργία είναι ενεργοποιημένη, κατά την επεξεργασία των δεδομένων των ροών (DataFrame), εκτελείται ένας επαναληπτικός έλεγχος σε κάθε γραμμή μέσω της ανίστοιχης δομής. Για κάθε ροή ανακτώνται οι διευθύνσεις πηγής και προορισμού κι εφόσον αυτές δεν ανήκουν στο σύνολο self.local_ips, προστίθενται αυτόματα στη whitelist (και αφαιρούνται από τη blacklist). Η συλλογή των τοπικών IP γίνεται με την έναρξη της εφαρμογής μέσω της βιβλιοθήκης psutil και του socket.AF_INET στη συνάρτηση που ανακτά τις διεπαφές δικτύου, ενώ, προαιρετικά, περιλαμβάνονται και οι διευθύνσεις 127.0.0.1 και 0.0.0.0 για να αποφεύγεται η εισαγωγή αντίστοιχων τοπικών κόμβων στη λίστα εμπιστοσύνης.

```
# Ενεργοποίηση / Απενεργοποίηση λειτουργίας εκμάθησης
def toggle_learning_mode():
      # Αποθήκευση κατάστασης του toggle button (True/False)
      self.learning_mode = self.learn_btn.isChecked()
      if self.learning_mode: # Αν η λειτουργία είναι ενεργή
            self.learn_btn.setText(
                  # Ενημέρωση του κειμένου του κουμπιού σε ON
                  "Learning Mode: ON (Adding detected remote IPs to
            Whitelist)"
            )
      else: # Αν η λειτουργία είναι ανενεργή
            # Ενημέρωση του κειμένου του κουμπιού σε OFF
            self.learn_btn.setText("Learning Mode: OFF")

# Τελική πρόβλεψη (Απόσπασμα συνάρτησης)
def combined_pred(row):
      # Αν υπάρχει source IP και δεν είναι η τοπική IP του μηχανήματος
      if src and src not in self.local_ips:
```

```
                self.whitelist.add(src) # Προσθήκη της IP στη whitelist
                self.blacklist.discard(src)   # Αφαίρεση της IP απο τη
        blacklist

        # Αν υπάρχει IP παραλήπτη και δεν είναι η τοπική IP του μηχανήματος
        if dst and dst not in self.local_ips:
                self.whitelist.add(dst) # Προσθήκη της IP στη whitelist
                self.blacklist.discard(dst) # Αφαίρεση της IP απο τη blacklist

        self.save_ip_lists() # Αποθήκευση λίστας
        self.refresh_ip_ui_lists() # Ανανέωση των πινάκων UI

# Επιστροφή των τοπικών IP διευθύνσεων του μηχανήματος (Απόσπασμα συνάρτησης)
def get_local_ips(self):
        local_ips = set() # Λίστα για τοπικές IPs
        local_ips.add("127.0.0.1") # Τοπική διεύθυνση
        local_ips.add("0.0.0.0") # Γενική διεύθυνση

        # Αναζήτηση σε τοπικές διεπαφές
        interfaces = psutil.net_if_addrs() # Διευθύνσεις δικτυακών διεπαφών
        # Για κάθε διεπαφή
        for iface_name, iface_addresses in interfaces.items():
                # Για κάθε διεύθυνση της διεπαφής
                for addr in iface_addresses:
                        # Αν η διεύθυνση είναι IPv4 (AF_INET)
                        if addr.family == socket.AF_INET:
                                # Προσθήκη στη λίστα
                                local_ips.add(addr.address)
```

Η λειτουργία του IP Control επηρεάζει άμεσα την τελική απόφαση του συστήματος ανίχνευσης καθώς, πριν από την παρουσίαση των αποτελεσμάτων, εφαρμόζεται ένας μηχανισμός παράκαμψης (override), με βάση τον οποίο δημιουργείται μια συνθήκη που εντοπίζει τις εγγραφές των οποίων η διεύθυνση πηγής ή προορισμού ανήκει στη whitelist. Για τις εγγραφές αυτές που εντοπίστηκαν, το πεδίο prediction τίθεται σε τιμή κανονικής κίνησης (0) και το πεδίο attack_merged σε "Normal", ανεξάρτητα από την έξοδο των μοντέλων μηχανικής μάθησης. Αντίστοιχα, υλοποιείται και ο μηχανισμός blacklist override, όπου εφόσον η διεύθυνση πηγής ή προορισμού ανήκει στη blacklist, η ροή

επισημαίνεται ως κακόβουλη, με τιμή prediction ίση με 1 και ενημέρωση των σχετικών πεδίων. Με αυτόν τον τρόπο διασφαλίζεται ότι η πολιτική του χρήστη δεν επηρεάζεται από την αυτόματη πρόβλεψη (υβριδικό μοντέλο).

```
# Snapshot και επεξεργασία δικτυακών ροών (Απόσπασμα κώδικα)
def on_snapshot(self):
      if self.whitelist: # Αν υπάρχουν IPs στη whitelist
            # Εύρεση ροών που περιέχουν IP (src - dest) της whitelist
            mask = df["srcip"].isin(self.whitelist) |
                  df["dstip"].isin(self.whitelist)
            if mask.any(): # Αν υπάρχουν τέτοιες ροές
                  # Ενημέρωση αντίστοιχης στήλης ως κανονική ροή
                  df.loc[mask, "prediction"] = 0
                  df.loc[mask, "attack_cic"] = "BENIGN"
                  df.loc[mask, "attack_unsw"] = "Normal"
                  df.loc[mask, "attack_merged"] = (
                        "Normal"  # Συνδυαστική ετικέτα
                  )
                  df.loc[mask, "color_tag"] = "green" # Πράσινη γραμμή

      if self.blacklist: # Αν υπάρχουν IPs στη Blacklist
            # Εύρεση ροών που περιέχουν IP (src - dest) της blacklist
            mask = df["srcip"].isin(self.blacklist) |
                  df["dstip"].isin(self.blacklist)
            if mask.any(): # Αν υπάρχουν τέτοιες ροές
                  # Ενημέρωση αντίστοιχης στήλης ως κακόβουλη
                  df.loc[mask, "attack_cic"] = "Malicious"
                  df.loc[mask, "attack_unsw"] = "Malicious"
                  df.loc[mask, "prediction"] = 1 # Malicious
                  df.loc[mask, "attack_merged"] = "Malicious"
                  df.loc[mask, "color_tag"] = "red" # Κόκκινη γραμμή
```

Τέλος, κάθε αλλαγή που γίνεται στις λίστες συγχρονίζεται άμεσα με το UI μέσω της συνάρτησης refresh_ip_ui_lists(), η οποία ενημερώνει τα στοιχεία των QListWidget με τις τρέχουσες εγγραφές των λιστών, ώστε παράλληλα να διασφαλίζεται ότι ο χρήστης έχει άμεση οπτική ανατροφοδότηση των ενεργειών του.

```
# Ανανέωση των whitelist και blacklist πινάκων (UI)
```

```
def refresh_ip_ui_lists(self):
      # Αν υπάρχουν οι λίστες κι έχουν δεδομένα
      if hasattr(self, "wl_list") and hasattr(self, "bl_list"):
            self.wl_list.clear() # Εκκαθάριση whitelist
            self.bl_list.clear() # Εκκαθάριση blacklist
            # Για κάθε IP της whitelist (ταξινομημένες αλφαβητικά)
            for ip in sorted(self.whitelist):
                  # Προσθήκη των IP στον πίνακα
                  self.wl_list.addItem(ip)
            # Για κάθε IP της blacklist (ταξινομημένες αλφαβητικά)
            for ip in sorted(self.blacklist):
                  # Προσθήκη των IP στον πίνακα
                  self.bl_list.addItem(ip)
            # Αν υπάρχει περιεχόμενο στο search input
            if hasattr(self, "search_input"):
                  # Εξαγωγή για να εφαρμοστεί το τρέχον φίλτρο
                  self.search_input.textChanged.emit(self.search_input.t
            ext())
```

9. Κατανομή Εργασιών, Συμπεράσματα και Μελλοντικές Επεκτάσεις

9.1. Κατανομή Εργασιών

Η εκπόνηση της παρούσας διπλωματικής εργασίας υλοποιήθηκε από δύο μέλη, τους φοιτητές κ. Ιωάννη Αλτανόπουλο και Ιωάννη Θεοδωρόπουλο. Η ανάπτυξη του συστήματος στο σύνολό του έγινε με εναλλαγή ρόλων στο backend τμήμα, ενώ το γραφικό περιβάλλον, η σύνδεση όλων των επιμέρους κομματιών και η συνολική εμπειρία χρήστη της κύριας εφαρμογής αναπτύσσονταν συνεχώς και από κοινού σε όλη τη διάρκεια του έργου. Πιο συγκεκριμένα, ο κ. Αλτανόπουλος ανέλαβε το backend τμήμα που αφορά την προεπεξεργασία των δεδομένων, ενώ ο κ. Θεοδωρόπουλος ανέλαβε το αντίστοιχο τμήμα που αφορά την εκπαίδευση του μοντέλου. Στη συνέχεια, όταν το έργο πέρασε στο κομμάτι της λειτουργίας σε πραγματικό χρόνο, οι ρόλοι εναλλάχθηκαν εκ νέου μεταξύ τους. Αυτή η προσέγγιση διασφάλισε ότι κανένα μέλος δεν περιορίστηκε σε ένα μόνο υποσύστημα και ότι αμφότεροι κατανοούσαν πλήρως τη συνολική αρχιτεκτονική του συστήματος. Το χρονοδιάγραμμα υλοποίησης κάλυψε το διάστημα Ιουλίου 2025 - Ιουνίου 2026 και οργανώθηκε σε πέντε φάσεις.

Η πρώτη φάση αφορούσε τη συγκέντρωση βιβλιογραφίας και το σχεδιασμό του έργου και εκτελέστηκε εξ ολοκλήρου από κοινού. Τα δύο μέλη συγκέντρωσαν και μελέτησαν συστηματικά τη σχετική βιβλιογραφία κατά το διάστημα Ιουλίου - Δεκεμβρίου 2025, την αξιολόγησαν και τη διαχώρισαν ανά θεματική κατηγορία, ενώ παράλληλα προχώρησαν στον κοινό σχεδιασμό της αρχιτεκτονικής του συστήματος. Η βιβλιογραφική έρευνα επεκτάθηκε χρονικά έως και τον Δεκέμβριο του 2025 καθώς κρίθηκε αναγκαίος ο συνεχής εμπλουτισμός της θεωρητικής τεκμηρίωσης παράλληλα με τα υπόλοιπα στάδια υλοποίησης.

Στη δεύτερη φάση (Αύγουστος – Σεπτέμβριος 2025) ο ρόλος των δύο μελών διαχωρίστηκε προκειμένου να υλοποιηθεί το backend τμήμα της προεπεξεργασίας και το αρχικό τμήμα της κύριας εφαρμογής. Ο κ. Αλτανόπουλος ανέλαβε το σύνολο του backend τμήματος της εφαρμογής για την προεπεξεργασίας δεδομένων. Η εργασία του περιέλαβε τη φόρτωση των συνόλων δεδομένων, τον αυτόματο εντοπισμό και αφαίρεση των μη χρήσιμων στηλών, καθώς και την αφαίρεση των διπλότυπων εγγραφών που θα μπορούσαν να παραμορφώσουν την εκπαίδευση. Στη συνέχεια υλοποίησε τη διαχείριση ελλειπουσών τιμών βάσει του

ποσοστού των γραμμών καθώς και την κωδικοποίηση κατηγορικών μεταβλητών σε αριθμητική μορφή και την επιλογή των είκοσι σημαντικότερων χαρακτηριστικών βάσει βαθμολόγησης σπουδαιότητας. Ο κ. Θεοδωρόπουλος παρείχε τεχνική υποστήριξη ενώ εργαζόταν παράλληλα στην κύρια εφαρμογή, στήνοντας τη δομή του γραφικού περιβάλλοντος έτσι ώστε τα αποτελέσματα της προεπεξεργασίας να προβάλλονται άμεσα σε πίνακες.

Στη τρίτη φάση (Σεπτέμβριος 2025) υλοποιήθηκε το backend τμήμα εκπαίδευσης του μοντέλου. Ο κ. Θεοδωρόπουλος ανέλαβε το τμήμα που αφορά την προετοιμασία και την εξισορρόπηση των δεδομένων εκπαίδευσης όπου και υλοποίησε τον διαχωρισμό σε σύνολο εκπαίδευσης και σύνολο ελέγχου με αναλογία 70% προς 30%, τις τρεις τεχνικές εξισορρόπησης των κλάσεων (SMOTE, SMOTE-ENN και υβριδική μέθοδος) καθώς επίσης και τα γραφήματα κατανομής κλάσεων πριν και μετά την εξισορρόπηση, ώστε να είναι άμεσα ορατή η επίδραση κάθε μεθόδου στη δομή του συνόλου εκπαίδευσης. Επιπλέον υλοποίησε την εκπαίδευση του ταξινομητή XGBoost και την αυτόματη ανίχνευση δυαδικής ή πολυκλασικής ταξινόμησης ανάλογα με τον αριθμό κλάσεων. Ο κ. Αλτανόπουλος ανέλαβε το τμήμα που αφορά την αξιολόγηση και την επεξηγησιμότητα οπότε και υλοποίησε την παραγωγή του πίνακα σύγχυσης και τον υπολογισμό των μετρικών ορθότητας, ακρίβειας, ανάκλησης και F1-βαθμολογίας, καθώς και την ανάλυση SHAP για την αρχική επαλήθευση των επεξηγήσεων του μοντέλου. Η τελική επαλήθευση της ορθής διασύνδεσης εξισορρόπησης, εκπαίδευσης και επεξηγησιμότητας εκτελέστηκε από κοινού.

Σε μια ενιαία και εκτεταμένη φάση κοινής υλοποίησης, την τέταρτη κατά σειρά (Οκτώβριος 2025 – Φεβρουάριος 2026), πραγματοποιήθηκε η πλήρης ανάπτυξη, διασύνδεση και βελτιστοποίηση της κεντρικής εφαρμογής, με κύριο άξονα την από κοινού συνεργασία και αλληλοϋποστήριξη για τη γεφύρωση του backend και του γραφικού περιβάλλοντος. Στο κομμάτι της διεπαφής χρήστη, σχεδιάστηκε η βασική δομή και τα επίπεδα της κύριας εφαρμογής, τα οποία εξελίχθηκαν δυναμικά με την προσθήκη live panels, πινάκων εμφάνισης ενεργών ροών σε πραγματικό χρόνο και ενός ολοκληρωμένου ταμπλό για τη στατιστική απεικόνιση των δεδομένων, δίνοντας ιδιαίτερη έμφαση στην ευχρηστία και την αισθητική αναβάθμιση της εμπειρίας του χρήστη. Παράλληλα, στο επίπεδο του backend, αναπτύχθηκε ένας ισχυρός μηχανισμός σύλληψης πακέτων και εξαγωγής χαρακτηριστικών από ζωντανή κίνηση δικτύου με δυνατότητα παράλληλης ανίχνευσης πολλαπλών ροών, στον οποίο ενσωματώθηκαν επιτυχώς οι προβλέψεις του εκπαιδευμένου μοντέλου

XGBoost, όπως επίσης και ένας αυτοματοποιημένος μηχανισμός καταγραφής περιστατικών με πλήρη στοιχεία και χρονοσήμανση. Για την ενίσχυση της διαφάνειας του συστήματος, η βιβλιοθήκη SHAP εντάχθηκε στο πλαίσιο διαδικασιών πραγματικού χρόνου, αναλύοντας τη συμβολή κάθε χαρακτηριστικού στις προβλέψεις, με τα αποτελέσματα να συνδέονται με τις δομές καταγραφής και να οπτικοποιούνται στο GUI μέσω εξειδικευμένων διαγραμμάτων. Η φάση αυτή ολοκληρώθηκε με την ενοποίηση όλων των επιμέρους υπό μονάδων σε ένα ενιαίο λειτουργικό περιβάλλον, αποτελούμενο από επτά διαφορετικές καρτέλες, στο οποίο περιβάλλον πραγματοποιήθηκαν εκτεταμένοι συνδυαστικοί έλεγχοι, αποσφαλμάτωση και δοκιμές σταθερότητας υπό πραγματικές συνθήκες. Τέλος, στήθηκε ένα απομονωμένο τοπικό περιβάλλον προσομοίωσης για την εκτέλεση ελεγχόμενων επιθέσεων (μέσω εργαλείων όπως Apache JMeter, LOIC, Nmap και custom Python scripts), επιτρέποντας την κοινή αξιολόγηση, επεξεργασία και τεκμηρίωση της συνολικής απόδοσης και της ταχύτητας ανίχνευσης της εφαρμογής.

Κατά την πέμπτη φάση (Φεβρουάριος – Ιούνιος 2026) πραγματοποιήθηκε η συγγραφή, τεκμηρίωση και παράδοση της εργασίας. Η συγγραφή κατανεμήθηκε ανά θεματική, όπου ο κ. Θεοδωρόπουλος ανέλαβε τα κεφάλαια που αφορούν τη λειτουργία και την αρχιτεκτονική της εφαρμογής, ενώ ο κ. Αλτανόπουλος ανέλαβε τα κεφάλαια που αφορούν τα σύνολα δεδομένων, τη μεθοδολογία μηχανικής μάθησης, την αξιολόγηση και τη συγκριτική ανάλυση. Η εισαγωγή, το θεωρητικό πλαίσιο και η μεθοδολογία συγγράφηκαν παράλληλα και από τα δύο μέλη, ενώ η ενοποίηση των κεφαλαίων, η επιμέλεια και η προετοιμασία της παρουσίασης υλοποιήθηκαν εξ ολοκλήρου από κοινού.

Συνολικά, το έργο εκτάθηκε σε περίπου δώδεκα μήνες (Ιούλιος 2025 – Ιούνιος 2026). Η εναλλασσόμενη κατανομή των backend αρμοδιοτήτων διασφάλισε ότι κανένα μέλος δεν περιορίστηκε σε ένα μόνο υποσύστημα. Η συνεχής, παράλληλη ανάπτυξη της κύριας εφαρμογής από τον διαθέσιμο κάθε φορά συνεργάτη εξασφάλισε ότι το γραφικό περιβάλλον εξελισσόταν αδιάκοπα και ότι αμφότεροι κατανοούσαν πλήρως τη συνολική αρχιτεκτονική. Τα σημεία σύγκλισης - ολοκλήρωση των σημείων τομών, η ενσωμάτωση SHAP, η αποσφαλμάτωση, οι τελικοί έλεγχοι και η συγγραφή υλοποιήθηκαν πάντοτε από κοινού, ενισχύοντας την αλληλοκατανόηση και τη συνοχή του τελικού αποτελέσματος.

9.2. Συμπεράσματα

Η παρούσα διπλωματική εργασία είχε ως αντικείμενο τον σχεδιασμό, την ανάπτυξη και την αξιολόγηση του Shield IDS, ενός ολοκληρωμένου συστήματος ανίχνευσης δικτυακών εισβολών σε πραγματικό χρόνο. Το σύστημα επιτυγχάνει να συνδυάσει με υψηλή απόδοση την ταξινόμηση και ερμηνευσιμότητα των αποφάσεων μέσω της επεξηγήσιμης τεχνητής νοημοσύνης και ένα λειτουργικό γραφικό περιβάλλον, κατάλληλο τόσο για τεχνικούς αναλυτές όσο και για διοικητικά στελέχη. Μέσα από τις επιμέρους φάσεις υλοποίησης του συστήματος, από την προεπεξεργασία έως και τις τελικές δοκιμές, προέκυψαν καίρια συμπεράσματα σχετικά με τη μεθοδολογία που ακολουθήθηκε, καθώς και τη σημασία των αποφάσεων σε κάθε στάδιο της διαδικασίας.

Η έρευνα κατέρριψε την αρχική υπόθεση που θέλει την υψηλή ακρίβεια σε ετερογενή δεδομένα να απαιτεί απαραίτητα πολύπλοκες αρχιτεκτονικές βαθιάς μάθησης. Αντίθετα, αποδείχθηκε ότι η συστηματική βελτιστοποίηση του αλγορίθμου XGBoost μέσω του RandomizedSearchCV, σε συνδυασμό με προσεκτική προεπεξεργασία, είναι επαρκής για την επίτευξη ιδιαίτερα υψηλών επιδόσεων. Η αφαίρεση στηλών που δεν μπορούν να χρησιμοποιηθούν με ασφάλεια, όπως οι διευθύνσεις IP και οι θύρες, σε συνδυασμό με την επιλογή συγκεκριμένων χαρακτηριστικών, βελτίωσαν σημαντικά την ποιότητα της εκπαίδευσης. Παράλληλα, διαπιστώθηκε ότι δεν υφίσταται καθολικά βέλτιστη λύση για τη διαχείριση των ελλειπουσών τιμών, καθώς η επίδρασή τους διαφοροποιείται ανάλογα με το σύνολο δεδομένων και τον τύπο της επίθεσης. Οι βέλτιστοι συνδυασμοί πέτυχαν κορυφαίες μετρικές που κυμάνθηκαν από 99.15% έως 99.97%, ισοσταθμίζοντας ή ξεπερνώντας πολύπλοκα μοντέλα της βιβλιογραφίας με ελάχιστο υπολογιστικό κόστος.

Η αντιμετώπιση των ανισόρροπων δεδομένων αναδείχθηκε σε ένα ιδιαίτερα σύνθετο ζήτημα, καθώς καμία τεχνική εξισορρόπησης δεν επικράτησε ως καθολικά ιδανική. Η υβριδική μέθοδος κρίθηκε πιο αποτελεσματική σε περιπτώσεις με εξαιρετικά σπάνιες κλάσεις, όπως οι επιθέσεις Heartbleed και SQL Injection, ενώ η απλή μέθοδος SMOTE κρίθηκε επαρκής σε περιπτώσεις ήπιων ανισορροπιών. Επισημαίνεται ότι η υπερβολική εξισορρόπηση (ψευδοτυχαία προσθήκη νέων δειγμάτων) μπορεί να υποβαθμίσει την απόδοση σε ήδη καλά εκπροσωπούμενες κλάσεις, ενώ οι τεχνητές δειγματοληψίες δεν είναι ουδέτερες, καθώς επηρεάζουν τη σχετική κατάταξη και τη δομή των χαρακτηριστικών.

Η ενσωμάτωση της ανάλυσης μέσω του SHAP προσέφερε πολύτιμη διαφάνεια στο μοντέλο, αποκαλύπτοντας ότι διαφορετικά σύνολα δεδομένων βασίζονται σε διαφορετικά κυρίαρχα χαρακτηριστικά για την ταξινόμηση της κίνησης, όπως το Init_Win_bytes_backward στο CIC-IDS2017 και το smeansz στο UNSW-NB15. Η σταθερότητα αυτών των χαρακτηριστικών επιβεβαίωσε ότι το μοντέλο βασίζεται σε γνήσια δικτυακά μοτίβα και όχι σε δημιουργήματα της εκπαίδευσης. Παράλληλα, η ανάλυση εξήγησε φυσικά τη σύγχυση μεταξύ των παρόμοιων επιθέσεων, όπως το DoS Slowloris και το DoS Slowhttptest, μεταφέροντας την ευθύνη από τον αλγόριθμο στην ίδια τη φύση της δικτυακής τους συμπεριφοράς, κάτι που προσφέρει ουσιαστική υποστήριξη και στον αναλυτή ασφαλείας.

Οι ελεγχόμενες δοκιμές που πραγματοποιήθηκαν σε τοπικό δίκτυο επιβεβαίωσαν τη σταθερότητα του μοντέλου κατά τη μετάβαση από το εκτός σύνδεσης (offline) περιβάλλον στην πραγματική κυκλοφορία. Η λειτουργία σε πραγματικό χρόνο ανέδειξε σημαντικές προκλήσεις, όπως η εξαγωγή χαρακτηριστικών ροής σε πραγματικό χρόνο από ακατέργαστα πακέτα και η ταυτόχρονη παραγωγή επεξηγήσεων. Το Shield IDS γεφυρώνει με επιτυχία το κενό των παραδοσιακών συστημάτων, καθώς παρέχει στον αναλυτή, μέσω των SHAP ερμηνειών και της αντιστοίχισης με τεχνικές του MITRE ATT&CK, το απαραίτητο πλαίσιο για δράση ενώ ταυτόχρονα, μέσω ενός πίνακα ελέγχου, μεταφράζει τα τεχνικά ευρήματα σε κατανοητή πληροφορία για στελέχη που δεν υποχρεούνται να φέρουν τις αντίστοιχες τεχνικές γνώσεις.

Συνολικά, η εργασία κατέδειξε ότι η XAI δεν αποτελεί μια απλή πρόσθετη λειτουργία, αλλά δομικό συστατικό που ενισχύει την εμπιστοσύνη και την πρακτική αξία ενός συστήματος ανίχνευσης εισβολών. Το τελικό αποτέλεσμα δεν είναι απλώς ένα θεωρητικό μοντέλο με υψηλές μετρικές, αλλά ένα ολοκληρωμένο, ρεαλιστικό και λειτουργικό σύστημα ικανό να ανιχνεύει επιθέσεις, να αιτιολογεί τις αποφάσεις του και να επικοινωνεί αποτελεσματικά τα ευρήματά του σε κάθε επίπεδο ενός οργανισμού, από τον τεχνικό αναλυτή έως τη διοίκηση.

9.3. Μελλοντικές Επεκτάσεις

Παρά τα σημαντικά αποτελέσματα που επιτεύχθηκαν στο πλαίσιο της παρούσας εργασίας, υπάρχουν αρκετές κατευθύνσεις που αξίζει να διερευνηθούν σε μελλοντική έρευνα. Αρχικά, η εκπαίδευση και η αξιολόγηση των μοντέλων βασίστηκε σε δημόσια διαθέσιμα σύνολα δεδομένων (CIC-IDS2017 και UNSW-NB15), τα οποία, παρά την ευρεία

βιβλιογραφική αποδοχή τους, δεν αντικατοπτρίζουν πάντα την κυκλοφορία ενός συγκεκριμένου οργανισμού. Μια ιδιαίτερα ενδιαφέρουσα κατεύθυνση θα ήταν η δημιουργία πρωτότυπου συνόλου δεδομένων, απευθείας από το ίδιο το Shield IDS, αξιοποιώντας τη δυνατότητα σύλληψης και καταγραφής πραγματικής δικτυακής κίνησης που ήδη διαθέτει και στη συνέχεια η χρήση του για επανεκπαίδευση και αξιολόγηση των μοντέλων σε συνθήκες πιο κοντά στην πραγματικότητα. Επιπλέον, η ενσωμάτωση τεχνικών ομοσπονδιακής μάθησης (federated learning) θα επέτρεπε τη συνεργατική εκπαίδευση μοντέλων από πολλαπλούς κόμβους, χωρίς κεντρική ανταλλαγή ευαίσθητων δεδομένων, ενισχύοντας τόσο την ιδιωτικότητα όσο και τη γενίκευση του συστήματος. Μια άλλη σημαντική επέκταση αφορά τη μετατροπή του Shield IDS από σύστημα ανίχνευσης (IDS) σε σύστημα πρόληψης εισβολών (IPS), εξοπλίζοντάς το με τη δυνατότητα αυτόματης λήψης μέτρων αντίδρασης, όπως ο αποκλεισμός κακόβουλων διευθύνσεων IP σε πραγματικό χρόνο. Στο επίπεδο της δικτυακής υποδομής, η τρέχουσα υλοποίηση περιορίζεται σε κίνηση IPv4, γεγονός που αποτελεί περιορισμό ενόψει της ολοένα αυξανόμενης υιοθέτησης του IPv6, η επέκταση της εξαγωγής χαρακτηριστικών ώστε να υποστηρίζει και τα δύο πρωτόκολλα αποτελεί επομένως προτεραιότητα για μελλοντική εργασία. Τέλος, η διερεύνηση τεχνικών αυξητικής ή διαδικτυακής μάθησης (online/incremental learning) θα επέτρεπε στα μοντέλα να προσαρμόζονται δυναμικά σε νέες και εξελισσόμενες κατηγορίες επιθέσεων χωρίς την ανάγκη πλήρους επανεκπαίδευσης, καθιστώντας το σύστημα πιο ανθεκτικό απέναντι στο συνεχώς μεταβαλλόμενο τοπίο των κυβερνοαπειλών.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

[1] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model," *Complexity*, vol. 2021, no. 1, p. 6634811, Jan. 2021, doi: 10.1155/2021/6634811.

[2] G. Ananya, N. Siri, A. Kodipalli, and T. Rao, "Explainable AI for Enhanced Intrusion Detection Systems," in *2024 3rd International Conference for Advancement in Technology (ICONAT)*, GOA, India: IEEE, Sep. 2024, pp. 1–6. doi: 10.1109/ICONAT61936.2024.10774733.

[3] D. K. K, S. G. G, A. S, and S. P. S, "Unraveling Network Attacks with Machine Learning and Explainable AI," in *2024 Second International Conference on Advances in Information Technology (ICAIT)*, Chikkamagaluru, Karnataka, India: IEEE, Jul. 2024, pp. 1–6. doi: 10.1109/ICAIT61638.2024.10690557.

[4] M. K. Nallakaruppan, S. R. K. Somayaji, S. Fuladi, F. Benedetto, S. K. Ulaganathan, and G. Yenduri, "Enhancing Security of Host-Based Intrusion Detection Systems for the Internet of Things," *IEEE Access*, vol. 12, pp. 31788–31797, 2024, doi: 10.1109/ACCESS.2024.3355794.

[5] K. C. Santos, R. S. Miani, and F. De Oliveira Silva, "Evaluating the Impact of Data Preprocessing Techniques on the Performance of Intrusion Detection Systems," *J Netw Syst Manage*, vol. 32, no. 2, p. 36, Apr. 2024, doi: 10.1007/s10922-024-09813-z.

[6] G. Ketepalli and P. Bulla, "Data Preparation and Pre-processing of Intrusion Detection Datasets using Machine Learning," in *2023 International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal: IEEE, Apr. 2023, pp. 257–262. doi: 10.1109/ICICT57646.2023.10134025.

[7] S. G, A. G, M. C, J. R, and P. S. P. C. S, "Explainable AI-Powered Networking Intrusion Detection Systems," in *2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India: IEEE, Feb. 2025, pp. 399–405. doi: 10.1109/IDCIOT64235.2025.10914852.

[8] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia: IEEE, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[9] K. Fatema *et al.*, "Federated XAI IDS: An Explainable and Safeguarding Privacy Approach to Detect Intrusion Combining Federated Learning and SHAP," *Future Internet*, vol. 17, no. 6, p. 234, May 2025, doi: 10.3390/fi17060234.

[10] S. Wali and I. Khan, "Explainable AI and Random Forest Based Reliable Intrusion Detection system," Dec. 18, 2021. doi: 10.36227/techrxiv.17169080.v1.

[11] C. I. Nwakanma *et al.*, "Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review," *Applied Sciences*, vol. 13, no. 3, p. 1252, Jan. 2023, doi: 10.3390/app13031252.

[12] B. Nugraha, A. V. Jnanashree, and T. Bauschert, “An Efficient Explainable Artificial Intelligence (XAI)-Based Framework for a Robust and Explainable IDS,” in *2024 8th Cyber Security in Networking Conference (CSNet)*, Paris, France: IEEE, Dec. 2024, pp. 173–181. doi: 10.1109/CSNet64211.2024.10851760.

[13] T. Tahrim, Md. A. Sharan, M. F. Ahmed, and A. Shakil, “Interpretable DDoS Attack Detection: Combining Machine Learning and SHAP,” in *2024 27th International Conference on Computer and Information Technology (ICCIT)*, Cox’s Bazar, Bangladesh: IEEE, Dec. 2024, pp. 1392–1397. doi: 10.1109/ICCIT64611.2024.11022481.

[14] T. Sood, S. Prakash, S. Sharma, A. Singh, and H. Choubey, “Intrusion Detection System in Wireless Sensor Network Using Conditional Generative Adversarial Network,” *Wireless Pers Commun*, vol. 126, no. 1, pp. 911–931, Sep. 2022, doi: 10.1007/s11277-022-09776-x.

[15] P. Jain, A. Rathour, A. Sharma, and G. S. Chhabra, “Bridging Explainability and Security: An XAI-Enhanced Hybrid Deep Learning Framework for IoT Device Identification and Attack Detection,” *IEEE Access*, vol. 13, pp. 127368–127390, 2025, doi: 10.1109/ACCESS.2025.3590159.

[16] Y. Lee, E. Lee, and T. Lee, “Human-Centered Efficient Explanation on Intrusion Detection Prediction,” *Electronics*, vol. 11, no. 13, p. 2082, Jul. 2022, doi: 10.3390/electronics11132082.

[17] S. Balusamy, R. K. Rengasamy, and A. J, “Employing Big Data and AI to Real-Time Detect Emerging Threats: Improved Machine Learning for Cyberattack Prediction,” in *2025 Global Conference in Emerging Technology (GINOTECH)*, PUNE, India: IEEE, May 2025, pp. 1–6. doi: 10.1109/GINOTECH63460.2025.11076727.

[18] P. M. Corea, Y. Liu, J. Wang, S. Niu, and H. Song, “Explainable AI for Comparative Analysis of Intrusion Detection Models,” in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, Madrid, Spain: IEEE, Jul. 2024, pp. 585–590. doi: 10.1109/MeditCom61057.2024.10621339.

[19] M. Mahmoud, Y. O. Youssef, and A. Abdel-Hamid, “An Explainable Intelligent Intrusion Detection System,” in *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, Washington DC, DC, USA: IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/ISNCC62547.2024.10758968.

[20] C. Hwang and T. Lee, “E-SFD: Explainable Sensor Fault Detection in the ICS Anomaly Detection System,” *IEEE Access*, vol. 9, pp. 140470–140486, 2021, doi: 10.1109/ACCESS.2021.3119573.

[21] Y.-H. Su, M. C. Y. Cho, and H.-C. Huang, “False Alert Buster: an Adaptive Approach for NIDS False Alert Filtering,” in *Proceedings of the 2nd International Conference on Computing and Big Data*, Taichung Taiwan: ACM, Oct. 2019, pp. 58–62. doi: 10.1145/3366650.3366657.

[22] E. Namjoo, A. N. O’Connor, J. Buckley, and C. Ryan, “GlassBoost: A Lightweight and Explainable Classification Framework for Tabular Datasets,” *Applied Sciences*, vol. 15, no. 12, p. 6931, Jun. 2025, doi: 10.3390/app15126931.

[23] R. Mallick, S. Rout, S. Biswas, L. Vashishtha, and S. Kumar Sahu, “Intrusion Detection using Explainable Machine Learning Techniques,” in *2023 OITS International Conference on Information Technology (OCIT)*, Raipur, India: IEEE, Dec. 2023, pp. 564–569. doi: 10.1109/OCIT59427.2023.10430641.

[24] J. V. Rani, H. A. Saeed Ali, and A. Jakka, "IoT Network Intrusion Detection: An Explainable AI Approach in Cybersecurity," in *2023 4th International Conference on Communication, Computing and Industry 6.0 (C216)*, Bangalore, India: IEEE, Dec. 2023, pp. 1–6. doi: 10.1109/C2I659362.2023.10430601.

[25] T. Zebin, S. Rezvy, and Y. Luo, "An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks," *IEEE Trans.Inform.Forensic Secur.*, vol. 17, pp. 2339–2349, 2022, doi: 10.1109/TIFS.2022.3183390.

[26] A. Gupta, A. K. Phulre, A. Patel, and R. U. Rahman, "Hybrid Ensemble Learning with Explainable AI for Anomaly Detection in Network Traffic," in *2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC)*, Davangere, India: IEEE, Oct. 2024, pp. 1–8. doi: 10.1109/ICICEC62498.2024.10808834.

[27] A. Ericson, K. Thar, and S. Forsström, "Enhancing Intrusion Detection in CPS and IIoT with Lightweight Explainable AI Models," in *2025 IEEE 21st International Conference on Factory Communication Systems (WFCS)*, Rostock, Germany: IEEE, Jun. 2025, pp. 1–8. doi: 10.1109/WFCS63373.2025.11077567.

[28] V. Werner de Vargas, J. A. Schneider Aranda, R. Dos Santos Costa, P. R. Da Silva Pereira, and J. L. Victória Barbosa, "Imbalanced data preprocessing techniques for machine learning: a systematic mapping study," *Knowl Inf Syst*, vol. 65, no. 1, pp. 31–57, Jan. 2023, doi: 10.1007/s10115-022-01772-8.

[29] I. U. Khan *et al.*, "A Proactive Attack Detection for Heating, Ventilation, and Air Conditioning (HVAC) System Using Explainable Extreme Gradient Boosting Model (XGBoost)," *Sensors*, vol. 22, no. 23, p. 9235, Nov. 2022, doi: 10.3390/s22239235.

[30] A. Bora, R. Sah, A. Singh, D. Sharma, and R. K. Ranjan, "Interpretation of machine learning models using XAI - A study on health insurance dataset," in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India: IEEE, Oct. 2022, pp. 1–6. doi: 10.1109/ICRITO56286.2022.9964649.

[31] A. F. Çelik, B. Sağlam, and S. Demirci, "Developing Explainable Intrusion Detection Systems for Internet of Things," in *2023 16th International Conference on Information Security and Cryptology (ISCTürkiye)*, Ankara, Turkiye: IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/ISCTrkiye61151.2023.10336102.

[32] F. Sohail, M. A. M. Bhatti, M. Awais, and A. Iqtidar, "Explainable Boosting Ensemble Methods for Intrusion Detection in Internet of Medical Things (IoMT) Applications," in *2024 4th International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, Islamabad, Pakistan: IEEE, Oct. 2024, pp. 1–8. doi: 10.1109/ICoDT262145.2024.10740251.

[33] B. Desai, K. Patil, I. Mehta, and A. Patil, "Explainable AI in Cybersecurity: A Comprehensive Framework for enhancing transparency, trust, and Human-AI Collaboration," in *2024 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia: IEEE, Sep. 2024, pp. 135–150. doi: 10.1109/iSemantic63362.2024.10762690.

[34] M. Rodríguez, Á. Alesanco, L. Mehavilla, and J. García, "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection," *Sensors*, vol. 22, no. 23, p. 9326, Nov. 2022, doi: 10.3390/s22239326.

[35] D. Gladić, J. Petrovački, S. Sladojević, M. Arsenović, and S. Ristić, "Analysis of different IDS-based machine learning models for secure data transmission in IoT networks," *Open Computer Science*, vol. 15, no. 1, p. 20250032, Jul. 2025, doi: 10.1515/comp-2025-0032.

[36] I. Uysal and U. Kose, "Analysis of Network Intrusion Detection via Explainable Artificial Intelligence: Applications with SHAP and LIME," in *2024 Cyber Awareness and Research Symposium (CARS)*, Grand Forks, ND, USA: IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/CARS61786.2024.10778742.

[37] Z. Rastin and D. Söffker, "Advanced POD-Based Performance Evaluation of Classifiers Applied to Human Driver Lane Changing Prediction," 2024, doi: 10.48550/ARXIV.2408.15722.

[38] J. Ribeiro, R. Silva, L. Cardoso, and R. Alves, "Does Dataset Complexity Matters for Model Explainers?," in *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA: IEEE, Dec. 2021, pp. 5257–5265. doi: 10.1109/BigData52589.2021.9671630.

[39] Z. Hu, G. Liu, Y. Li, and S. Zhuang, "SAGB: self-attention with gate and BiGRU network for intrusion detection," *Complex Intell. Syst.*, vol. 10, no. 6, pp. 8467–8479, Dec. 2024, doi: 10.1007/s40747-024-01577-y.

[40] P. Barnard, N. Marchetti, and L. A. DaSilva, "Robust Network Intrusion Detection Through Explainable Artificial Intelligence (XAI)," *IEEE Netw. Lett.*, vol. 4, no. 3, pp. 167–171, Sep. 2022, doi: 10.1109/LNET.2022.3186589.

[41] H. Xuan and M. Manohar, "Intrusion Detection System with Machine Learning and Multiple Datasets," 2023, *arXiv*. doi: 10.48550/ARXIV.2312.01941.

[42] W. Chimphlee and S. Chimphlee, "Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset," *IJ-AI*, vol. 13, no. 1, p. 817, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp817-826.

[43] S. Peng, Y. Han, X. Liang, C. Yang, W. Gui, and N. Zhou, "ROSE-BOX: An Approach for Intrusion Detection in Industrial Internet of Things," in *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, Kaifeng, China: IEEE, Oct. 2024, pp. 2276–2277. doi: 10.1109/ISPA63168.2024.00325.

[44] Z. Asimiyu, "Visualizing Cyber Threats: The Role of Data Analytics in Building Actionable Dashboards." 2024. [Online]. Available: https://www.researchgate.net/publication/385896107_Visualizing_Cyber_Threats_The_Role_of_Data_Analytics_in_Building_Actionable_Dashboards

[45] M. Mccarty *et al.*, "Cybersecurity Resilience Demonstration for Wind Energy Sites in Co-Simulation Environment," *IEEE Access*, vol. 11, pp. 15297–15313, 2023, doi: 10.1109/ACCESS.2023.3244778.

[46] A. Q. Sabri and H. B. M. Dahlan, "Decision-Making Model for Risk Assessment in Cloud Computing Using the Enhanced Hierarchical Holographic Modeling," *Computers*, vol. 14, no. 11, p. 491, Nov. 2025, doi: 10.3390/computers14110491.

[47] A. Bensaoud and J. Kalita, "Optimized detection of cyber-attacks on IoT networks via hybrid deep learning models," 2025, doi: 10.48550/ARXIV.2502.11470.

[48] K. Razikin and B. Soewito, "Cybersecurity decision support model to designing information technology security system based on risk analysis and cybersecurity framework," *Egyptian Informatics Journal*, vol. 23, no. 3, pp. 383–404, Sep. 2022, doi: 10.1016/j.eij.2022.03.001.

[49] G. Rjoub *et al.*, "A Survey on Explainable Artificial Intelligence for Cybersecurity," *IEEE Trans. Netw. Serv. Manage.*, vol. 20, no. 4, pp. 5115–5140, Dec. 2023, doi: 10.1109/TNSM.2023.3282740.

[50] A. Sharma, S. Rani, and M. Shabaz, "A comprehensive review of explainable AI in cybersecurity: Decoding the black box," *ICT Express*, vol. 11, no. 6, pp. 1200–1219, Dec. 2025, doi: 10.1016/j.icte.2025.10.004.

[51] M. J. Khalid, S. M. Mustuzhar Gilani, A. Kabir, and Q. Nawaz, "SDN-based Intrusion Detection and Prevention System Against ARP Spoofing Attacks," *FUJEAS*, vol. 6, no. 1, pp. 1–16, Jul. 2025, doi: 10.33897/fujeas.v6i1.965.

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.