«Σχολή Θετικών Επιστημών»

«Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα »

Πτυχιακή / Διπλωματική Εργασία

Ανίχνευση ανωμαλιών στα αρχεία καταγραφής vSphere με χρήση τεχνικών μηχανικής μάθησης

Καλλούδης Αλέξανδρος

Επιβλέπων καθηγητής: Καραπιπέρης Δημήτριος

Πάτρα, Δεκέμβριος 2025

ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

«Ανίχνευση ανωμαλιών στα αρχεία καταγραφής vSphere με χρήση τεχνικών μηχανικής μάθησης»

Αλέξανδρος Καλλούδης

Επιτροπή Επίβλεψης Πτυχιακής / Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:	Συν-Επιβλέπων Καθηγητής:
ΚΑΡΑΠΙΠΕΡΗΣ ΔΗΜΗΤΡΙΟΣ	ΑΜΑΝΑΤΙΔΗΣ ΔΗΜΗΤΡΙΟΣ

Πάτρα, Δεκέμβριος 2025

Στην Οικογένεια μου Θέμις, Ειρήνη

Περίληψη

Η Μηχανική Μάθηση αποτελεί στις μέρες μας έναν από τους πιο δυναμικά εξελισσόμενους κλάδους που αφορούν την Τεχνητή Νοημοσύνη, καθώς η πρόοδος στους αλγορίθμους της, τα εργαλεία και την υπολογιστική της ισχύ, έχει καταστήσει εφικτή την εφαρμογή της σε ένα ευρύ φάσμα πεδίων, που ξεκινούν από την ιατρική και την αναγνώριση εικόνας και ομιλίας, και φτάνουν μέχρι τα δίκτυα, τις τηλεπικοινωνίες, τα συστήματα αυτόνομης οδήγησης κ.α.

Στην εργασία αυτή εξετάζουμε πώς τεχνικές Μηχανικής Μάθησης μπορούν να εντοπίσουν ανωμαλίες στα αρχεία καταγραφής του VMware vSphere. Η πλατφόρμα vSphere χρησιμοποιείται για τη διαχείριση εικονικών μηχανών και παράγει μεγάλο όγκο logs σε καθημερινή βάση. Αν μέσα σε αυτά τα δεδομένα χαθούν προειδοποιητικά σημάδια, τότε η αξιοπιστία και η απόδοση του συστήματος μπορούν να επηρεαστούν σοβαρά. Στόχος της διπλωματικής είναι να φιλτράρει αυτόν τον όγκο πληροφορίας και να αναδείξει τα γεγονότα που ξεφεύγουν από την τυπική λειτουργία του host.

Για την επίτευξη αυτού του στόχου, θα αξιοποηθεί η γλώσσα προγραμματισμού Python, οι βιβλιοθήκες της και η Μηχανική Μάθηση σε συνδυασμό με κατάλληλες αλγοριθμικές τεχνικές. Η μεθοδολογία περιλαμβάνει τη συλλογή και την προεπεξεργασία των αρχείων καταγραφής με κατάλληλες τεχνικές, την εξαγωγή χαρακτηριστικών από τα μηνύματα των logs, τη μετατροπή τους σε αριθμητική μορφή με χρήση TF-IDF και τη χρήση τριών μοντέλων ανίχνευσης ανωμαλιών: του Isolation Forest, του One-Class SVM (με μείωση διαστάσεων) και ενός νευρωνικού μοντέλου ακολουθιών τύπου GRU.

Η εργασία θα προσπαθήσει να συμβάλει στη συστηματική αξιοποίηση των τεχνικών Μηχανικής Μάθησης για την ανάλυση των log αρχείων σε περιβάλλοντα virtualization, παρέχοντας ένα πλαίσιο που μπορεί να υποστηρίξει τη βελτίωση της διαχείρισης και την πρόληψη πιθανών ανωμαλιών στο vSphere.

Λέξεις – Κλειδιά

Μηχανική Μάθηση, Python, Scikit-learn, Isolation Forest, One-Class SVM, GRU, TF-IDF, Ανίχνευση Ανωμαλιών, Αρχεία Καταγραφής vSphere, Log Templates

Detecting anomalies in Vsphere log files using Machine Learning techniques

Alexandros Kalloudis

Abstract

Machine Learning is currently one of the most dynamically evolving branches of Artificial Intelligence, as the progress in its algorithms, tools and computing power has made it possible to apply it in a wide range of fields, starting from medicine and image and speech recognition, and reaching networks, telecommunications, autonomous driving systems, etc. In this work, we examine how Machine Learning techniques can identify anomalies in VMware vSphere log files. The vSphere platform is used to manage virtual machines and produces a large volume of logs on a daily basis. If warning signs are lost in this data, then the reliability and performance of the system can be seriously affected. The goal of the thesis is to filter this volume of information and highlight events that deviate from the typical operation of the host.

To achieve this goal, the Python programming language, its libraries and Machine Learning will be utilized in combination with appropriate algorithmic techniques. The methodology includes the collection and preprocessing of log files with appropriate techniques, the extraction of features from the log messages, their conversion into numerical form using TF-IDF and the use of three anomaly detection models: Isolation Forest, One-Class SVM (with dimensionality reduction) and a GRU-type neural sequence model.

The work will attempt to contribute to the systematic utilization of Machine Learning techniques for the analysis of log files in virtualization environments, providing a framework that can support the improvement of management and the prevention of possible anomalies in vSphere.

Keywords

Machine Learning, Python, Scikit-learn, Isolation Forest, One-Class SVM, GRU, TF-IDF, Anomaly Detection, vSphere Log Files, Log Templates

Περιεχόμενα

Κατάλογος Εικόνων / Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες & Ακρωνύμια

ML	Machine Learning (Μηχανική Μάθηση)
AI	Artificial Intelligence (Τεχνητή Νοημοσύνη)
VM	Virtual Machine (Εικονική Μηχανή)
vSphere	Πλατφόρμα διαχείρισης εικονικών μηχανών της VMware
SVM	Support Vector Machine
OC-SVM	One-Class Support Vector Machine
IF	Isolation Forest
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Network
DNN	Deep Neural Network
NN	Neural Network
NLP	Natural Language Processing
TF-IDF	Term Frequency–Inverse Document Frequency
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
RF	Random Forest
API	Application Programming Interface
JSON	JavaScript Object Notation
CSV	Comma-Separated Values
LAD	Log-based Anomaly Detection
REGEX	Regular Expressions
IDS	Intrusion Detection System
DDoS	Distributed Denial of Service
EDA	Exploratory Data Analysis
RAM	Random Access Memory
VMkernel	Πυρήνας λειτουργίας του VMware ESXi
ESXi	VMware hypervisor για virtualization servers

1 Εισαγωγή

Η ραγδαία εξάπλωση των τεχνολογιών virtualization έχει μεταβάλει ριζικά τον τρόπο με τον οποίο οργανισμοί σχεδιάζουν, υλοποιούν και διαχειρίζονται τις υποδομές πληροφορικής τους. Πλατφόρμες όπως το VMware vSphere χρησιμοποιούνται ευρέως για τη φιλοξενία κρίσιμων υπηρεσιών, προσφέροντας αυξημένη ευελιξία και καλύτερη αξιοποίηση πόρων [1]. Παρά τα πλεονεκτήματα αυτά, η αυξημένη πολυπλοκότητα των εικονικοποιημένων περιβαλλόντων δημιουργεί σημαντικές προκλήσεις στη διαχείριση, την παρακολούθηση και την έγκαιρη διάγνωση σφαλμάτων, ειδικά όταν τα συμβάντα και οι αστοχίες εκδηλώνονται σε διαφορετικά επίπεδα (host/agent/VM) [2].

Σε τέτοια περιβάλλοντα, τόσο οι hosts όσο και οι εικονικές μηχανές παράγουν συνεχώς αρχεία καταγραφής (log files), τα οποία αποτυπώνουν γεγονότα, προειδοποιήσεις και σφάλματα. Τα logs αποτελούν βασική πηγή πληροφορίας για troubleshooting, καθώς καταγράφουν τη χρονική ακολουθία συμβάντων και διευκολύνουν τη διερεύνηση της συμπεριφοράς του συστήματος [2]. Παράλληλα, η πρακτική αξιοποίησή τους προϋποθέτει γνώση της θέσης και του ρόλου των επιμέρους log αρχείων (π.χ. host logs, VM logs), ώστε να γίνεται στοχευμένη αναζήτηση των κρίσιμων πληροφοριών [3].

Ωστόσο, ο μεγάλος όγκος δεδομένων, η μη δομημένη μορφή των logs και η συνεχής ροή νέων εγγραφών καθιστούν την παραδοσιακή, χειροκίνητη ανάλυσή τους ιδιαίτερα χρονοβόρα και επιρρεπή σε παραλείψεις. Η βιβλιογραφία υπογραμμίζει ότι η αυτόματη ανάλυση logs μπορεί να συμβάλει σε έγκαιρη ανίχνευση συμβάντων και βελτίωση της λειτουργικής απόκρισης, μειώνοντας την εξάρτηση από αποκλειστικά χειροκίνητες διαδικασίες [4].

Στην πράξη, η διαχείριση σφαλμάτων σε virtualized περιβάλλοντα βασίζεται συχνά σε αντιδραστικές διαδικασίες όπου δηλαδή το πρόβλημα εντοπίζεται αφού έχει ήδη εκδηλωθεί και έχει επηρεάσει τη λειτουργία των εικονικών μηχανών ή των παρεχόμενων υπηρεσιών. Για τον λόγο αυτό, η μετάβαση σε προληπτικές προσεγγίσεις ανίχνευσης και πρόβλεψης ανωμαλιών μέσω logs αποτελεί ενεργό και πρακτικά κρίσιμο πεδίο, όπως αναδεικνύεται και από πρόσφατες ερευνητικές ανασκοπήσεις [4].

Τα τελευταία χρόνια, τεχνικές μηχανικής μάθησης και βαθιάς μάθησης έχουν εφαρμοστεί εκτενώς στην ανάλυση δεδομένων καταγραφής, με στόχο τον εντοπισμό ανωμαλιών και προτύπων που σχετίζονται με αστοχίες. Ενδεικτικά, προσεγγίσεις που αντιμετωπίζουν τα

logs ως ακολουθίες γεγονότων μπορούν να μάθουν φυσιολογική συμπεριφορά και να αναγνωρίζουν αποκλίσεις που προηγούνται σφαλμάτων [5]. Παράλληλα, έχει τεκμηριωθεί ότι η προεπεξεργασία των logs (ιδίως το log parsing) επηρεάζει ουσιαστικά την αποτελεσματικότητα των μοντέλων ανίχνευσης ανωμαλιών, κάτι που καθιστά αναγκαία τη συστηματική σχεδίαση του pipeline και την πειραματική αξιολόγηση [6], [7].

Στο πλαίσιο αυτό, η παρούσα εργασία διερευνά τη δυνατότητα αξιοποίησης τεχνικών μηχανικής μάθησης για την ανίχνευση και πρόβλεψη σφαλμάτων σε περιβάλλον VMware vSphere μέσω της ανάλυσης αρχείων καταγραφής. Η μελέτη βασίζεται σε πραγματικά δεδομένα logs και περιλαμβάνει προεπεξεργασία, μετασχηματισμό των logs σε κατάλληλη αναπαράσταση και πειραματική αξιολόγηση επιλεγμένων μεθόδων, με στόχο την ανάδειξη της καταλληλότερης προσέγγισης για το συγκεκριμένο πρόβλημα [4][7].

1.1 Πλαίσιο προβλήματος και γενική προσέγγιση

Η παρούσα εργασία αφορά την ανίχνευση ανωμαλιών σε αρχεία καταγραφής (logs) περιβάλλοντος VMware vSphere. Στόχος είναι να εντοπίζονται καταγραφές ή μοτίβα που αποκλίνουν από τη συνήθη λειτουργία, καθώς τέτοιες αποκλίσεις συχνά αποτελούν προειδοποιητικά σημάδια για σφάλματα, δυσλειτουργίες ή αστάθεια του συστήματος.

Στην πράξη, τα δεδομένα logs σπάνια συνοδεύονται από ετικέτες που να δηλώνουν ποια γραμμή είναι σωστή ή λανθασμένη. Για τον λόγο αυτό, η εργασία δεν βασίζεται σε επιβλεπόμενη ταξινόμηση/παλινδρόμηση, αλλά σε μεθόδους που μπορούν να λειτουργήσουν χωρίς labels:

(α) ανίχνευση ανωμαλιών πάνω σε κατάλληλη αριθμητική αναπαράσταση των logs

(β) μοντελοποίηση της ακολουθίας γεγονότων ώστε να εντοπίζονται απρόσμενες επόμενες καταστάσεις.

Οι αλγόριθμοι που χρησιμοποιούνται είναι οι εξής και αξιολογούνται συγκριτικά:

- Isolation Forest [8]
- One-Class SVM [9]
- μοντέλο ακολουθιών με GRU για next-event prediction, σε λογική αντίστοιχη με sequence-based log anomaly detection [10], [11].

Οι λεπτομέρειες υλοποίησης (παράμετροι, preprocessing pipeline, κατώφλια/thresholds και διαδικασία πειραμάτων) παρουσιάζονται στο Κεφάλαιο 3, ενώ στο Κεφάλαιο

αποτελεσμάτων τεκμηριώνεται ποια προσέγγιση υπερέχει και υπό ποιες συνθήκες. ενδιαφέρον.

Οι παραπάνω αλγόριθμοι επιλέχθηκαν ώστε να καλύπτουν διαφορετικές πτυχές του ίδιου προβλήματος και να επιτρέπουν ουσιαστική σύγκριση μεταξύ στατικών και ακολουθιακών προσεγγίσεων. Η τελική αξιολόγηση και η απάντηση στο ερώτημα ποια μέθοδος υπερέχει και υπό ποιες συνθήκες προκύπτει από τη συγκριτική πειραματική ανάλυση που παρουσιάζεται στα επόμενα κεφάλαια.

1.2 Αναπαράσταση των logs και βασικά στάδια προεπεξεργασίας

Τα logs του vSphere είναι ημι-δομημένα κείμενα που περιέχουν επαναλαμβανόμενα πρότυπα, αλλά και δυναμικά τμήματα (π.χ. αριθμούς, διευθύνσεις, αναγνωριστικά). Για να καταστούν κατάλληλα για αλγορίθμους ανάλυσης, απαιτείται μετασχηματισμός τους σε πιο σταθερή μορφή. Στην εργασία εφαρμόζεται κανονικοποίηση του κειμένου και δημιουργία προτύπων που συμπυκνώνουν το κοινό μέρος των μηνυμάτων, ώστε να μειώνεται ο θόρυβος που προκαλούν οι μεταβαλλόμενες παράμετροι [11].

Με βάση αυτή την προεπεξεργασία, χρησιμοποιούνται δύο συμπληρωματικές αναπαραστάσεις των logs:

- Αναπαράσταση κειμένου με διανυσματοποίηση TF-IDF, ώστε οι εγγραφές να μετατρέπονται σε αριθμητικά διανύσματα που αποτυπώνουν τη σημασία των όρων/μοτίβων στο σύνολο των δεδομένων [12]. Η αναπαράσταση αυτή είναι χρήσιμη όταν θέλουμε να μετρήσουμε διαφορές στο περιεχόμενο των μηνυμάτων σε κλίμακα μεγάλου όγκου.
- Αναπαράσταση γεγονότων/ακολουθιών, όπου κάθε template αντιστοιχεί σε ένα event id και τα logs αντιμετωπίζονται ως χρονολογική ακολουθία γεγονότων. Αυτή η μορφή επιτρέπει τη μοντελοποίηση της ροής (sequence) και την ανίχνευση αποκλίσεων όταν η πραγματική συνέχεια των γεγονότων είναι ασυνήθιστη σε σχέση με το «φυσιολογικό» μοτίβο [10].

Τα παραπάνω αποτελούν το ελάχιστο απαραίτητο υπόβαθρο για να κατανοηθεί τι σημαίνει ανωμαλία στα logs και γιατί η προεπεξεργασία/αναπαράσταση επηρεάζει άμεσα τα αποτελέσματα. Οι ακριβείς κανόνες καθαρισμού, η δημιουργία templates και τα τελικά χαρακτηριστικά παρουσιάζονται αναλυτικά στο Κεφάλαιο 3, ώστε να αποφευχθεί επανάληψη.

1.3 Μεθοδολογία και πειραματικός σχεδιασμός

Η αξιολόγηση οργανώνεται ως συγκριτική μελέτη μεταξύ τριών προσεγγίσεων, Isolation Forest, One-Class SVM και μοντέλου ακολουθιών GRU. Για να είναι δίκαιη η σύγκριση, όλες οι μέθοδοι εφαρμόζονται στα ίδια προεπεξεργασμένα δεδομένα και εξετάζονται ως προς τη δυνατότητά τους να αναδεικνύουν ύποπτες εγγραφές ή μοτίβα.

Για το One-Class SVM, λαμβάνεται υπόψη ότι οι αναπαραστάσεις υψηλής διάστασης (π.χ. μετά από κωδικοποίηση/διανυσματοποίηση) μπορούν να επηρεάσουν τόσο την ταχύτητα όσο και τη σταθερότητα της εκπαίδευσης. Για τον λόγο αυτό, εφαρμόζεται μείωση διαστάσεων πριν το μοντέλο, ώστε η σύγκριση να είναι πρακτικά εφαρμόσιμη και αριθμητικά σταθερή [13]. Αντίστοιχα, το sequence model με GRU αξιολογεί την αναμενόμενη συνέχεια των γεγονότων και σημαίνει απόκλιση όταν η πραγματική εξέλιξη δεν ταιριάζει με τα μοτίβα που έχει μάθει [10], [14].

Τα αποτελέσματα παρουσιάζονται με τρόπο που να απαντά καθαρά στο ερώτημα «ποιος υπερέχει και γιατί», δηλαδή συγκριτικοί πίνακες, παραδείγματα ανωμαλιών που εντοπίζονται, καθώς και ποιοτική ερμηνεία των ευρημάτων σε σχέση με το περιεχόμενο των logs. Οι τελικές επιλογές παραμέτρων, τα κατώφλια ανωμαλίας και η τεκμηρίωση των συμπερασμάτων δίνονται στο κεφάλαιο πειραματικής ανάλυσης, ώστε να παραμένει το παρόν κεφάλαιο συνοπτικό και χωρίς επανάληψη.

1.4 Σχετικές Εργασίες

Η ανάλυση αρχείων καταγραφής αποτελεί καθιερωμένη πρακτική για την παρακολούθηση και τη διάγνωση προβλημάτων σε σύνθετα πληροφοριακά συστήματα. Τα logs καταγράφουν γεγονότα που σχετίζονται με τη λειτουργία εφαρμογών, υπηρεσιών και υποδομών, προσφέροντας κρίσιμες πληροφορίες για τον εντοπισμό και την κατανόηση σφαλμάτων [15]. Σε περιβάλλοντα μεγάλης κλίμακας, όπως data centers, cloud και εικονικοποιημένες υποδομές, ο όγκος, η ποικιλία και η ταχύτητα παραγωγής των logs αυξάνονται σημαντικά, γεγονός που καθιστά τη χειροκίνητη ανάλυση αναποτελεσματική και δύσκολα κλιμακούμενη [16].

Η βιβλιογραφία αναγνωρίζει ότι τα αρχεία καταγραφής είναι κατά κανόνα μη δομημένα, περιέχουν υψηλό επίπεδο θορύβου και παρουσιάζουν έντονη επανάληψη παρόμοιων μηνυμάτων, στοιχεία που δυσχεραίνουν την εξαγωγή χρήσιμης πληροφορίας με απλές

τεχνικές φιλτραρίσματος ή αναζήτησης [17]. Για τον λόγο αυτό, έχει αναπτυχθεί έντονο ερευνητικό ενδιαφέρον για την αυτοματοποίηση της ανάλυσης logs, με στόχο τη μείωση του χρόνου εντοπισμού σφαλμάτων και τη βελτίωση της διαθεσιμότητας των συστημάτων. Βασικό βήμα στις περισσότερες αυτοματοποιημένες προσεγγίσεις αποτελεί η προεπεξεργασία των δεδομένων και ειδικότερα το log parsing, δηλαδή ο μετασχηματισμός των αδόμητων μηνυμάτων καταγραφής σε δομημένες αναπαραστάσεις γεγονότων (log templates). Μελέτες έχουν δείξει ότι η ποιότητα του parsing επηρεάζει άμεσα την απόδοση των μεθόδων ανίχνευσης ανωμαλιών που εφαρμόζονται στη συνέχεια [18]. Ενδεικτικά, ο αλγόριθμος Drain έχει χρησιμοποιηθεί ευρέως στη βιβλιογραφία λόγω της αποδοτικότητάς του και της ικανότητάς του να διαχειρίζεται μεγάλα σύνολα δεδομένων καταγραφής με σταθερό υπολογιστικό κόστος [19].

Στο πεδίο της ανίχνευσης ανωμαλιών, αρχικές προσεγγίσεις βασίστηκαν σε στατιστικά μοντέλα και τεχνικές κλασικής μηχανικής μάθησης, όπως clustering και outlier detection. Ωστόσο, οι μέθοδοι αυτές παρουσιάζουν περιορισμούς, καθώς δεν λαμβάνουν επαρκώς υπόψη τις χρονικές εξαρτήσεις μεταξύ διαδοχικών γεγονότων στα logs [20]. Η εισαγωγή μοντέλων που αντιμετωπίζουν τα logs ως ακολουθίες γεγονότων αποτέλεσε σημαντικό βήμα προόδου. Χαρακτηριστικό παράδειγμα αποτελεί η προσέγγιση DeepLog, η οποία χρησιμοποιεί νευρωνικά δίκτυα τύπου LSTM για να μάθει πρότυπα φυσιολογικής συμπεριφοράς και να εντοπίζει αποκλίσεις που προηγούνται της εμφάνισης σφαλμάτων [21].

Πρόσφατες ανασκοπήσεις της βιβλιογραφίας συνοψίζουν τις εξελίξεις στον τομέα της log-based ανίχνευσης ανωμαλιών και επισημαίνουν βασικές προκλήσεις, όπως η γενίκευση των μοντέλων, η μεταβολή της συμπεριφοράς των συστημάτων με την πάροδο του χρόνου (concept drift) και η ανάγκη αξιολόγησης σε πραγματικά δεδομένα παραγωγής [17], [22]. Ιδιαίτερη έμφαση δίνεται στον ρόλο της προεπεξεργασίας και του συνολικού pipeline ανάλυσης, καθώς έχει αποδειχθεί ότι διαφορετικές επιλογές log parsing και αναπαράστασης των δεδομένων μπορούν να επηρεάσουν σημαντικά την τελική απόδοση των μοντέλων [23].

Παρότι η υπάρχουσα βιβλιογραφία είναι εκτενής, η πλειονότητα των μελετών βασίζεται σε γενικά system logs ή σε συνθετικά benchmark datasets, ενώ περιορισμένος αριθμός εργασιών εξετάζει δεδομένα από πραγματικά εικονικοποιημένα περιβάλλοντα. Τα περιβάλλοντα εικονικοποίησης, όπως αυτά που βασίζονται σε VMware vSphere, χαρακτηρίζονται από αυξημένη πολυπλοκότητα, καθώς τα σφάλματα μπορεί να

προκύπτουν από αλληλεπιδράσεις μεταξύ host, hypervisor και εικονικών μηχανών [24]. Οι επίσημες τεχνικές τεκμηριώσεις περιγράφουν αναλυτικά τη δομή και τη σημασία των logs σε τέτοια περιβάλλοντα, χωρίς όμως να εστιάζουν σε αυτοματοποιημένες μεθόδους ανίχνευσης ή πρόβλεψης σφαλμάτων [25].

Κατά συνέπεια, προκύπτει ερευνητικό κενό ως προς τη συστηματική εφαρμογή και συγκριτική αξιολόγηση τεχνικών μηχανικής μάθησης σε πραγματικά δεδομένα logs από περιβάλλοντα VMware vSphere. Η παρούσα εργασία τοποθετείται σε αυτό το κενό, επιδιώκοντας να γεφυρώσει τη θεωρητική έρευνα με την πρακτική ανάλυση logs σε επιχειρησιακά εικονικοποιημένα περιβάλλοντα.

1.5 Εφαρμογές και Περιορισμοί στην Ανίχνευση Ανωμαλιών

Η χρήση supervised learning στην ανίχνευση ανωμαλιών είναι μια περιοχή με ιδιαίτερο ενδιαφέρον. Η βασική ιδέα είναι ότι εάν ένα μοντέλο έχει εκπαιδευτεί επαρκώς με ιστορικά δεδομένα όπου κάθε δείγμα έχει χαρακτηριστεί ως κανονικό ή ανώμαλο, τότε μπορεί να προβλέπει με μεγάλη ακρίβεια την κατηγορία νέων δεδομένων ή εισόδων. Αυτό ισχύει ιδιαίτερα στην κυβερνοασφάλεια, όπου η ταξινόμηση πακέτων δικτύου σε μη-κακόβουλα και κακόβουλα έχει επιδείξει πολύ υψηλά ποσοστά επιτυχίας [26]. Παρόμοια, στην ανάλυση συστημάτων cloud, supervised learning τεχνικές έχουν χρησιμοποιηθεί για την πρόβλεψη αποτυχιών κάνοντας χρήση αρχείων log [27].

Υπάρχουν αρκετά τέτοια παραδείγματα εφαρμογών. Για παράδειγμα, διάφορα μοντέλα ταξινόμησης έχουν αξιοποιηθεί για τον εντοπισμό μοτίβων αποτυχιών σε κατανεμημένα συστήματα, με σκοπό την έγκαιρη ανίχνευση σφαλμάτων πριν αυτά οδηγήσουν σε διακοπές λειτουργίας ή διάφορα άλλα προβλήματα.

Επίσης, η επιβλεπόμενη μάθηση έχει βρει εφαρμογή και στην ανάλυση καταγραφών συστήματος, όπου με τη χρήση labeled δεδομένων επιτυγχάνεται υψηλή ακρίβεια στην αναγνώριση ανώμαλων γεγονότων.

Παρά τα πλεονεκτήματά του, το supervised learning παρουσιάζει σημαντικούς περιορισμούς όταν εφαρμόζεται στην ανάλυση των αρχείων καταγραφής. Ο μεγαλύτερος περιορισμός είναι η ανάγκη ύπαρξης labels. Στην πραγματικότητα, τα logs παράγουν τεράστιες ποσότητες δεδομένων, τα οποία συχνά είναι μη δομημένα και δεν έχουν ομοιογένεια. Το χειροκίνητο labeling απαιτεί εξειδικευμένη γνώση και πολύ χρόνο, ενώ οι

μορφές των ανωμαλιών μπορεί να είναι σπάνιες και διαφορετικές από περίπτωση σε περίπτωση. Αυτό οδηγεί στο πρόβλημα όπου οι κανονικές εγγραφές υπερτερούν αριθμητικά έναντι των ανώμαλων, κάνοντας πολύ πιο δύσκολη την εκπαίδευση αξιόπιστων supervised μοντέλων [28].

Επιπλέον, οι supervised μέθοδοι έχουν περιορισμένη ικανότητα στο να εντοπίζουν νέους, άγνωστους τύπους ανωμαλιών. Επειδή το μοντέλο εκπαιδεύεται αποκλειστικά πάνω σε ιστορικά δεδομένα τα οποία είναι labeled, οποιαδήποτε νέα μορφή σφάλματος που δεν έχει καταγραφεί στο training set μπορεί να αγνοηθεί ή να μην αναγνωριστεί ορθά. Σε δυναμικά περιβάλλοντα, όπως αυτά που βασίζονται σε VMware vSphere, όπου οι συνθήκες αλλάζουν διαρκώς και προκύπτουν απρόβλεπτες καταστάσεις, αυτό μπορεί να αποτελέσει ένα σοβαρό μειονέκτημα.

Η επιβλεπόμενη μάθηση έχει αποδείξει την αξία του σε πολλές εφαρμογές της μηχανικής μάθησης και έχει συμβάλει καθοριστικά στην πρόοδο του συγκεκριμένου πεδίου. Πάρο όλα αυτά, για την ανάλυση των αρχείων καταγραφής και την ανίχνευση ανωμαλιών σε πραγματικά περιβάλλοντα, οι περιορισμοί καθιστούν απαραίτητη την αναζήτηση εναλλακτικών μεθόδων, όπως είναι οι unsupervised learning τεχνικές, οι οποίες δεν εξαρτώνται από την ύπαρξη labels και μπορούν να εντοπίσουν αποκλίσεις χωρίς προκαθορισμένη γνώση και τις οποίες θα αναλύσουμε στο επόμενο κεφάλαιο.

1.6 Μη-Επιβλεπόμενη μάθηση (Uns*upervised Learning)*

Η μη επιβλεπόμενη μάθηση (unsupervised learning) είναι και αυτή μία από τις βασικές κατηγορίες της μηχανικής μάθησης, μαζί με τη επιβλεπόμενη μάθηση που αναλύσαμε στο πιο πάνω κεφάλαιο. Σε αντίθεση με τη επιβλεπόμενη, όπου κάθε δείγμα συνοδεύεται από μία ετικέτα που καθορίζει την επιθυμητή έξοδο, στην μη-επιβλεπόμενη μάθηση τα δεδομένα δεν διαθέτουν labels. Σκοπός είναι το σύστημα να ανακαλύψει από μόνο του τη δομή, τις συσχετίσεις και τις ιδιότητες των δεδομένων που υπάρχουν [29].

Η απουσία των labels καθιστά την μη-επιβλεπόμενη μάθηση κατάλληλη για προβλήματα όπου τα δεδομένα είναι άφθονα αλλά η κατηγοριοποίηση τους είναι δύσκολη, ανέφικτη ή και χρονοβόρα. Αυτό ισχύει σε τομείς όπως η ανάλυση του κειμένου, η ανίχνευση απάτης, η βιοπληροφορική και φυσικά η ανάλυση αρχείων καταγραφής. Σε αυτές τις περιπτώσεις,

η μη επιβλεπόμενη μάθηση επιτρέπει την εύρεση μοτίβων χωρίς να απαιτείται προκαθορισμένη γνώση για το τι θεωρείται σωστό ή λάθος.

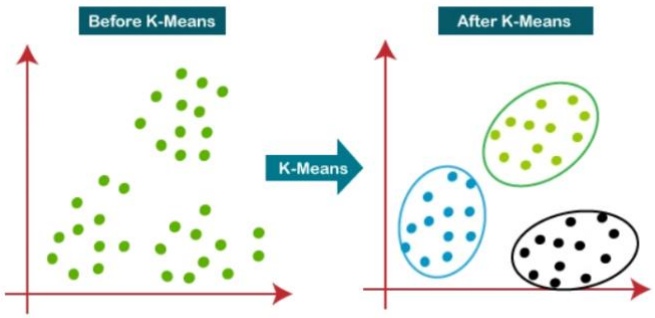
1.6.1 Clustering

Η ομαδοποίηση (clustering) είναι μία από τις πιο θεμελιώδεις τεχνικές της μη-επιβλεπόμενης μάθησης. Στόχος της είναι να χωρίσει τα δεδομένα σε διάφορες ομάδες (clusters) με τέτοιο τρόπο ώστε τα αντικείμενα που ανήκουν στην ίδια ομάδα να παρουσιάζουν μεταξύ τους μεγαλύτερη ομοιότητα σε σχέση με αντικείμενα άλλων ομάδων (άρα να ομαδοποιηθούν). Η μέτρηση της ομοιότητας συνήθως βασίζεται σε αποστάσεις, όπως για παράδειγμα η ευκλείδεια απόσταση, αν και σε πιο σύνθετα προβλήματα μπορεί να χρησιμοποιηθούν πολύ πιο εξειδικευμένες συναρτήσεις ομοιότητας [30]

Η ομαδοποίηση χρησιμοποιείται ευρέως σε εφαρμογές όπως είναι η εξόρυξη κειμένου (text mining), η τμηματοποίηση των πελατών (customer segmentation), η ανάλυση κοινωνικών δικτύων και, πιο πρόσφατα, η ανίχνευση ανωμαλιών σε δίκτυα και αρχεία καταγραφής. Μέσω της διαδικασίας αυτής είναι δυνατό να αποκαλυφθούν κρυφές δομές στα δεδομένα που δεν ήταν από πριν γνωστές.

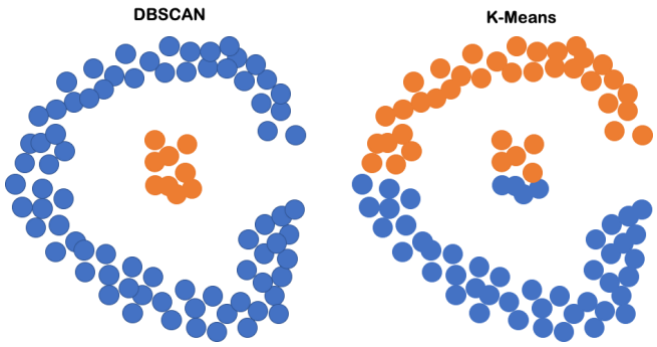
Κύριοι αλγόριθμοι του clustering:

- **k-means**: Πρόκειται για έναν από τους πιο διαδεδομένους αλγόριθμους ομαδοποίησης. Ορίζει εκ των προτέρων τον αριθμό k των ομάδων και κατανέμει τα δεδομένα σε clusters με στόχο την ελαχιστοποίηση της απόστασης από τα κέντρα τους. Η απλότητά του τον καθιστά κατάλληλο για μεγάλα datasets, αλλά είναι ευαίσθητος στην επιλογή του k και στην παρουσία outliers.



Σχήμα 1.1: Ομαδοποίηση με τον αλγόριθμο K-Means. Αριστερά εμφανίζονται τα δεδομένα πριν την εφαρμογή του αλγορίθμου και δεξιά τα ίδια δεδομένα χωρισμένα σε τρεις ομάδες (clusters).[31]

- **DBSCAN**: Βασίζεται στην ιδέα της πυκνότητας. Ανιχνεύει περιοχές υψηλής πυκνότητας σημείων, δημιουργώντας clusters αυθαίρετου σχήματος, ενώ τα σημεία χαμηλής πυκνότητας θεωρούνται outliers. Αυτό τον καθιστά ιδιαίτερα χρήσιμο στην ανίχνευση ανωμαλιών.



Σχήμα 1.2: Σύγκριση DBSCAN (αριστερά) και K-Means (δεξιά).[32]

- **Ιεραρχική Ομαδοποίηση**: Δημιουργεί μια ιεραρχία ομάδων είτε συγχωνεύοντας μικρότερα clusters είτε διαχωρίζοντας μεγαλύτερα. Το αποτέλεσμα συχνά απεικονίζεται με δενδρογράμματα, τα οποία επιτρέπουν την κατανόηση της εσωτερικής δομής των δεδομένων.

1.6.2 Μείωση Διαστάσεων (Dimensionality Reduction)

Η μείωση διαστάσεων αποτελεί μια από τις πιο σημαντικές τεχνικές της μη-επιβλεπόμενης μάθησης. Σαν κύριο στόχο έχει να συμπιέζει τα δεδομένα σε μικρότερο αριθμό χαρακτηριστικών (features), προσπαθώντας να διατηρήσει όσο το δυνατόν περισσότερη πληροφορία γίνεται. Στη σύγχρονη ανάλυση δεδομένων, πολλά σύνολα δεδομένων περιέχουν εκατοντάδες ή και χιλιάδες χαρακτηριστικά, γεγονός που δημιουργεί προβλήματα αποδοτικότητας αλλά και ακρίβειας με αποτέλεσμα να οι αλγόριθμοι να τρέχουν πιο αργά και να μην είναι τόσο ακριβείς. Η μείωση διαστάσεων προσπαθεί να αντιμετωπίσει το παραπάνω πρόβλημα, παρέχοντας μια πιο απλή και κατανοητή αναπαράσταση των δεδομένων.

Η πιο κλασική μέθοδος μείωσης διαστάσεων είναι η Principal Component Analysis (PCA) [13], η οποία δημιουργεί νέες μεταβλητές που εξηγούν το μεγαλύτερο μέρος της διακύμανσης στα δεδομένα. Με τον τρόπο αυτό επιτυγχάνεται μια πιο συμπαγής αλλά ταυτόχρονα περιεκτική αναπαράσταση. Εκτός από το PCA, ιδιαίτερα διαδεδομένες είναι

και πιο σύγχρονες μη γραμμικές μέθοδοι όπως το t-SNE και το UMAP. Το t-SNE χρησιμοποιείται κυρίως για την οπτικοποίηση δεδομένων υψηλής διάστασης σε δύο ή τρεις διαστάσεις, ενώ το UMAP επιτυγχάνει παρόμοιο αποτέλεσμα με μικρότερο υπολογιστικό κόστος και καλύτερη διατήρηση της εσωτερικής δομής.

Η μείωση διαστάσεων έχει αποδειχθεί χρήσιμη σε πολλές εφαρμογές, όπως στη βιοπληροφορική, στην επεξεργασία εικόνας και στην ανάλυση των logs [33]. Στο πλαίσιο της ανίχνευσης ανωμαλιών, η χρήση της επιτρέπει τον καθαρισμό των δεδομένων, διευκολύνοντας έτσι την ανίχνευση αποκλίσεων. Παράλληλα, προσφέρει τη δυνατότητα καλύτερης κατανόησης των δεδομένων μέσα από τη γραφική τους αναπαράσταση, επιτρέποντας τον εντοπισμό μοτίβων που διαφορετικά θα έμεναν κρυμμένα.

Παρά τα πλεονεκτήματα που έχει, η μείωση διαστάσεων συνοδεύεται και από ορισμένους περιορισμούς. Η ερμηνεία των νέων μεταβλητών δεν είναι εύκολη, καθώς συχνά αποτελούν αφηρημένους συνδυασμούς των αρχικών χαρακτηριστικών του dataset. Επίσης, υπάρχει ο κίνδυνος να χαθεί σημαντική πληροφορία κατά τη συμπίεση, κάτι που μπορεί να μειώσει την απόδοση σε κάποιες περιπτώσεις. Παρόλα αυτά, παραμένει ένα από τα πιο αποτελεσματικά εργαλεία για την ανάλυση δεδομένων χωρίς labels και χρησιμοποιείται πάρα πολύ συχνά.

1.6.3 Αλγόριθμοι για Ανίχνευση Ανωμαλιών

Η ανίχνευση ανωμαλιών αποτελεί έναν σημαντικό τομέα εφαρμογής της μη-επιβλεπόμενης μάθησης και αποτελεί το κύριο θέμα της διπλωματικής αυτής εργασίας. Στόχος είναι να εντοπιστούν σημεία ή γεγονότα που αποκλίνουν έντονα από τη κανονική συμπεριφορά του συστήματος, κάτι που μπορεί να υποδεικνύει σφάλματα, κακόβουλες ενέργειες ή απρόσμενες αλλαγές στη λειτουργία, τις οποίες ο ανθρώπινος παράγοντας δεν θα μπορούσε να βρει με ευκολία αν ανέλυε τα logs. Η πρόκληση έγκειται στο ότι σε πραγματικά δεδομένα, και ιδιαίτερα σε αρχεία καταγραφής, οι ανωμαλίες είναι σπάνιες, ανομοιογενείς και συχνά δεν έχουν σαφή ορισμό. Για τον λόγο αυτό, η ανάγκη για αλγορίθμους που δεν απαιτούν labels είναι ιδιαίτερα επιτακτική.

Μία από τις πιο γνωστές προσεγγίσεις είναι η μοντελοποίηση της κανονικότητας (normality modeling). Σε αυτήν την κατηγορία ανήκει ο One-Class SVM, ο οποίος εκπαιδεύεται μόνο με κανονικά δεδομένα και στη συνέχεια θεωρεί ανώμαλα όσα δείγματα βρίσκονται εκτός

του ορίου που έχει μάθει να ορίζει [34]. Η μέθοδος αυτή έχει χρησιμοποιηθεί εκτενώς σε δίκτυα και σε logs για να διαχωρίσει την καθημερινή λειτουργία από ασυνήθιστα γεγονότα. Σημαντικό ρόλο έχουν και οι νευρωνικές προσεγγίσεις με μοντέλα ακολουθιών, όπως τα δίκτυα τύπου GRU (Gated Recurrent Unit). Τα δίκτυα αυτά λαμβάνουν ως είσοδο ακολουθίες γεγονότων από τα αρχεία καταγραφής και μαθαίνουν τις συνηθισμένες μεταβάσεις μεταξύ τους. Για κάθε νέο βήμα στην ακολουθία, το μοντέλο προβλέπει το επόμενο αναμενόμενο γεγονός ή την κατανομή πιθανοτήτων πάνω στα επόμενα γεγονότα. Όταν η πραγματική εξέλιξη αποκλίνει έντονα από τα μοτίβα που έχει μάθει το GRU, η αντίστοιχη ακολουθία χαρακτηρίζεται ως ανώμαλη. Η προσέγγιση αυτή είναι ιδιαίτερα κατάλληλη για logs συστημάτων cloud, όπου η χρονική σειρά και το πλαίσιο των γεγονότων παίζουν κρίσιμο ρόλο στην κατανόηση της συμπεριφοράς του συστήματος.

Ένας ακόμη διαδεδομένος αλγόριθμος είναι το Isolation Forest, το οποίο χρησιμοποιεί τυχαία δέντρα απόφασης για να απομονώσει σημεία του dataset [35]. Οι ανωμαλίες, όντας σπάνιες και διαφορετικές, τείνουν να απομονώνονται πιο εύκολα και με λιγότερα βήματα σε σχέση με τα κανονικά δεδομένα. Η μέθοδος αυτή είναι ιδιαίτερα αποδοτική σε μεγάλα σύνολα δεδομένων και έχει εφαρμοστεί σε πεδία όπως η κυβερνοασφάλεια και η χρηματοοικονομική ανάλυση.

Πέρα από αυτές τις βασικές τεχνικές, υπάρχουν και άλλες μέθοδοι που βασίζονται σε στατιστικά μοντέλα, σε αποστάσεις μεταξύ σημείων ή σε clustering. Κάθε κατηγορία έχει πλεονεκτήματα και περιορισμούς, ενώ η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από τα χαρακτηριστικά των δεδομένων, την κλίμακα του προβλήματος και το επιθυμητό επίπεδο ακρίβειας.

Οι αλγόριθμοι μη-επιβλεπόμενης ανίχνευσης ανωμαλιών αποτελούν ισχυρά εργαλεία για τον έγκαιρο εντοπισμό προβλημάτων σε πολύπλοκα περιβάλλοντα. Ειδικά στα αρχεία καταγραφής, όπου η χειροκίνητη επισήμανση είναι σχεδόν αδύνατη, τέτοιες μέθοδοι παρέχουν μια αξιόπιστη βάση για αυτοματοποιημένη παρακολούθηση. Οι συγκεκριμένες τεχνικές που θα χρησιμοποιηθούν και θα αξιολογηθούν στο πλαίσιο της παρούσας μελέτης θα παρουσιαστούν αναλυτικά σε μεταγενέστερο κεφάλαιο, όπου θα εξεταστούν τα αποτελέσματά τους σε πραγματικά δεδομένα.

1.6.4 Εφαρμογές στην Ανίχνευση Ανωμαλιών

Η μη-επιβλεπόμενη μάθηση έχει βρει εκτεταμένη εφαρμογή στην ανίχνευση ανωμαλιών, καθώς δίνει τη δυνατότητα να εντοπιστούν αποκλίσεις χωρίς να απαιτείται η ύπαρξη ετικετών. Αυτό είναι ιδιαίτερα κρίσιμο σε σενάρια στον πραγματικό κόσμου, όπου τα δεδομένα είναι τεράστια, ανομοιογενή και δύσκολο να επισημανθούν χειροκίνητα.

Ένα από τα πιο χαρακτηριστικά πεδία εφαρμογής είναι η κυβερνοασφάλεια. Σε δίκτυα υπολογιστών, αλγόριθμοι όπως το DBSCAN και το One-Class SVM έχουν χρησιμοποιηθεί για την αναγνώριση επιθέσεων τύπου DDoS, την ανίχνευση εισβολών και τον εντοπισμό κακόβουλης δραστηριότητας. Οι μέθοδοι αυτές θεωρούν ύποπτες τις αποκλίσεις από το κανονικό μοτίβο κυκλοφορίας δικτύου και προσφέρουν τη δυνατότητα έγκαιρης αντίδρασης [36]

Σημαντικές εφαρμογές υπάρχουν και στις υποδομές cloud, όπου η αξιοπιστία των υπηρεσιών είναι κρίσιμη. Μοντέλα όπως το Isolation Forest και νευρωνικά δίκτυα ακολουθιών τύπου GRU έχουν αξιοποιηθεί για την ανάλυση καταγραφών cloud συστημάτων, με στόχο την αναγνώριση σφαλμάτων, την πρόληψη αστοχιών και την παρακολούθηση της απόδοσης. Σε τέτοια περιβάλλοντα, όπου ο όγκος των logs είναι πολύ μεγάλος, η αυτοματοποιημένη ανίχνευση ανωμαλιών αποτελεί βασικό εργαλείο για τη διασφάλιση της συνεχούς λειτουργίας.

Η μη-επιβλεπόμενη μάθηση έχει επίσης εφαρμοστεί άμεσα στην ανάλυση αρχείων καταγραφής (log analysis). Μέθοδοι clustering, όπως το k-means, έχουν χρησιμοποιηθεί για την αυτόματη ομαδοποίηση καταγραφών, διευκολύνοντας τον εντοπισμό ασυνήθιστων γεγονότων που αποκλίνουν από το συνηθισμένο μοτίβο λειτουργίας. Σε μεγάλες υποδομές, η προσέγγιση αυτή συμβάλλει σημαντικά στη μείωση του χρόνου εντοπισμού σφαλμάτων και στη βελτίωση της αξιοπιστίας [37].

Οι εφαρμογές της μη-επιβλεπόμενης μάθησης καλύπτουν ένα ευρύ φάσμα, από την ασφάλεια δικτύων έως τις cloud υπηρεσίες και την ανάλυση logs. Η κοινή συνισταμένη είναι η δυνατότητα εντοπισμού σπάνιων και απρόβλεπτων γεγονότων χωρίς την ανάγκη προκαθορισμένων ετικετών, κάτι που την καθιστά ιδανική για σύγχρονα και δυναμικά υπολογιστικά περιβάλλοντα.

2 vSphere και logs

2.1 Εισαγωγή στο VMware vSphere

Το vSphere αποτελεί το βασικό σύστημα εικονικοποίησης της VMware, Inc.. Εγκαθίσταται σε φυσικούς hosts μέσω του hypervisor ESXi, επιτρέπει τη δημιουργία εικονικών μηχανών δηλαδή Virtual Machines και τη συγκεντρωτική διαχείριση μέσω του vCenter Server. Αυτό το περιβάλλον επιτρέπει την αποδοτική χρήση υλικού, ευελιξία στη διαχείριση πόρων και αυξημένη διαθεσιμότητα. Στο πλαίσιο της διπλωματικής εργασίας, η πλατφόρμα αυτή λειτουργεί ως η πηγή των δεδομένων, δηλαδή των αρχείων καταγραφής που θα αναλύσουμε.

Τα logs παράγονται συνεχώς όσο λειτουργούν οι hosts και τα VMs, και στο σύστημα vSphere έχουν πολύ σημαντικό ρόλο. Καταγράφουν κάθε αλλαγή που συμβαίνει στον πόρο, κάθε πιθανό σφάλμα καθώς και κάθε διακοπή ή ανανέωση της κατάστασης του συστήματος. Για παράδειγμα, ένας host που ξεκινά από την κατάσταση λειτουργίας σε standby καταγράφει τα σχετικά γεγονότα, ένα VM που αλλάζει κατάσταση ή μετακινείται (vMotion) επίσης αφήνει αποτύπωμα μέσα σε log αρχείο.

2.2 Τα αρχεία καταγραφής στο vSphere και η σημασία τους

Οι αρχές μιας πλατφόρμας εικονικοποίησης όπως είναι η VMware vSphere είναι οι εξής. Πολλαπλοί hosts, εικονικές μηχανές, δυναμική μετακίνηση πόρων και συνεχής λειτουργία χωρίς καμία διακοπή. Κάθε τέτοια διεργασία παράγει δεδομένα και τα logs είναι η γραπτή καταγραφή αυτών των διεργασιών, των σφαλμάτων, των προειδοποιήσεων και των αλλαγών κατάστασης. Για παράδειγμα, όταν ένα VM κάνει vMotion από host A σε host B, ή όταν ο host αντιμετωπίζει υπερφόρτωση I/O, ή όταν αλλάζει πολιτική πόρων, τις περισσότερες φορές, υπάρχει εγγραφή σε ένα τέτοιο αρχείο καταγραφής.

Τα logs στο vSphere είναι κρίσιμα για τρεις βασικούς λόγους:

- **Διαγνωστική χρήση**: Όταν εμφανιστεί πρόβλημα όπως για παράδειγμα υποβάθμιση επιδόσεων, αποτυχία VM, διακοπή δικτύου, τα logs παρέχουν τη χρονική ακολουθία των γεγονότων που οδήγησαν στο πρόβλημα. Χωρίς logs, η αναδρομική ανάλυση είναι εξαιρετικά δύσκολη.
- **Ασφάλεια & συμμόρφωση**: Εξωτερικές εισβολές, μη εξουσιοδοτημένες αλλαγές στα VM, προβλήματα δικτύου που μπορεί να οφείλονται σε κακόβουλη δράση. Όλα αφήνουν αποτύπωμα στα logs. Για παράδειγμα, μια αλλαγή στη διαμόρφωση

storage που δεν τεκμηριώθηκε μπορεί να εντοπιστεί από ασυνήθιστη εγγραφή στο vmkernel.

- **Προγνωστική και ανίχνευση ανωμαλιών**: Όχι απλώς η καταγραφή σφαλμάτων, αλλά και η συνεχής παρακολούθηση των logs επιτρέπει την ανίχνευση μικρών αποκλίσεων πριν αυτή η κατάσταση εξελιχθεί σε πλήρες σφάλμα. Οι μελέτες δείχνουν ότι το πρόβλημα της ανίχνευσης ανωμαλιών από logs το οποίο είναι γνωστό ως Log-based Anomaly Detection (LAD), έχει λάβει μεγάλη έμφαση, γιατί τα παραδοσιακά συστήματα κανόνων/στατιστικών δεν επαρκούν σε μεγάλες, δυναμικές υποδομές. [38]

Τα logs στο vSphere εμφανίζουν τις εξής προκλήσεις:

- **Είναι semi-structured**: Τα αρχεία περιέχουν σταθερά σημεία αναφοράς (π.χ. "vmx-operation started"), αλλά και παράμετρους που αλλάζουν (όπως VM ID, host ID, χρονική σήμανση). Για να γίνουν χρήσιμα σε ML, χρειάζεται parsing και εξαγωγή χαρακτηριστικών.
- **Ο όγκος τους μπορεί να είναι τεράστιος**: μεγάλες εγκαταστάσεις vSphere παράγουν gigabytes logs ημερησίως. Αυτό δημιουργεί πρόκληση αποθήκευσης και προ-επεξεργασίας.
- **Οι μορφές τους εξελίσσονται**: με κάθε αναβάθμιση vSphere ή κάθε αλλαγή υποδομής, μπορούν να γίνουν προσθήκες σε μορφές logs ή νέα πεδία πράγμα το οποίο σημαίνει ότι ένα μοντέλο ανίχνευσης πρέπει να είναι ευέλικτο. Για παράδειγμα, σε ανασκόπηση μεθόδων LAD οι συγγραφείς αναφέρουν ότι «η ανίχνευση ανωμαλιών από logs έχει αντιμετωπιστεί κυρίως με τεχνικές deep learning, αλλά υπάρχει μεγάλη ποικιλία δεδομένων και μορφών, γεγονός που δυσκολεύει την γενίκευση».

Η ουσιαστική αξία των αρχείων καταγραφής στο vSphere βρίσκεται στο ότι αποτελούν τη βάση για κατανόηση, πρόβλεψη και αυτοματοποίηση. Κάθε αρχείο καταγραφής είναι ένα κομμάτι της ιστορίας λειτουργίας του συστήματος και, όταν συνδυαστούν, σχηματίζουν ένα δυναμικό χρονικό που μπορεί να αποκαλύψει πρότυπα κανονικότητας αλλά και σημάδια επικείμενων δυσλειτουργιών. Η επεξεργασία τους δεν είναι απλώς τεχνική διαδικασία, αλλά προϋπόθεση για την έξυπνη διαχείριση εικονικών υποδομών. Στην επόμενη ενότητα θα παρουσιαστούν οι βασικές κατηγορίες των log αρχείων που παράγει το vSphere, ώστε

να γίνει κατανοητό πώς αυτά εντάσσονται στη μεθοδολογία ανίχνευσης ανωμαλιών που ακολουθεί η εργασία.

2.3 Κατηγορίες αρχείων καταγραφής στο vSphere

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, η πλατφόρμα VMware vSphere παράγει ένα ευρύ φάσμα αρχείων καταγραφής που καταγράφουν κάθε επίπεδο λειτουργίας του συστήματος, από τον πυρήνα του ESXi μέχρι τις ενέργειες διαχείρισης των χρηστών μέσω του vCenter. Τα logs αποτελούν την πιο λεπτομερή πηγή πληροφορίας για τη συμπεριφορά των hosts και των VMs και λειτουργούν ως θεμέλιο για κάθε μελέτη ανίχνευσης ανωμαλιών.

Στο πλαίσιο της παρούσας εργασίας, πραγματοποιήθηκε εξαγωγή και συγκέντρωση όλων των log αρχείων από το περιβάλλον ESXi του host esx-vshere3.kos.gr, σε διαδοχικά στιγμιότυπα. Τα δεδομένα προήλθαν από διαδοχικά hosts dumps, τα οποία αντιστοιχούν σε πλήρη αποτυπώματα της λειτουργίας του συστήματος ανά περίπου 3 λεπτά. Η συλλογή αυτή έδωσε ουσιαστικά 13 διακριτά αρχεία καταγραφής, τα οποία μετατράπηκαν σε μορφή CSV για ομοιογενή ανάλυση:

- auth.log
- dhclient.log
- esxupdate.log
- fdm.log
- hostd.log
- shell.log
- syslog.log
- vmauthd.log
- vmkernel.log
- vmkeventd.log
- vmkwarning.log
- vobd.log
- vpxa.log

Κάθε ένα από αυτά τα αρχεία καταγράφει διαφορετικού τύπου γεγονότα. Για παράδειγμα το vmkernel.log περιέχει τις πιο κρίσιμες πληροφορίες για τη λειτουργία του πυρήνα του hypervisor καταγράφοντας διεργασίες I/O, διαχείριση storage και ενδεχόμενα σφάλματα hardware. Το hostd.log αντιπροσωπεύει τον agent του ESXi που δέχεται και εκτελεί εντολές διαχείρισης από το vCenter Server, καταγράφοντας ενέργειες όπως δημιουργία, τερματισμό ή μετακίνηση εικονικών μηχανών. Το vpxa.log λειτουργεί ως ενδιάμεσο κανάλι επικοινωνίας ανάμεσα στον host και το vCenter και περιλαμβάνει μηνύματα συγχρονισμού, εντολές και αποκρίσεις διαχείρισης.

Στα vmkwarning.log και syslog.log καταγράφονται προειδοποιήσεις και γενικά συμβάντα του συστήματος, όπως εκκινήσεις υπηρεσιών, ενημερώσεις ή προσωρινές αστοχίες. Το shell.log παρακολουθεί τη δραστηριότητα της ESXi Shell, όπου φαίνονται χειροκίνητες ενέργειες διαχειριστών ή τοπικές παρεμβάσεις στο host. Το auth.log είναι εστιασμένο στη διαχείριση πρόσβασης και την επαλήθευση χρηστών, ενώ το fdm.log σχετίζεται με την υπηρεσία Fault Tolerance Management. Το esxupdate.log καταγράφει ενημερώσεις και αναβαθμίσεις του συστήματος, το vobd.log παρακολουθεί μηχανισμούς παρακολούθησης σφαλμάτων και το vmauthd.log σχετίζεται με τον μηχανισμό πιστοποίησης vMotion και VMware Tools. Τέλος, το vmkeventd.log ανήκει στο σύστημα εξυπηρέτησης συμβάντων (VMkernel Event Daemon) και είναι ιδιαίτερα χρήσιμο για τη χρονοσειριακή ανάλυση των γεγονότων ενός host.

Η ποικιλία και η πυκνότητα των logs αποτυπώνουν την πολυεπίπεδη λειτουργία του vSphere. Κάθε κατηγορία παράγει διαφορετικό είδος πληροφορίας, για παράδειγμα οι προειδοποιήσεις του vmkwarning είναι προάγγελοι αποτυχιών, ενώ οι εγγραφές του hostd αποκαλύπτουν ανθρώπινες ενέργειες ή αυτόματες εντολές του vCenter. Η ενοποίηση όλων των αρχείων σε μορφή CSV επιτρέπει την ομοιογενή επεξεργασία και τη μετατροπή τους σε dataset κατάλληλο για αλγορίθμους μηχανικής μάθησης. Σύμφωνα με πρόσφατες μελέτες, η πολυπηγική ανάλυση (log fusion) βελτιώνει αισθητά την ακρίβεια στην ανίχνευση ανωμαλιών, καθώς οι αλγόριθμοι εκμεταλλεύονται τη διασύνδεση γεγονότων μεταξύ υποσυστημάτων [39].

Η παρούσα συλλογή αρχείων καταγραφής αποτελεί τον πυρήνα της πειραματικής ενότητας της εργασίας. Η ενοποίησή τους σε κοινή μορφή και η ταξινόμησή τους ανά host επιτρέπουν την παρακολούθηση της χρονικής εξέλιξης των συμβάντων.

Στην επόμενη ενότητα περιγράφεται η διαδικασία με την οποία τα αρχεία αυτά συγκεντρώθηκαν και οργανώθηκαν σε ένα ενιαίο αρχείο δεδομένων, έτοιμο για περαιτέρω επεξεργασία.

2.4 Συλλογή των αρχείων καταγραφής και αρχική οργάνωση των δεδομένων

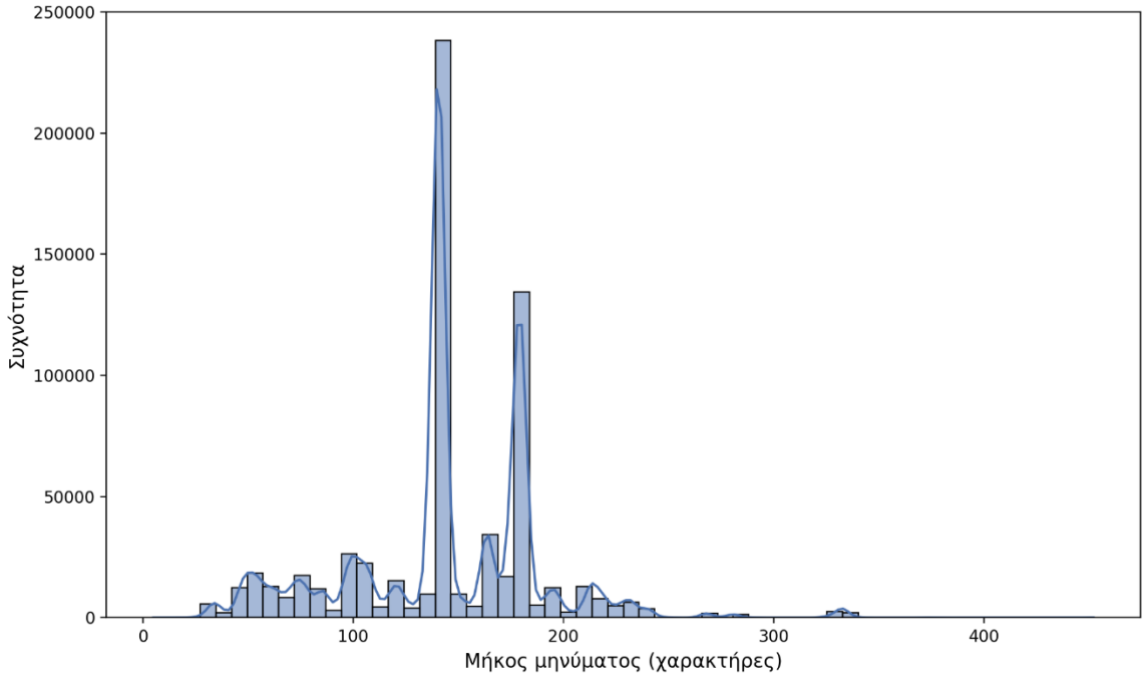
Όπως αναφέρθηκε στην προηγούμενη ενότητα, το vSphere δημιουργεί διαφορετικά αρχεία καταγραφής για κάθε υποσύστημα, όπως για παράδειγμα τα αρχεία που αφορούν τον πυρήνα, τις υπηρεσίες διαχείρισης, την ασφάλεια και τη δικτυακή λειτουργία. Η συλλογή των δεδομένων έγινε με συγκέντρωση όλων αυτών των αρχείων από τα διαδοχικά host dumps, ώστε να σχηματιστεί μια πλήρης εικόνα της δραστηριότητας του συστήματος σε βάθος χρόνου.

Αφού ολοκληρώθηκε η διαδικασία εξαγωγής, όλα τα αρχεία αποσυμπιέστηκαν και τοποθετήθηκαν σε έναν ενιαίο φάκελο εργασίας. Με τη χρήση της γλώσσας Python και της βιβλιοθήκης pandas δημιουργήθηκε ένα αρχικό αρχείο μορφής CSV, το οποίο συγκέντρωσε όλες τις γραμμές από τα διαφορετικά logs. Κάθε γραμμή του αρχείου αντιστοιχεί σε ένα γεγονός και περιλαμβάνει τρία βασικά πεδία:

- το όνομα του host,
- το όνομα του αρχείου καταγραφής από το οποίο προήλθε η γραμμή,
- το πλήρες ακατέργαστο κείμενο του μηνύματος (raw_line)

Στο στάδιο αυτό δεν εφαρμόστηκε ακόμη σύνθετη ανάλυση ή εξειδικευμένος καθαρισμός. Στόχος ήταν πρώτα να δημιουργηθεί ένα συνεκτικό και πλήρες αρχείο, στο οποίο να έχουν συγκεντρωθεί όλα τα μηνύματα του συστήματος σε κοινή μορφή. Με αυτό τον τρόπο έγινε εφικτή η χρονολογική ταξινόμηση των γεγονότων, αλλά και η μελλοντική συσχέτιση συμβάντων ανάμεσα σε διαφορετικά αρχεία καταγραφής.

Για να αποκτηθεί μια πρώτη εικόνα της μορφής των δεδομένων, υπολογίστηκε το μήκος κάθε μηνύματος καταγραφής και δημιουργήθηκε η κατανομή των τιμών αυτών. Η απεικόνιση παρουσιάζεται στο ακόλουθο σχήμα:



Σχήμα 2.1: Κατανομή μήκους μηνυμάτων στα αρχεία καταγραφής του vSphere.

Η κατανομή δείχνει το μήκος των μηνυμάτων όπως προέκυψαν μετά τον βασικό καθαρισμό των logs. Η πλειονότητα των καταγραφών συγκεντρώνεται γύρω από δύο διακριτές περιοχές, με μέσο μήκος περίπου 140-180 χαρακτήρες. Το γεγονός αυτό υποδηλώνει ότι το σύστημα παράγει κυρίως δύο τύπους εγγραφών: σύντομα μηνύματα κατάστασης και πιο εκτενείς περιγραφές γεγονότων ή σφαλμάτων. Η εικόνα αυτή επιβεβαιώνει ότι τα δεδομένα παρουσιάζουν δομή και επαναληψιμότητα, στοιχεία που είναι απαραίτητα για να εφαρμοστούν αργότερα τεχνικές ανάλυσης.

Το αποτέλεσμα της φάσης συλλογής και αρχικής οργάνωσης είναι ένα ενιαίο σύνολο ακατέργαστων καταγραφών, ταξινομημένο ανά host και αρχείο και έτοιμο για περαιτέρω προ-επεξεργασία. Στο Κεφάλαιο 3 παρουσιάζεται αναλυτικά η μεθοδολογία που ακολουθήθηκε, οι επιλεγμένοι αλγόριθμοι, καθώς και ο τρόπος με τον οποίο αξιολογήθηκε η ικανότητά τους να εντοπίζουν ασυνήθιστες ή μη φυσιολογικές συμπεριφορές στα logs του vSphere.

3 Μεθοδολογία

3.1 Εισαγωγή

Η μεθοδολογία που ακολουθήθηκε στην εργασία αυτή στοχεύει στη συστηματική ανίχνευση ανωμαλιών στα αρχεία καταγραφής του συστήματος VMware vSphere. Η ανίχνευση ανωμαλιών αποτελεί κρίσιμο κομμάτι της διαχείρισης πληροφοριακών συστημάτων, καθώς επιτρέπει την έγκαιρη ανίχνευση προβλημάτων πριν αυτά εξελιχθούν σε πραγματικές βλάβες ή διακοπές λειτουργίας. Στην πράξη, τα logs ενός συστήματος περιλαμβάνουν εκατοντάδες χιλιάδες γραμμές, οι περισσότερες εκ των οποίων καταγράφουν φυσιολογική δραστηριότητα. Οι ανωμαλίες, δηλαδή τα γεγονότα που υποδηλώνουν κάποια δυσλειτουργία ή ασυνήθιστη συμπεριφορά, είναι λίγες και συνήθως κρυμμένες μέσα σε αυτή τη μάζα των δεδομένων. Η εύρεσή τους με χειροκίνητο τρόπο όπως αναφέρθηκε και παραπάνω είναι πρακτικά αδύνατη, γι' αυτό και επιλέγεται η προσέγγιση της μηχανικής μάθησης, η οποία επιτρέπει την αυτόματη ανάλυση και ταξινόμηση.

Η βασική ιδέα της μεθοδολογίας είναι ότι κάθε πληροφοριακό σύστημα έχει ένα συγκεκριμένο μοτίβο συμπεριφοράς το οποίο μπορεί να αναπαρασταθεί μαθηματικά. Όταν το σύστημα λειτουργεί ομαλά, τα αρχεία καταγραφής ακολουθούν συγκεκριμένες ακολουθίες και συχνότητες. Όταν όμως παρουσιαστεί πρόβλημα, το μοτίβο αυτό διαταράσσεται δηλαδή αλλάζει η συχνότητα, η σειρά ή το περιεχόμενο των μηνυμάτων. Ο στόχος των μεθόδων που χρησιμοποιήθηκαν είναι να μάθουν αυτό το φυσιολογικό μοτίβο και να εντοπίζουν κάθε απόκλιση από αυτό.

Στην εργασία χρησιμοποιήθηκαν τρεις διαφορετικές προσεγγίσεις μηχανικής μάθησης. Τα δεδομένα που προήλθαν από τα logs δεν διαθέτουν ετικέτες που να δηλώνουν αν μια εγγραφή είναι φυσιολογική ή όχι. Αυτό σημαίνει πως δεν είναι δυνατή η εφαρμογή επιβλεπόμενων μοντέλων, όπου το σύστημα μαθαίνει βάσει παραδειγμάτων. Αντίθετα, τα μοντέλα που χρησιμοποιήθηκαν μαθαίνουν μόνα τους τη φυσιολογική δομή των δεδομένων και χαρακτηρίζουν ως ανωμαλίες τις καταγραφές που αποκλίνουν από αυτή τη δομή.

Οι τρεις μέθοδοι που επιλέχθηκαν, δηλαδή Isolation Forest, One-Class SVM και νευρωνικό μοντέλο ακολουθιών με μονάδες GRU, έχουν έναν κοινό στόχο, αλλά διαφορετικό τρόπο προσέγγισης. Με τη συνδυασμένη τους χρήση, η εργασία εξετάζει τόσο στατιστικές όσο και χρονοσειριακές προσεγγίσεις στην ανίχνευση ανωμαλιών..

Η μεθοδολογία οργανώνεται σε διαδοχικά στάδια, ξεκινώντας από τον καθαρισμό και την προετοιμασία των logs, την εξαγωγή των απαραίτητων χαρακτηριστικών, τη μετατροπή

τους σε αριθμητική μορφή κατάλληλη για ανάλυση, και καταλήγοντας στην εφαρμογή των επιλεγμένων αλγορίθμων. Με αυτόν τον τρόπο, επιτυγχάνεται μια μετάβαση από τα ακατέργαστα δεδομένα σε χρήσιμα αποτελέσματα, ικανά να υποστηρίξουν τη λήψη αποφάσεων για τη σταθερότητα και ασφάλεια του συστήματος.

Η ανίχνευση ανωμαλιών σε αρχεία καταγραφής αποτελεί ιδιαίτερα απαιτητική διαδικασία, καθώς κάθε υποσύστημα μπορεί να παράγει διαφορετικού τύπου καταγραφές, με ποικίλες δομές, μορφές και επίπεδα λεπτομέρειας. Για τον λόγο αυτό, είναι κρίσιμο τα δεδομένα να υποστούν προσεκτική προ-επεξεργασία, ώστε να εξαλειφθούν οι θόρυβοι και οι μη χρήσιμες πληροφορίες. Μέσω τεχνικών κανονικοποίησης, καθαρισμού και ομαδοποίησης, τα logs μετατράπηκαν σε ομοιογενή μορφή, επιτρέποντας την αξιόπιστη εκπαίδευση των μοντέλων.

Αφού ολοκληρωθεί η προετοιμασία των logs, η ανάλυση βασίζεται σε τρία διαφορετικά μοντέλα. Ο Isolation Forest κοιτάζει τα δεδομένα στατιστικά και προσπαθεί να ξεχωρίσει τα σημεία που απομακρύνονται από τη μάζα των κανονικών εγγραφών. Ο One-Class SVM αντιμετωπίζει το πρόβλημα γεωμετρικά, χαράζοντας ένα όριο γύρω από την περιοχή όπου συγκεντρώνεται η συνηθισμένη συμπεριφορά του συστήματος. Το τρίτο μοντέλο είναι ένα νευρωνικό δίκτυο ακολουθιών με GRU, το οποίο παρατηρεί τη σειρά εμφάνισης των προτύπων στα logs και σημαίνει συναγερμό όταν η πραγματική συνέχεια μοιάζει πολύ απίθανη σε σχέση με όσα έχει μάθει.

Η επιλογή των συγκεκριμένων μοντέλων στηρίχθηκε σε σχετική βιβλιογραφία για ανίχνευση ανωμαλιών σε logs. Ο Isolation Forest και ο One-Class SVM εμφανίζονται συχνά σε εφαρμογές χωρίς labels, ενώ τα νευρωνικά δίκτυα ακολουθιών προτείνονται όταν ενδιαφέρει η χρονική ροή των γεγονότων και όχι μόνο τα στατικά χαρακτηριστικά κάθε γραμμής [40].

Παρά τις διαφορές τους, οι μέθοδοι αυτές ακολουθούν κοινή λογική καθώς επιχειρούν να μάθουν τη φυσιολογική κατάσταση του συστήματος μέσα από τα ίδια τα δεδομένα. Με αυτόν τον τρόπο, μπορούν να αναγνωρίζουν ως ύποπτες τις καταγραφές που διαφέρουν είτε ως προς τη συχνότητα, είτε ως προς το περιεχόμενο, είτε ως προς τη χρονική τους θέση μέσα στην ακολουθία των γεγονότων.

Η αξία της πολυμεθοδολογικής αυτής προσέγγισης βρίσκεται στην πολυδιάστατη κατανόηση της συμπεριφοράς του συστήματος. Κανένα μοντέλο δεν είναι επαρκές από μόνο του για να αποδώσει πλήρη εικόνα της λειτουργίας. Ο Isolation Forest εντοπίζει απομονωμένα σημεία, ο One-Class SVM εξετάζει τις οριακές περιπτώσεις, ενώ τα

Νευρωνικά Δίκτυα αποκαλύπτουν τις χρονικές συσχετίσεις που υποκρύπτουν πιο σύνθετα πρότυπα.

Επιπλέον, η μεθοδολογία αυτή έχει σχεδιαστεί με στόχο να μπορεί να επαναχρησιμοποιηθεί και να επεκταθεί. Παρέχει σαφή στάδια, παραμέτρους και διαδικασίες που μπορούν να προσαρμοστούν σε άλλους τύπους συστημάτων ή εφαρμογών. Έτσι, δεν περιορίζεται αποκλειστικά στο vSphere, αλλά μπορεί να αποτελέσει γενικό πρότυπο για την αυτοματοποιημένη παρακολούθηση μεγάλων υποδομών, όπως δίκτυα, βάσεις δεδομένων ή περιβάλλοντα cloud.

Τέλος, πρέπει να τονιστεί ότι η επιτυχία της ανίχνευσης ανωμαλιών δεν εξαρτάται μόνο από τη σωστή επιλογή αλγορίθμου, αλλά και από την ποιότητα της αναπαράστασης των δεδομένων. Η διαδικασία εξαγωγής χαρακτηριστικών παίζει καθοριστικό ρόλο καθώς καθορίζει το πώς βλέπει το μοντέλο τα δεδομένα. Χωρίς κατάλληλη αναπαράσταση, ακόμη και ο πιο ισχυρός αλγόριθμος δεν μπορεί να επιτύχει ικανοποιητικά αποτελέσματα. Για τον λόγο αυτό, η παρούσα εργασία δίνει ιδιαίτερη έμφαση στα στάδια προ-επεξεργασίας και μετασχηματισμού των logs, ώστε η είσοδος προς τα μοντέλα να είναι όσο το δυνατόν πιο καθαρή και αντιπροσωπευτική της πραγματικής λειτουργίας του συστήματος.

Συνοψίζοντας, η μεθοδολογία που εφαρμόστηκε καλύπτει όλα τα βασικά στάδια της ανάλυσης αρχείων καταγραφής. Συνδυάζει τεχνικές στατιστικής απομόνωσης, γεωμετρικής μοντελοποίησης και νευρωνικών δικτύων με κοινό στόχο τον έγκαιρο εντοπισμό σφαλμάτων και αποκλίσεων. Η ανάλυση που ακολουθεί στις επόμενες ενότητες παρουσιάζει αναλυτικά τα βήματα της προετοιμασίας των δεδομένων, τα χαρακτηριστικά των αλγορίθμων που επιλέχθηκαν και τα κριτήρια με τα οποία αξιολογήθηκε η αποτελεσματικότητά τους.

3.2 Προετοιμασία των Δεδομένων για Ανάλυση

Η προετοιμασία των δεδομένων αποτέλεσε το πιο κρίσιμο και χρονοβόρο στάδιο της μεθοδολογίας. Τα αρχεία καταγραφής του συστήματος vSphere είναι πολύπλοκα, ημι-δομημένα και συχνά ετερογενή, γεγονός που καθιστά απαραίτητη μια προσεκτική διαδικασία καθαρισμού και μετασχηματισμού πριν την εφαρμογή οποιουδήποτε αλγορίθμου μηχανικής μάθησης. Σκοπός της ενότητας αυτής είναι να περιγράψει τα στάδια

που μετέτρεψαν το ακατέργαστο κείμενο των logs σε αριθμητικό πίνακα δεδομένων (feature matrix), ικανό να τροφοδοτήσει τους αλγορίθμους ανίχνευσης ανωμαλιών.

Η διαδικασία περιλάμβανε πέντε βασικά στάδια:

- Parsing και καθαρισμός των γραμμών,
- Εξαγωγή προτύπων (templates) και παραμέτρων,
- Δημιουργία και κανονικοποίηση χαρακτηριστικών (feature engineering),
- Κωδικοποίηση κατηγορικών μεταβλητών (encoding),
- Μείωση διαστάσεων (dimensionality reduction).

Κάθε στάδιο περιγράφεται στη συνέχεια αναλυτικά.

3.2.1 Αναγκαιότητα προετοιμασίας αρχείων καταγραφής

Τα logs αποτελούν ένα ζωντανό αρχείο, όπου το σύστημα γράφει τον τρόπο που λειτουργεί. Σε κάθε στιγμή, το vSphere δημιουργεί μικρά κομμάτια πληροφορίας όπου κάθε μια γραμμή κειμένου περιγράφει μια ενέργεια.

Τα logs έχουν ανομοιόμορφη δομή καθώς άλλες γραμμές έχουν καθαρή χρονοσήμανση ενώ άλλες όχι. Μερικές περιέχουν μόνο μια σύντομη περιγραφή ενώ άλλες περιλαμβάνουν τεράστια ποσότητα τεχνικών λεπτομερειών. Αν κάποιος το σκεφτεί, σχηματίζουν μια εικόνα με θόρυβο και το πρόβλημα είναι ότι η μηχανική μάθηση δεν μπορεί να λειτουργήσει σε τέτοιο υλικό το οποίο περιέχει θόρυβο. Χρειάζεται μια καθαρή, οργανωμένη και συνεπή μορφή και για να επιτευχθεί αυτό, η προετοιμασία είναι αναγκαία.

Η προ-επεξεργασία επιτρέπει να αναγνωριστούν οι σταθερές δομές των αρχείων καταγραφής και δίνει τη δυνατότητα να απομονωθούν οι πληροφορίες που έχουν αξία και αποφεύγεται η σύγχυση από στοιχεία που δεν έχουν σημασία για την ανάλυση. Για παράδειγμα, η ίδια γραμμή μπορεί να εμφανίζεται εκατοντάδες φορές μέσα στο αρχείο, αλλά με διαφορετικά αναγνωριστικά, ημερομηνίες ή μικρές αποκλίσεις. Χωρίς κατάλληλο καθαρισμό, η παρουσία τέτοιων διαφορών θα εμφανίζεται ως διαφορετικό γεγονός, ενώ στην πραγματικότητα πρόκειται για το ίδιο μοτίβο.

Η βιβλιογραφία τονίζει συχνά ότι η ποιότητα του preprocessing καθορίζει την επιτυχία κάθε μεθόδου που ακολουθεί. Ο Xu, σε μια από τις πρώτες συστηματικές αναλύσεις, ανέφερε ότι

τα logs σπάνια έχουν πλήρη μορφή και ότι η εισαγωγή μιας οργανωμένης διαδικασίας καθαρισμού αυξάνει την απόδοση της ανάλυσης [41]. Η μελέτη αυτή συνδέθηκε με άλλα έργα, όπως το DeepLog που πρότεινε ότι η αναγνώριση ανωμαλιών γίνεται πιο αξιόπιστη όταν το κείμενο έχει οργανωθεί σε σταθερές δομές. Αυτό σημαίνει ότι η σωστή προετοιμασία δεν είναι απλώς τεχνική διαδικασία, αλλά προϋπόθεση για την επιτυχία της ανάλυσης.

Στα logs του vSphere, αυτό είναι ιδιαίτερα εμφανές. Οι γραμμές περιέχουν στοιχεία όπως ημερομηνίες, αναγνωριστικά hosts, τεχνικά μονοπάτια, αριθμούς και διάφορα προσωρινά δεδομένα. Αυτά δεν έχουν σταθερή δομή. Κάποιες γραμμές ξεκινούν με ημερομηνία, άλλες περιέχουν την ημερομηνία στη μέση ενώ σε πολλές περιπτώσεις, τα logs δεν έχουν καθόλου ημερομηνίες, είτε επειδή το σύστημα δεν πρόσθεσε χρονοσήμανση είτε επειδή η μορφή καταγραφής δεν το απαιτούσε.

Η προετοιμασία των logs εξυπηρετεί τρεις βασικούς στόχους. Ο πρώτος στόχος είναι η δημιουργία μιας κοινής μορφής. Αυτό επιτρέπει στους αλγορίθμους να αναγνωρίζουν την ίδια πληροφορία με τον ίδιο τρόπο. Ο δεύτερος στόχος είναι η απομόνωση της χρήσιμης πληροφορίας από αυτή η οποία δεν είναι χρήσιμη. Στα αρχεία καταγραφής υπάρχουν στοιχεία που δεν βοηθούν στην ανάλυση, όπως ιδιαίτερα μεγάλοι αριθμοί, κωδικοί ή τεχνικά μονοπάτια. Η αντικατάστασή τους με απλά σύμβολα μειώνει τον θόρυβο. Ο τρίτος στόχος είναι η δημιουργία ενός συνόλου δεδομένων που επιτρέπει την εξαγωγή χαρακτηριστικών. Αυτά τα χαρακτηριστικά θα αποτελέσουν τη βάση της ανάλυσης.

Στην παρούσα εργασία, τα logs που συλλέχθηκαν από το vSphere παρουσίαζαν υψηλή ετερογένεια. Η διαδικασία προετοιμασίας ήταν απαραίτητη για να μετατραπεί αυτός ο όγκος δεδομένων σε χαρακτηριστικά που μπορούν να αναλυθούν. Η μετατροπή της πληροφορίας έγινε με συγκεκριμένο τρόπο, που περιλαμβάνει την αναγνώριση των βασικών στοιχείων κάθε γραμμής, την ομαδοποίηση και τον καθαρισμό των μηνυμάτων. Οι μεταβλητές που προέκυψαν αντιπροσωπεύουν τη δομή του συστήματος και επιτρέπουν την εφαρμογή τεχνικών μηχανικής μάθησης.

Η προ-επεξεργασία έδειξε επίσης ότι τα logs περιέχουν κρυφές αποκλίσεις, οι οποίες δεν φαίνονται εύκολα με οπτικό τρόπο. Με την οργάνωση των δεδομένων αποκαλύφθηκαν ακολουθίες που συμβαίνουν σπάνια. Αυτές οι ακολουθίες ήταν δύσκολο να εντοπιστούν χωρίς την υποστήριξη των μεταβλητών που δημιουργήθηκαν. Η σωστή προετοιμασία επιτρέπει να αναγνωριστούν τέτοια φαινόμενα και να καταγραφούν ως πιθανές ανωμαλίες.

Η ανάγκη προετοιμασίας δεν είναι μόνο τεχνική. Είναι και μεθοδολογική. Η διαδικασία επιτρέπει στο σύστημα να μιλήσει με τρόπο που μπορεί να αναλυθεί. Η μετατροπή των logs από ακατέργαστο κείμενο σε σύνολο δεδομένων δημιουργεί τη βάση για την ανίχνευση των ανωμαλιών. Χωρίς αυτό το στάδιο, η μεθοδολογία δεν θα μπορούσε να προχωρήσει.
Η προεπεξεργασία εξασφαλίζει ότι τα δεδομένα έχουν την απαραίτητη ποιότητα και όπως αναφέραμε και προηγουμένως οι αλγόριθμοι μπορούν να λειτουργήσουν μόνο όταν τα δεδομένα έχουν σωστή δομή. Η προετοιμασία των δεδομένων αποτελεί το σημείο όπου η ασάφεια μετατρέπεται σε συγκεκριμένο σχήμα. Από εκεί και πέρα, οι διαδικασίες της μηχανικής μάθησης μπορούν να εφαρμοστούν με αξιοπιστία.

3.2.2 Ιδιαιτερότητες των αρχείων καταγραφής

Τα αρχεία καταγραφής του vSphere έχουν μια σειρά από χαρακτηριστικά που τα κάνουν ξεχωριστά σε σχέση με άλλους τύπους συστημάτων. Το περιεχόμενό τους δεν ακολουθεί έναν ενιαίο κανόνα, επειδή κάθε μέρος του hypervisor δημιουργεί δικές του μορφές καταγραφής.
Η πρώτη ιδιαιτερότητα αφορά τη δομή των γραμμών. Τα logs δεν είναι πλήρως δομημένα. Δεν σχηματίζουν σταθερές στήλες όπως σε ένα αρχείο CSV. Το κείμενο έχει ελεύθερη μορφή, όμως μέσα σε αυτό υπάρχουν επαναλαμβανόμενα σχήματα. Κάποια μηνύματα ξεκινούν με χρονοσήμανση. Άλλα την τοποθετούν μετά από μια μικρή περιγραφή. Υπάρχουν και περιπτώσεις όπου το σύστημα δεν καταγράφει ημερομηνία. Αυτό δημιουργεί μικρές ασυνέχειες που δεν φαίνονται εύκολα, αλλά έχουν σημασία κατά την ανάλυση.
Μια δεύτερη ιδιαιτερότητα προκύπτει από το περιεχόμενο. Τα logs περιλαμβάνουν τεχνικά στοιχεία όπως διευθύνσεις δικτύου, μονοπάτια αρχείων, ακολουθίες αριθμών και αναγνωριστικά. Αυτά τα στοιχεία αλλάζουν σχεδόν σε κάθε γραμμή, παρότι η γενική δομή του μηνύματος παραμένει ίδια. Έτσι, δύο γραμμές που στην ουσία περιγράφουν το ίδιο γεγονός φαίνονται διαφορετικές εξαιτίας των μεταβλητών που περιέχουν. Η παρουσία αυτών των στοιχείων αυξάνει τον όγκο της πληροφορίας που πρέπει να φιλτραριστεί.
Μια τρίτη ιδιαιτερότητα αφορά το πλήθος των αρχείων. Το vSphere παράγει logs από πολλά υποσυστήματα. Για παράδειγμα, ο πυρήνας γράφει για τη λειτουργία του υλικού, η υπηρεσία hostd καταγράφει ενέργειες διαχείρισης, το vpxa λειτουργεί ως ενδιάμεσος ανάμεσα στον host και το κέντρο ελέγχου ενώ το syslog καλύπτει γενικά γεγονότα. Η

παρουσία τόσων διαφορετικών πηγών δημιουργεί εικόνα με πολλές πλευρές. Κάθε πλευρά δίνει μια διαφορετική οπτική για το ίδιο σύστημα.

Μία ακόμη ιδιαιτερότητα έχει να κάνει με τη συχνότητα καταγραφής. Κάποια αρχεία δημιουργούν μεγάλο όγκο γραμμών μέσα σε λίγα δευτερόλεπτα. Άλλα αρχεία παραμένουν σχεδόν αδρανή και ενεργοποιούνται μόνο όταν συμβεί συγκεκριμένο γεγονός. Αυτή η ασυμμετρία οδηγεί σε ανομοιόμορφη πυκνότητα δεδομένων. Η ανομοιομορφία αυτή επηρεάζει την χρονοσειρά και δημιουργεί διαστήματα με διαφορετική ένταση.

Τέλος, τα αρχεία καταγραφής περιέχουν αρκετές γραμμές που δεν έχουν προφανή αξία. Ορισμένες γραμμές περιλαμβάνουν μηνύματα ελέγχου. Άλλες αποτυπώνουν εσωτερικές ρουτίνες του hypervisor. Αυτές οι γραμμές εμφανίζονται συχνά και μπορούν να κρύψουν πιο σημαντικές πληροφορίες. Για αυτό, η παρουσία τους πρέπει να αναγνωριστεί και να αντιμετωπιστεί σε επόμενα στάδια.

Η κατανόηση αυτών των ιδιαιτεροτήτων επιτρέπει πιο στοχευμένη επεξεργασία. Κάθε ένα από τα παραπάνω στοιχεία παίζει ρόλο στη δημιουργία των τελικών χαρακτηριστικών. Τα raw logs δεν είναι απλώς κείμενο, σχηματίζουν ένα σύνολο πληροφοριών που χρειάζεται προσεκτική ανάγνωση και η σωστή κατανόηση της δομής τους διευκολύνει τα επόμενα βήματα της προετοιμασίας.

3.2.3 Καθαρισμός και μετασχηματισμός των μηνυμάτων

Ο καθαρισμός των μηνυμάτων αποτέλεσε ένα από τα πιο απαιτητικά στάδια της προετοιμασίας. Τα αρχεία καταγραφής που παράγει το vSphere περιέχουν πολλά στοιχεία τα οποία δεν βοηθούν την ανάλυση. Το κείμενο κάθε γραμμής μοιάζει με απλή πρόταση όμως στην πράξη έχει μέσα του τιμές που αλλάζουν συνεχώς, τεχνικούς όρους, αναγνωριστικά και διαδρομές συστήματος. Αυτά δημιουργούν μεγάλη ποικιλία που δεν αντικατοπτρίζει τη λειτουργική δομή του συστήματος. Για να σχηματιστεί σταθερό σύνολο δεδομένων, το κείμενο χρειάστηκε προσεκτικό καθαρισμό.

Η διαδικασία ξεκίνησε με την αναγνώριση των στοιχείων που δημιουργούν θόρυβο. Οι διευθύνσεις IP ήταν τα πιο συχνά και εμφανίζονταν σε πολλές γραμμές, με μικρές αλλαγές. Η παρουσία τους αλλοιώνει το κείμενο χωρίς να προσφέρει χρήσιμη πληροφορία έτσι κατά τον καθαρισμό αντικαταστάθηκαν από ένα ενιαίο σύμβολο. Η χρήση αυτού του συμβόλου επέτρεψε την αναγνώριση της ίδιας λογικής γραμμής, ακόμη και όταν η διεύθυνση άλλαζε σε κάθε εκτέλεση.

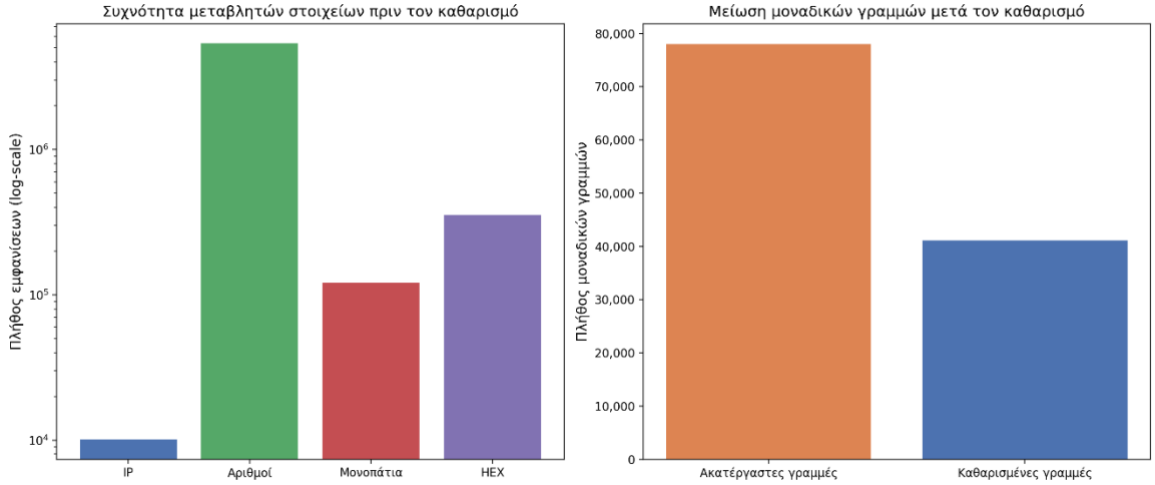
Στη συνέχεια καθαρίστηκαν οι αριθμοί αφού τα logs περιείχαν μεγάλο πλήθος αριθμών που σχετίζονται με χρονικά βήματα, counters, μεγέθη ή αναγνωριστικά. Αυτοί οι αριθμοί δεν έδιναν σταθερό μοτίβο και η μεταβολή τους δυσκόλευε την αναγνώριση της κανονικής δομής. Αντικαταστάθηκαν από σύμβολο που δεν επηρεάζει το υπόλοιπο μήνυμα έτσι έγινε πιο εύκολη η ομαδοποίηση παρόμοιων καταστάσεων.

Ένα ακόμη σημείο ήταν τα μονοπάτια αρχείων. Αυτά είχαν μεγάλη ποικιλία και περιείχαν ακολουθίες που δεν εμφανίζονται σε σταθερή μορφή και η παρουσία τους οδηγεί σε ψευδή διαφοροποίηση. Η αντικατάστασή τους με μικρό δείκτη απλοποίησε το κείμενο και επέτρεψε την εξαγωγή καθαρότερου προτύπου ανά γραμμή. Παρόμοια μεταχείριση δόθηκε και στις δεκαεξαδικές ακολουθίες αφού εμφανίζονται συχνά σε καταγραφές του πυρήνα και αλλάζουν σε κάθε εκτέλεση. Έτσι λοιπόν και αυτές μετατράπηκαν σε ενιαίο σύμβολο, ώστε να αποφεύγεται η δημιουργία τεχνητής ποικιλίας.

Μετά την αφαίρεση των μεταβλητών στοιχείων, κάθε γραμμή πήρε πιο καθαρή μορφή ενώ το μήνυμα παρέμεινε στην ουσία του ίδιο, αλλά απέκτησε σταθερό σχήμα. Η σταθερότητα αυτή είναι σημαντική, επειδή οι αλγόριθμοι αναγνώρισης χρειάζονται κείμενο με επαναλαμβανόμενη δομή. Αν το κείμενο διαφέρει κάθε φορά, η ανάλυση γίνεται αβέβαιη. Με τον καθαρισμό απομακρύνθηκαν τα στοιχεία που δημιουργούν διαφοροποιήσεις χωρίς ουσιαστικό λόγο.

Ακόμη, έγινε και έλεγχος του συνολικού περιεχομένου. Πολλές γραμμές είχαν σύμβολα που δεν συνδέονται με το υπόλοιπο κείμενο και η αφαίρεσή τους έδωσε πιο απλή δομή. Η απλότητα αυτή βοηθά τη δημιουργία χαρακτηριστικών σε επόμενα στάδια. Τα χαρακτηριστικά που χρησιμοποιούνται σε μοντέλα βασίζονται σε στατιστική σχέση με το μήνυμα. Με καθαρό μήνυμα, η δημιουργία χαρακτηριστικών γίνεται πιο σταθερή.

Η διαδικασία καθαρισμού είχε ως αποτέλεσμα δύο πράγματα. Πρώτον, τα μηνύματα έγιναν πιο συνεπή αφού η συνοχή αυτή επιτρέπει στο σύστημα να αναγνωρίζει το ίδιο πρότυπο σε πολλές γραμμές. Δεύτερον, η ποσότητα των μοναδικών γραμμών μειώθηκε και η μείωση αυτή δείχνει ότι πολλές γραμμές είχαν την ίδια λογική δομή αλλά διαφορετικές τιμές. Με την αφαίρεση των τιμών, οι γραμμές αυτές ενώθηκαν σε κοινό σύνολο.



Σχήμα 3.1: Επίδραση του καθαρισμού στα δεδομένα.

Στο αριστερό διάγραμμα φαίνεται ο αριθμός των μεταβλητών στοιχείων που εντοπίστηκαν πριν την επεξεργασία. Στο δεξί διάγραμμα απεικονίζεται η μείωση των μοναδικών γραμμών μετά τον καθαρισμό, γεγονός που δείχνει την εξάλειψη θορύβου και την ενοποίηση παρόμοιων μηνυμάτων σε κοινά μοτίβα

Ο καθαρισμός και η μετατροπή των γραμμών δημιούργησαν βάση για πιο βαθιά ανάλυση. Τα logs στη νέα μορφή τους επιτρέπουν τη δημιουργία templates, τη μέτρηση συχνότητας και την εξαγωγή μεταβλητών που βοηθούν στην αναγνώριση ανωμαλιών. Κάθε γραμμή μετά τον καθαρισμό περιέχει την ουσία της λειτουργίας και αφήνει έξω στοιχεία που θα δημιουργούσαν δυσκολίες. Αυτό το στάδιο είναι καθοριστικό για την ποιότητα των επόμενων βημάτων.

3.2.4 Δημιουργία προτύπων

Η εξαγωγή προτύπων ήταν το επόμενο σταθερό βήμα μέσα στη διαδικασία προετοιμασίας. Αφού τα αρχεία καταγραφής περιείχαν πολλές γραμμές με παρόμοια δομή, οι γραμμές αυτές άλλαζαν μόνο σε στοιχεία που δεν είχαν σταθερή αξία. Η διαδικασία masking είχε ήδη μειώσει αυτά τα στοιχεία άρα το επόμενο ζητούμενο ήταν η αναγνώριση του σταθερού μέρους κάθε γραμμής έτσι αυτό το σταθερό μέρος αποτέλεσε το πρότυπο.

Στην ουσία, κάθε μήνυμα είχε λίγες λέξεις που το χαρακτήριζαν και οι πρώτες λέξεις της γραμμής έδιναν συχνά μια σύντομη περιγραφή του γεγονότος αυτού. Για αυτό επιλέχθηκαν τα τρία πρώτα tokens ως βάση ενός προτύπου. Η επιλογή αυτή έδωσε σταθερότητα και

διατήρησε το νόημα. Οι γραμμές που μοιράζονταν αυτά τα tokens ανήκαν στην ίδια κατηγορία γεγονότων.

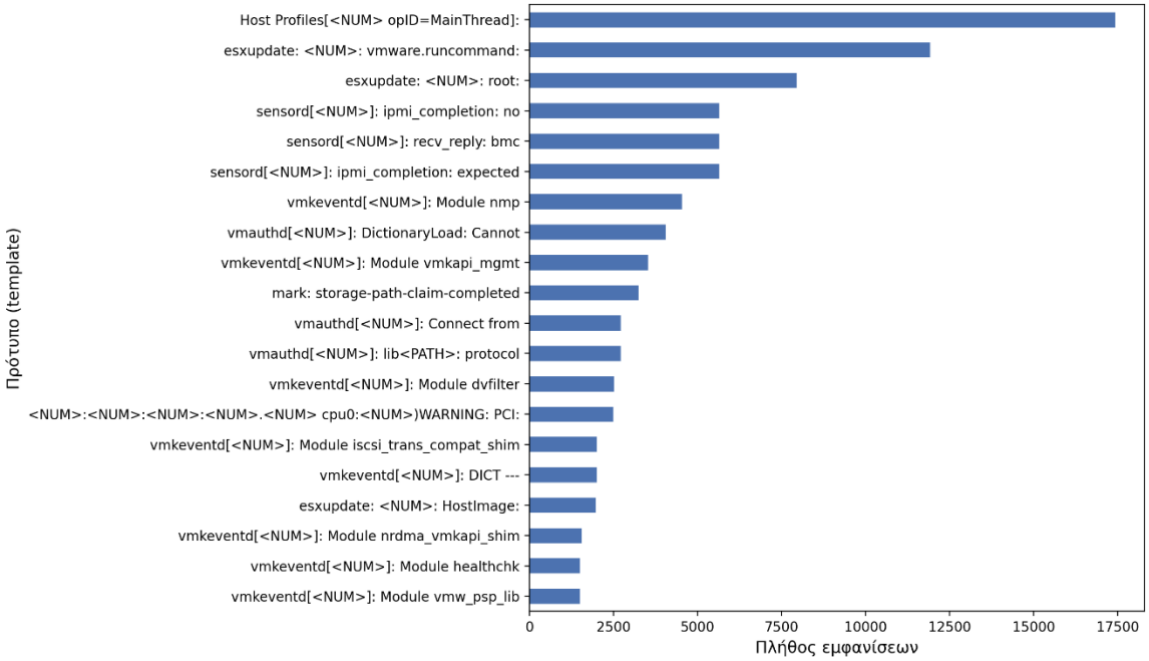
Το επόμενο στάδιο ήταν η δημιουργία μοναδικού αναγνωριστικού. Το αναγνωριστικό αυτό έπρεπε να παραμένει ίδιο για κάθε γραμμή που είχε κοινό πρότυπο. Για αυτό χρησιμοποιήθηκε συνάρτηση κατακερματισμού όπως η χρήση hash που έδωσε μικρό και σταθερό δείκτη. Ο δείκτης αυτός διευκόλυνε την ομαδοποίηση και επέτρεψε τη μέτρηση συχνότητας.

Η συχνότητα είχε μεγάλη σημασία. Οι γραμμές που εμφανίζονταν συχνά ανήκαν σε συνηθισμένες λειτουργίες του συστήματος ενώ οι γραμμές που εμφανίζονταν λίγες φορές έδιναν πιο σπάνιες περιπτώσεις. Η καταγραφή της συχνότητας έδειξε τη συμπεριφορά του συστήματος με καθαρό τρόπο και έδωσε δείκτη για το ποιες καταγραφές αντιστοιχούν σε σταθερά μοτίβα και ποιες δείχνουν μικρές αποκλίσεις.

Η διαδικασία παραγωγής προτύπων υποστήριξε και την αναγνώριση ακολουθιών. Με σταθερά πρότυπα, οι ακολουθίες μπορούσαν να εξεταστούν χωρίς τον θόρυβο που είχαν οι μεταβλητές τιμές έτσι το σύστημα μπορούσε να διαβάσει την εξέλιξη των γεγονότων με σαφή ροή. Η ύπαρξη προτύπων επέτρεψε να εντοπιστούν σπάνιοι συνδυασμοί μέσα σε μεγάλες ακολουθίες.

Ενώ η αρχική γραμμή έδινε πολύπλοκο κείμενο, το πρότυπο έδινε σταθερή μορφή. Η σταθερή μορφή επέτρεψε την αναγνώριση μοτίβων και η αναγνώριση αυτών των μοτίβων έδωσε τη βάση για την επόμενη ανάλυση.

Η μέθοδος αυτή είχε και ένα ακόμη πλεονέκτημα. Μείωσε τον αριθμό των μοναδικών στοιχείων που έβλεπαν οι αλγόριθμοι. Η μείωση αυτή έκανε πιο σταθερή και πιο γρήγορη τη μάθηση.



Σχήμα 3.2: Συχνότερα λειτουργικά πρότυπα καταγραφών.

Το διάγραμμα παραπάνω απεικονίζει τα είκοσι πρότυπα καταγραφών με τη μεγαλύτερη συχνότητα εμφάνισης μετά τη διαδικασία καθαρισμού και ομαδοποίησης. Τα πρότυπα αναδεικνύουν τα πιο επαναλαμβανόμενα λειτουργικά γεγονότα του vSphere και επιβεβαιώνουν ότι η μέθοδος εξαγωγής templates συγκέντρωσε παρόμοια μηνύματα σε κοινές κατηγορίες. Η διαδικασία εξαγωγής προτύπων ολοκλήρωσε το στάδιο καθαρισμού. Η πληροφορία της γραμμής διατηρήθηκε, ενώ ο θόρυβος μειώθηκε έτσι το σύστημα πήρε σταθερό χάρτη των γεγονότων και των συχνότητων τους. Αυτός ο χάρτης αποτέλεσε τη βάση για την εισαγωγή χαρακτηριστικών στο επόμενο στάδιο.

3.2.5 Δημιουργία χαρακτηριστικών

Μετά τον καθαρισμό και την εξαγωγή προτύπων, το σύνολο των καταγραφών απέκτησε πιο καθαρή και οργανωμένη μορφή. Η πληροφορία όμως παρέμενε σε επίπεδο κειμένου και προκειμένου να είναι δυνατή η ανάλυση από αλγορίθμους μηχανικής μάθησης, το κείμενο έπρεπε να μετατραπεί σε αριθμητικές τιμές. Η μετατροπή αυτή έγινε με τον σχεδιασμό συγκεκριμένων χαρακτηριστικών που περιγράφουν όσο πιο αντικειμενικά γίνεται τη λειτουργική συμπεριφορά του συστήματος.

Η δημιουργία των χαρακτηριστικών στηρίχθηκε σε δύο βασικές αρχές. Η πρώτη ήταν η αποτύπωση πληροφοριών που σχετίζονται με το ίδιο το μήνυμα. Εδώ ενδιαφέρουν όσα στοιχεία δείχνουν το ύφος και το περιεχόμενο του κειμένου ενώ η δεύτερη αρχή ήταν η αποτύπωση της χρονικής συμπεριφοράς. Τα αρχεία καταγραφής δεν είναι ανεξάρτητα μηνύματα και αποτελούν ακολουθίες γεγονότων. Η κάθε γραμμή είναι μέρος μιας ευρύτερης ροής και για αυτό χρησιμοποιήθηκαν χαρακτηριστικά που περιγράφουν την εξέλιξη των γεγονότων μέσα στον χρόνο.

Το πρώτο χαρακτηριστικό που δημιουργήθηκε ήταν το μήκος του μηνύματος. Η τιμή αυτή είναι απλή αλλά λειτουργική. Τα μηνύματα μικρού μήκους συνήθως ανήκουν σε απλές λειτουργίες ενώ τα μηνύματα μεγάλου μήκους περιέχουν σύνθετες πληροφορίες. Η διαφορά ανάμεσα στα δύο μπορεί να δώσει ενδείξεις για τη φύση των γεγονότων. Σε ορισμένες περιπτώσεις, οι μεγάλες διακυμάνσεις στο μήκος αποτελούν σημάδι ασυνήθιστης συμπεριφοράς.

Για να γίνει το χαρακτηριστικό αυτό πιο εύκολο στη χρήση, εφαρμόστηκε διαδικασία κανονικοποίησης. Η διαδικασία αυτή δίνει σε κάθε τιμή μέτρο σύγκρισης με τις υπόλοιπες. Η κανονικοποίηση βοηθά τους αλγορίθμους να λειτουργούν ομαλά, επειδή δεν επηρεάζονται από ακραίες τιμές, έτσι το μήκος μετατράπηκε σε νέα αριθμητική μεταβλητή, πιο εύχρηστη για ανάλυση.

Η επόμενη ομάδα χαρακτηριστικών είχε σχέση με τον χρόνο αφού τώρα πια κάθε μήνυμα περιείχε χρονοσήμανση. Η χρονοσήμανση μετατράπηκε σε επιμέρους στοιχεία και από αυτή προέκυψαν η ώρα, η ημέρα της εβδομάδας και ο δείκτης που έδειχνε αν μια καταγραφή έγινε σε ημέρα αργίας ή σε ημέρα εργασίας. Αυτές οι πληροφορίες δίνουν χρήσιμα σήματα για το πότε συμβαίνουν συγκεκριμένα γεγονότα. Σε πραγματικά συστήματα, η λειτουργία αλλάζει ανάλογα με την ώρα και τη μέρα. Η αποτύπωση αυτής της αλλαγής είναι σημαντική κατά την ανάλυση.

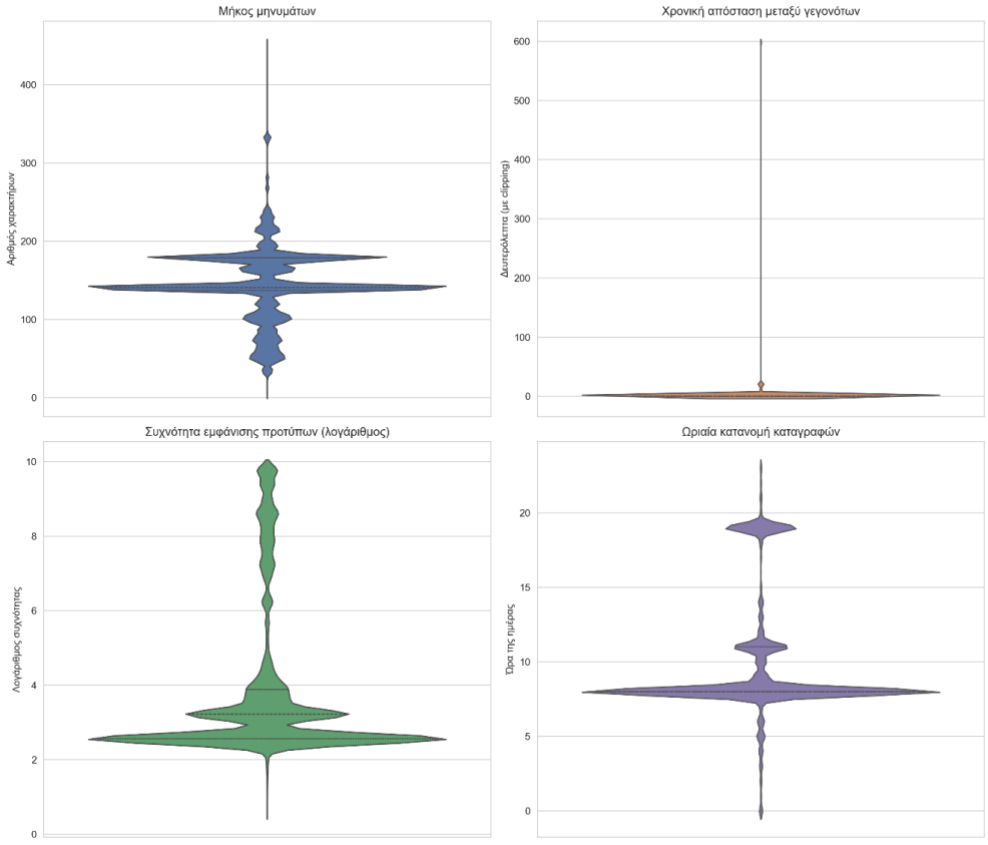
Ένα ακόμη σημαντικό χαρακτηριστικό προέκυψε από τη διαφορά χρόνου ανάμεσα σε διαδοχικές γραμμές. Το χαρακτηριστικό αυτό αποτυπώνει την ταχύτητα με την οποία εμφανίζονται γεγονότα αφού μικρές τιμές υποδηλώνουν έντονη δραστηριότητα και μεγάλες τιμές δείχνουν ήσυχα διαστήματα. Η παρουσία απότομων αλλαγών μπορεί να οδηγήσει σε υποψία ασυνήθιστης συμπεριφοράς και για αυτό η διαφορά χρόνου καταγράφηκε για κάθε γραμμή, ομαδοποιημένη ανά host.

Παράλληλα, αξιοποιήθηκε η συχνότητα εμφάνισης του κάθε template. Η συχνότητα αποτελεί καθοριστικό παράγοντα για τον εντοπισμό σπάνιων γεγονότων. Όσο πιο σπάνιο

είναι ένα πρότυπο, τόσο πιο πιθανό είναι να αντιστοιχεί σε ασυνήθιστη ενέργεια του συστήματος. Οι συχνότητες υπολογίστηκαν αυτόματα και προστέθηκαν στον πίνακα δεδομένων. Με αυτό τον τρόπο, κάθε γραμμή συνδέθηκε με ένα αριθμητικό μέτρο της κανονικότητάς της.

Ένα ακόμη στοιχείο που βοηθά τη μηχανική μάθηση είναι οι ακολουθίες των προτύπων. Παρότι οι ακολουθίες αναλύθηκαν εκτενέστερα σε επόμενο στάδιο, στη δημιουργία χαρακτηριστικών συμπεριλήφθηκε και το προηγούμενο template. Η πληροφορία αυτή προστέθηκε απλά για να υποστηρίξει αργότερα την αναγνώριση μεταβατικών σημείων. Σε πολλές περιπτώσεις, η ανωμαλία δεν βρίσκεται στο ίδιο το μήνυμα αλλά στη σειρά με την οποία εμφανίζονται τα γεγονότα.

Η συνολική διαδικασία έδωσε ένα σύνολο από αριθμητικά χαρακτηριστικά που περιγράφουν όσο πιο καθαρά γίνεται τη λειτουργία του συστήματος. Ο πίνακας που προέκυψε ήταν έτοιμος για επόμενες τεχνικές, όπως η μετατροπή κειμένου σε διανύσματα και η δημιουργία πιο σύνθετων αναπαραστάσεων. Σε αυτό το στάδιο διαμορφώθηκε η βάση πάνω στην οποία λειτουργούν οι αλγόριθμοι ανίχνευσης ανωμαλιών.



Σχήμα 3.3: Κάθετα violin plots των βασικών χαρακτηριστικών που δημιουργήθηκαν.

Τα διαγράμματα δείχνουν τη συμπεριφορά τεσσάρων χαρακτηριστικών που προέκυψαν από τα καθαρισμένα δεδομένα. Το μήκος των μηνυμάτων συγκεντρώνεται σε συγκεκριμένο εύρος τιμών, γεγονός που αντανακλά την τυπική μορφή των καταγραφών του συστήματος. Η χρονική απόσταση ανάμεσα σε γεγονότα είναι μικρή για το μεγαλύτερο μέρος των γραμμών, με λίγες εξαιρέσεις που αφορούν πιο αργές μεταβάσεις. Η συχνότητα εμφάνισης των προτύπων παρουσιάζει έντονη ανισορροπία, αφού λίγα πρότυπα εμφανίζονται συχνά και πολλά εμφανίζονται πολύ σπάνια. Η κατανομή των ωρών δείχνει συγκέντρωση γεγονότων σε συγκεκριμένα χρονικά διαστήματα, κάτι που συνδέεται με τον τρόπο λειτουργίας των υπηρεσιών του vSphere.

3.2.6 Μετατροπή σε αριθμητική μορφή με χρήση TF-IDF

Η μετατροπή του κειμένου σε αριθμητική μορφή ήταν απαραίτητο βήμα. Τα μηνύματα έμειναν ως κείμενο μετά τον καθαρισμό, και οι αλγόριθμοι δεν μπορούν να δουλέψουν με κείμενο άμεσα. Η μέθοδος TF-IDF έδωσε έναν πρακτικό τρόπο να αποτυπωθεί η σημασία κάθε λέξης χωρίς να χαθεί το περιεχόμενο. Κάθε λέξη παίρνει βάρος ανάλογα με το πόσο συχνά εμφανίζεται σε μια γραμμή και πόσο σπάνια είναι στο σύνολο των μηνυμάτων. Οι λέξεις που επαναλαμβάνονται σε όλα τα logs δίνουν μικρή πληροφορία και παίρνουν μικρό βάρος και οι λέξεις που ξεχωρίζουν σε συγκεκριμένα γεγονότα παίρνουν μεγαλύτερο.

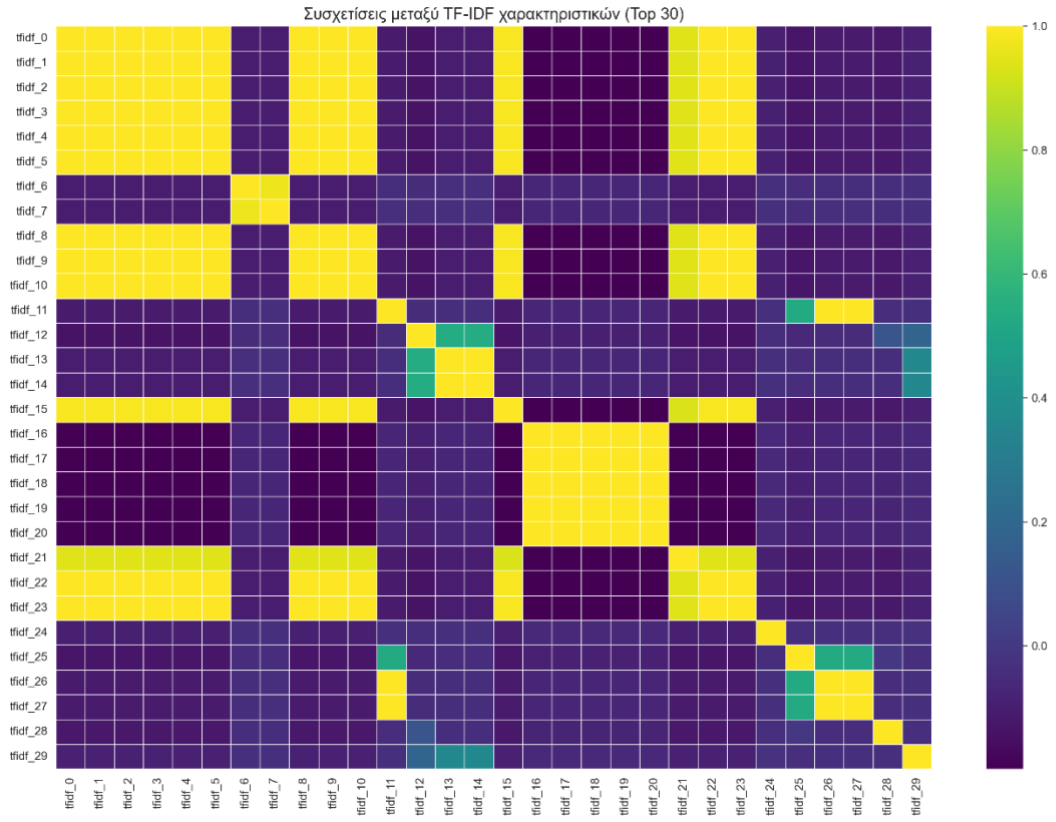
Η διαδικασία ξεκίνησε με τον διαχωρισμό του κειμένου σε λέξεις. Οι λέξεις κρατήθηκαν σε απλή μορφή χωρίς περίπλοκες γλωσσολογικές επεξεργασίες, γιατί τα logs έχουν σταθερή δομή και δεν χρειάζονται λεπτές γλωσσικές παρεμβάσεις. Στη συνέχεια δημιουργήθηκε ένα λεξιλόγιο με τις πιο συχνές λέξεις. Το λεξιλόγιο είχε περιορισμένο μέγεθος ώστε να υπάρχει ισορροπία ανάμεσα στην ακρίβεια και τη σταθερότητα των υπολογισμών. Το μέγεθος των 200 χαρακτηριστικών ήταν αρκετό για να αποτυπώσει τη συνολική εικόνα.

Κάθε μήνυμα μετατράπηκε σε διάνυσμα με σταθερό μήκος. Το διάνυσμα αυτό είχε μια τιμή για κάθε λέξη του λεξιλογίου και η τιμή αυτή έδειχνε πόσο σημαντική ήταν η λέξη στο συγκεκριμένο μήνυμα. Το αποτέλεσμα ήταν πίνακας με αριθμητικές τιμές που διατηρούσε το περιεχόμενο του κειμένου χωρίς να εξαρτάται από τη σειρά των λέξεων.

Η χρήση TF-IDF αποκάλυψε αρκετές λεπτομέρειες. Ορισμένες λέξεις εμφανίζονταν σε μεγάλα τμήματα του συστήματος και είχαν μικρό βάρος ενώ άλλες λέξεις ήταν πιο σπάνιες και συνδέονταν με συγκεκριμένες λειτουργίες ή μικρές εξαιρέσεις. Αυτό το βάρος βοήθησε

στην ανίχνευση ασυνήθιστων συμπεριφορών. Αν ένα μήνυμα περιείχε λέξεις που το σύστημα δεν έβλεπε συχνά, το άθροισμα των τιμών στο διάνυσμα ήταν διαφορετικό από τα συνηθισμένα.

Η μετατροπή σε TF-IDF λειτούργησε ως συμπλήρωμα αφού τα αριθμητικά στοιχεία, όπως το μήκος του μηνύματος και τα χρονικά διαστήματα, έδιναν δομική πληροφορία. Το TF-IDF έδινε λεπτή κειμενική πληροφορία και ο συνδυασμός τους έκανε τη συμπεριφορά των logs πιο ευδιάκριτη. Οι αλγόριθμοι μπορούσαν να ξεχωρίσουν μοτίβα που δεν φαίνονταν καθαρά ούτε από το καθαρό κείμενο ούτε από τα απλά χαρακτηριστικά. Κάθε γραμμή είχε αριθμητική αναπαράσταση που περιέγραφε το περιεχόμενο, τη δομή και τη συχνότητα. Αυτή η μορφή ήταν συμβατή με όλες τις μεθόδους ανίχνευσης ανωμαλιών που ακολούθησαν στο επόμενο κεφάλαιο και επέτρεψε την ομαλή σύγκριση διαφορετικών μοντέλων.



Σχήμα 3.4: Χάρτης συσχέτισης των 30 πρώτων TF-IDF χαρακτηριστικών.

Η εικόνα δείχνει τις συσχετίσεις ανάμεσα στα τριάντα πιο συχνά TF-IDF χαρακτηριστικά που εξήχθησαν από τα μηνύματα των αρχείων καταγραφής. Τα περισσότερα χαρακτηριστικά παρουσιάζουν πολύ χαμηλό βαθμό συσχέτισης μεταξύ τους, κάτι που σημαίνει ότι η μετατροπή TF-IDF παρήγαγε διακριτή και μη πλεονάζουσα πληροφορία.

Αυτό βοηθά στην ανάλυση, επειδή κάθε χαρακτηριστικό αντιπροσωπεύει διαφορετικό τμήμα της κειμενικής δομής των logs. Οι λίγες περιοχές με μέτρια συσχέτιση αντιστοιχούν σε λέξεις που εμφανίζονται μαζί στα ίδια είδη γεγονότων και υποδηλώνουν μικρές ομάδες λειτουργικών μοτίβων.

3.2.7 Έλεγχος σπανιότητας προτύπων και εντοπισμός ασυνήθιστων ακολουθιών

Τα περισσότερα γεγονότα ακολουθούν μια σταθερή ροή, όπου οι ίδιες ομάδες προτύπων εμφανίζονται με σχεδόν την ίδια σειρά. Η σταθερότητα αυτή επιτρέπει την ανίχνευση ασυνήθιστων ακολουθιών, δηλαδή εκείνων που διαφέρουν από τα συνηθισμένα πρότυπα λειτουργίας. Η διαδικασία αυτή δεν βασίζεται στο περιεχόμενο του κειμένου, αλλά στις σχέσεις ανάμεσα στα πρότυπα.

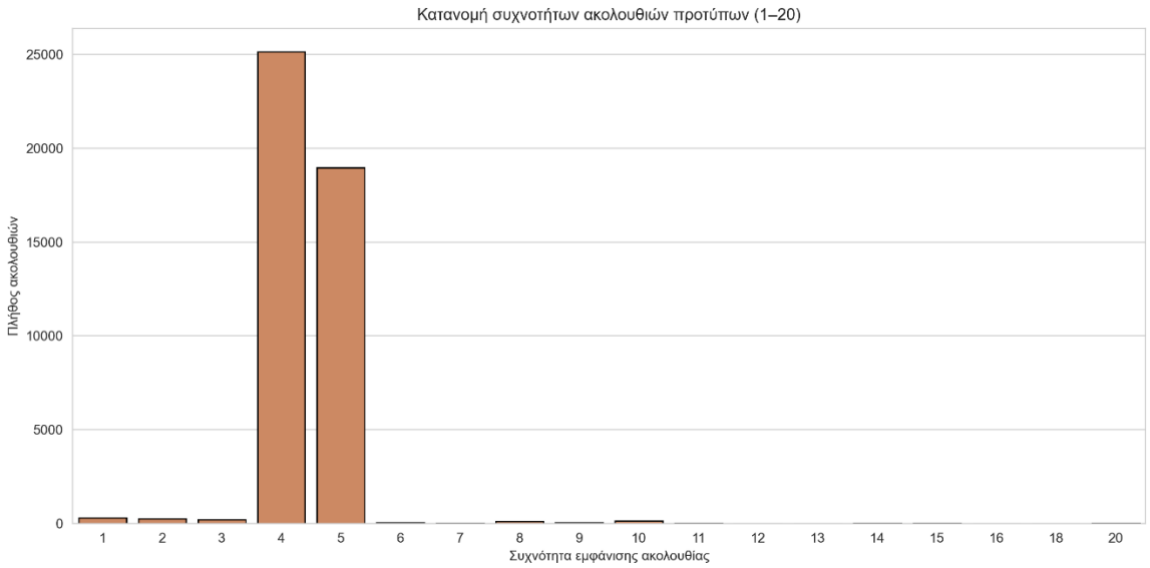
Κάθε μήνυμα μετατράπηκε σε πρότυπο, και τα πρότυπα αυτά μπήκαν στη σειρά με βάση τον χρόνο εμφάνισής τους. Από αυτό το σύνολο δημιουργήθηκαν ακολουθίες μήκους τριών προτύπων πράγμα το οποίο είναι αρκετό για να αποτυπώσει μικρές λειτουργικές μεταβάσεις, χωρίς να δημιουργεί υπερβολικό πλήθος πιθανών συνδυασμών. Με αυτό το βήμα προέκυψε ένα πλήθος ακολουθιών που κάλυπταν όλη τη λειτουργία του συστήματος. Οι ακολουθίες μετρήθηκαν για να φανεί πόσο συχνά εμφανίζονται. Οι περισσότερες εμφανίστηκαν πολλές φορές, γεγονός που επιβεβαίωσε ότι το σύστημα λειτουργεί με επαναλαμβανόμενο τρόπο και άλλες, πολύ λιγότερες, εμφανίστηκαν μόνο δύο ή τρεις φορές. Υπήρχαν και ακολουθίες που εμφανίστηκαν μία μόνο φορά. Αυτές οι ακολουθίες θεωρήθηκαν σπάνιες. Η σπανιότητα δεν σημαίνει απαραίτητα ότι υπάρχει σφάλμα αλλά δείχνει απλώς ότι το σύστημα απέκλινε από τα συνηθισμένα μοτίβα.

Με βάση αυτές τις συχνότητες δημιουργήθηκε ένας δείκτης σπανιότητας. Κάθε ακολουθία που εμφανίστηκε λιγότερες από τρεις φορές πήρε θετικό δείκτη. Ο δείκτης αυτός εφαρμόστηκε στο επίπεδο της γραμμής. Αν μια γραμμή συμμετείχε σε μια τέτοια ακολουθία, τότε η γραμμή χαρακτηριζόταν ως υποψήφια ανωμαλία. Επίσης, ο δείκτης αυτός δεν βασίζεται στο περιεχόμενο του μηνύματος αλλά στη θέση του μηνύματος σε σχέση με άλλα μηνύματα.

Τα logs συχνά δεν εμφανίζουν καθαρό error ή warning. Πολλές φορές το σύστημα κινείται σε πλαίσια που δεν θεωρούνται άμεσα σφάλματα, αλλά η σειρά των γεγονότων δεν

ταιριάζει με τις αναμενόμενες ροές. Η εξέταση των ακολουθιών εντοπίζει αυτή τη λεπτή απόκλιση. Η μέθοδος αναδεικνύει τη δομή των μεταβάσεων και δίνει μια πιο άμεση εικόνα για τη συμπεριφορά του συστήματος.

Ο δείκτης σπανιότητας δεν αντικατέστησε τους υπόλοιπους δείκτες αλλά συνδυάστηκε με το επίπεδο του μηνύματος και με τα αριθμητικά χαρακτηριστικά. Η σύνθεση αυτών των στοιχείων έδωσε ένα πιο σταθερό αποτέλεσμα και έκανε το σύστημα ανίχνευσης πιο ευαίσθητο σε λειτουργικές μεταβολές που δεν φαίνονται από ένα μόνο χαρακτηριστικό.



Σχήμα 3.5: Κατανομή συχνοτήτων εμφάνισης των ακολουθιών τριών προτύπων.

Η εικόνα παρουσιάζει την κατανομή της συχνότητας εμφάνισης των ακολουθιών τριών προτύπων σε ένα δείγμα 250.000 γραμμών. Το μεγαλύτερο μέρος των ακολουθιών εμφανίζεται τέσσερις ή πέντε φορές, γεγονός που δείχνει ότι το σύστημα παράγει σταθερές και επαναλαμβανόμενες ροές γεγονότων. Οι ακολουθίες με πολύ χαμηλή ή πολύ υψηλή συχνότητα είναι λίγες, κάτι που υποδεικνύει ότι οι περισσότερες μεταβάσεις ακολουθούν προβλέψιμο μοτίβο. Το μοτίβο αυτό αξιοποιείται στη διαδικασία ανίχνευσης ασυνήθιστων ακολουθιών, όπου η χαμηλή συχνότητα λειτουργεί ως ένδειξη απόκλισης από τη συνηθισμένη λειτουργία.

3.2.8 Τελικός πίνακας χαρακτηριστικών (Feature Matrix)

Η διαδικασία προετοιμασίας των δεδομένων κατέληξε σε έναν ολοκληρωμένο πίνακα χαρακτηριστικών, ο οποίος συγκεντρώνει όλα τα στοιχεία που εξήχθησαν από τις αρχικές γραμμές των logs. Ο πίνακας αυτός περιέχει τόσο αριθμητικά όσο και κατηγορικά πεδία, μαζί με τα στοιχεία που προέκυψαν από τον καθαρισμό, την ανάλυση προτύπων και την επεξεργασία της κειμενικής πληροφορίας. Η τελική μορφή του πίνακα δεν αποτελεί απλή μεταγραφή του αρχικού κειμένου. Είναι το αποτέλεσμα μιας διαδοχικής μετατροπής, στην οποία κάθε στάδιο πρόσθεσε ένα νέο επίπεδο πληροφορίας.

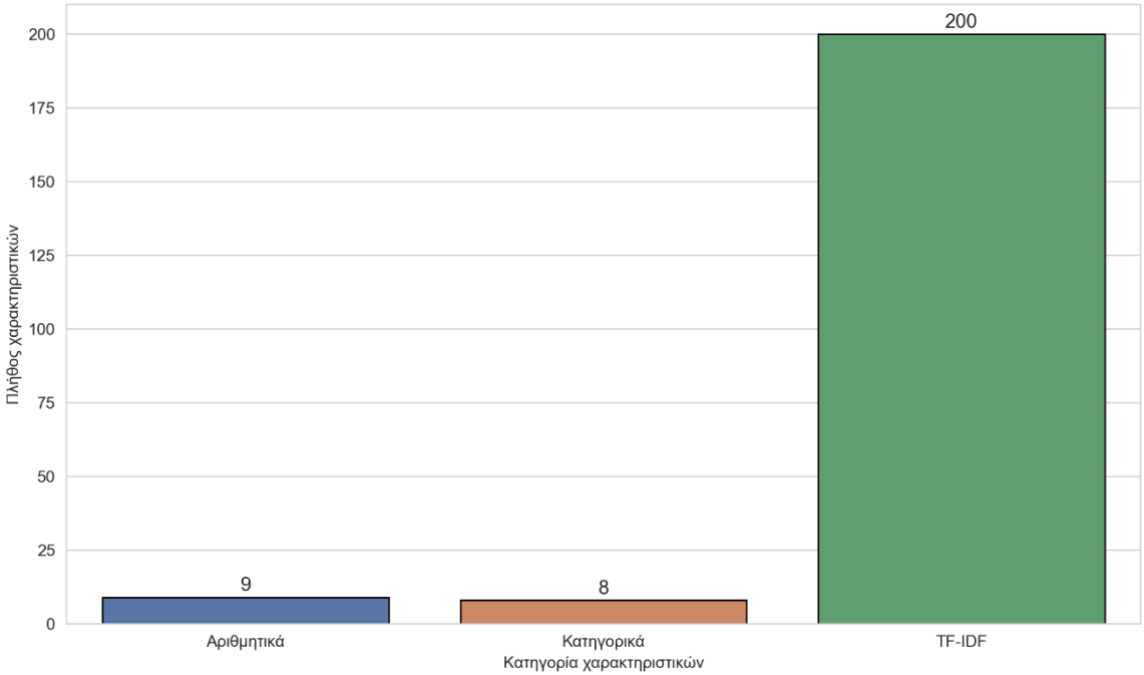
Ο πίνακας περιλαμβάνει τα βασικά πεδία που περιγράφουν κάθε εγγραφή. Τα πεδία αυτά είναι το όνομα του host, το αρχείο προέλευσης, το χρονικό στίγμα και το καθαρισμένο μήνυμα. Πάνω σε αυτά προστέθηκαν τα δομικά χαρακτηριστικά που αφορούν τη θέση της γραμμής στον χρόνο, όπως η ώρα και η ημέρα, μαζί με τον χρόνο που μεσολάβησε από το προηγούμενο γεγονός. Οι πληροφορίες αυτές αποδίδουν τη ρυθμικότητα των γεγονότων και επιτρέπουν τη διάκριση ανάμεσα στη συνηθισμένη λειτουργία και σε περιστασιακές μεταβολές.

Πέρα από τα απλά χρονικά και δομικά χαρακτηριστικά, ο πίνακας περιλαμβάνει στοιχεία που περιγράφουν την εσωτερική δομή των μηνυμάτων. Το μήκος του κειμένου, η συχνότητα των προτύπων και τα χαρακτηριστικά που προέκυψαν από τη μέθοδο TF-IDF αποτυπώνουν τη λεπτομέρεια και το περιεχόμενο των μηνυμάτων με συστηματικό τρόπο. Τα TF-IDF χαρακτηριστικά περιλαμβάνονται ως σταθερού μεγέθους διανύσματα, ενώ η συχνότητα των προτύπων αποτυπώνει τη θέση του κάθε μηνύματος στη συνολική συμπεριφορά του συστήματος.

Στον πίνακα έχουν προστεθεί και οι δείκτες που αφορούν την απόκλιση από τα συνηθισμένα μοτίβα. Ο δείκτης επιπέδου (ERROR, WARN, INFO, OTHER) μετατράπηκε σε αριθμητική μορφή, ενώ ο δείκτης σπανιότητας αποτυπώνει κατά πόσο μια γραμμή συμμετέχει σε μια σπάνια ακολουθία προτύπων. Οι δύο αυτοί δείκτες συνθέτουν μια εικόνα για το αν ένα μήνυμα βρίσκεται στο όριο της λειτουργικής κανονικότητας. Οι δείκτες συνδυάζονται με τα υπόλοιπα χαρακτηριστικά για να σχηματίσουν το πεδίο is_anomaly, το οποίο χρησιμοποιείται στα επόμενα στάδια της ανάλυσης.

Η τελική έκδοση του πίνακα χαρακτηρίζεται από μεγάλο πλήθος στηλών αφού το σύνολο των χαρακτηριστικών ξεπερνά τις διακόσιες στήλες, λόγω της παρουσίας των TF-IDF πεδίων. Παρόλα αυτά, ο πίνακας διατηρεί σταθερή δομή και μπορεί να αξιοποιηθεί από

οποιοδήποτε μοντέλο ανίχνευσης ανωμαλιών. Το πλήθος των γραμμών, μετά τη διαδικασία καθαρισμού και τον έλεγχο εγκυρότητας των timestamps, φτάνει περίπου τις 607.000. Το μεγάλο αυτό πλήθος επιτρέπει την εκπαίδευση μοντέλων που βασίζονται τόσο στην πυκνότητα όσο και στη συμπεριφορά των χαρακτηριστικών.



Σχήμα 3.6: Κατανομή των χαρακτηριστικών του τελικού πίνακα δεδομένων.

Το παραπάνω σχήμα συνοψίζει τις κατηγορίες των χαρακτηριστικών που περιέχονται στον τελικό πίνακα δεδομένων. Τα αριθμητικά πεδία περιλαμβάνουν στοιχεία που αφορούν τον χρόνο, τη δομή των μηνυμάτων και τους δείκτες λειτουργικής συμπεριφοράς. Τα κατηγορικά πεδία διατηρούν την περιγραφική πληροφορία, ενώ τα TF-IDF χαρακτηριστικά αποτυπώνουν την κειμενική πληροφορία σε αριθμητική μορφή. Ο συνδυασμός αυτών των τριών κατηγοριών δημιουργεί έναν σταθερό και πλήρη πίνακα εισόδου για τα μοντέλα ανίχνευσης ανωμαλιών.

Ο πίνακας χαρακτηριστικών αποτελεί το σημείο αφετηρίας για το υπόλοιπο της ανάλυσης. Μετά την ολοκλήρωση του μετασχηματισμού, κάθε γραμμή του συστήματος περιγράφεται με τρόπο συγκρίσιμο και συνεπή. Το αποτέλεσμα είναι ένα σύνολο δεδομένων που διατηρεί την πληροφορία του αρχικού κειμένου αλλά είναι κατάλληλο για μηχανική μάθηση. Στο επόμενο τμήμα αναλύεται η επιλογή των κατάλληλων μοντέλων και ο τρόπος με τον οποίο αυτά μπορούν να αξιοποιήσουν τον τελικό πίνακα για την ανίχνευση των ανωμαλιών.

3.3 Περιγραφή μοντέλων ανίχνευσης ανωμαλιών

Στην ενότητα αυτή παρουσιάζονται τα μοντέλα ανίχνευσης ανωμαλιών που χρησιμοποιήθηκαν πάνω στα δεδομένα καταγραφής του vSphere. Στο προηγούμενο υποκεφάλαιο ολοκληρώθηκε ο καθαρισμός και η μετατροπή των logs σε δομημένη μορφή, με πρότυπα, χαρακτηριστικά και τελικό πίνακα εισόδου. Εδώ παίρνουμε αυτόν τον πίνακα και τον δίνουμε σε διαφορετικούς αλγορίθμους, που προσπαθούν να ξεχωρίσουν τις φυσιολογικές εγγραφές από εκείνες που φαίνεται να ξεφεύγουν από τη συνήθη συμπεριφορά του συστήματος.

Το πρόβλημα είναι καθαρά μη επιβλεπόμενο αφού δεν υπάρχουν ετικέτες που να λένε ποια γραμμή θεωρείτε ανωμαλία και ποια όχι. Στην πράξη υποθέτουμε ότι η μεγάλη πλειονότητα των καταγραφών είναι κανονική λειτουργία και ότι τα πραγματικά προβληματικά γεγονότα είναι λίγα και σπάνια. Πάνω σε αυτή την παραδοχή επιλέχθηκαν τρεις διαφορετικές οικογένειες μοντέλων, ένα μοντέλο βασισμένο σε σύνολο τυχαίων δέντρων (Isolation Forest), ένα μοντέλο οριοθέτησης της normal περιοχής στον χώρο των χαρακτηριστικών (One-Class SVM) και ένα νευρωνικό δίκτυο ακολουθιών τύπου GRU, που προσπαθεί να μάθει την τυπική σειρά των γεγονότων.

Το Isolation Forest εστιάζει στο πόσο εύκολα απομονώνεται μια εγγραφή από τις υπόλοιπες. Το One-Class SVM προσπαθεί να χαράξει ένα σύνορο γύρω από την κανονική συμπεριφορά ενώ το νευρωνικό δίκτυο εστιάζει στη χρονική ροή των προτύπων μέσα στα logs. Δεν υπάρχει ένα και μοναδικό σωστό εργαλείο. Για αυτό η εργασία δεν περιορίζεται σε έναν αλγόριθμο, αλλά συνδυάζει περισσότερα μοντέλα και στη συνέχεια συγκρίνει τα αποτελέσματά τους, τόσο ποσοτικά όσο και ποιοτικά, στο επόμενο κεφάλαιο.

3.3.1 Κριτήρια επιλογής αλγορίθμων

Το πρόβλημα της ανίχνευσης ανωμαλιών στα logs του vSphere έχει συγκεκριμένα πρακτικά και θεωρητικά χαρακτηριστικά, τα οποία βάζουν όρια στο τι έχει νόημα να χρησιμοποιηθεί. Πρώτο κριτήριο είναι η έλλειψη ετικετών. Στα διαθέσιμα δεδομένα δεν υπάρχει στήλη που να λέει ποιες γραμμές αντιστοιχούν σε ανωμαλία και ποιες σε κανονική λειτουργία. Αυτό σημαίνει ότι τα κλασικά supervised μοντέλα, όπως logistic regression δεν μπορούν να χρησιμοποιηθούν. Χρειαζόμαστε μεθόδους που μαθαίνουν από τα ίδια τα δεδομένα, χωρίς ground truth, και προσπαθούν να ορίσουν τι θεωρείται σύνηθες και τι θεωρείται σπάνιο.

Δεύτερο κριτήριο είναι ο όγκος και η φύση των logs. Τα αρχεία καταγραφής του vSphere δεν είναι μερικές εκατοντάδες γραμμές αλλά πρόκειται για μεγάλο αριθμό εγγραφών, με διαφορετικούς hosts, πολλά αρχεία, επαναλαμβανόμενα templates και πλήθος παραμέτρων. Ένα μοντέλο που δεν κλιμακώνεται καλά, θα κολλήσει απλά στο training. Χρειάζονται τεχνικές που αντέχουν πολλά δείγματα και μεγάλο αριθμό χαρακτηριστικών, χωρίς υπερβολικό χρόνο εκπαίδευσης. Ο Isolation Forest ταιριάζει, γιατί βασίζεται σε τυχαία δέντρα, ενώ ο One Class SVM χρησιμοποιείται πάνω σε μειωμένη αναπαράσταση (με SVD) για να περιοριστεί το κόστος.

Τρίτο κριτήριο είναι η ετερογένεια των χαρακτηριστικών ενώ ακόμη ένα τέταρτο κριτήριο είναι η δυνατότητα ερμηνείας του αποτελέσματος. Στόχος δεν είναι να βγει απλά ένας αριθμός, αλλά ένα σύνολο από γραμμές logs που μπορεί να διαβάσει άνθρωπος. Ένας διαχειριστής θα θέλει να δει ποια συγκεκριμένα μηνύματα εμφανίζονται ως ύποπτα. Για αυτό προτιμώνται μοντέλα που παράγουν:

- σαφές binary flag για ανωμαλία
- συνεχές score ανωμαλίας, ώστε να γίνει ταξινόμηση

Τέταρτο κριτήριο είναι η κάλυψη διαφορετικών οπτικών πάνω στα ίδια δεδομένα. Η συμπεριφορά του συστήματος δεν είναι μόνο θέμα πού βρίσκεται το σημείο στον χώρο των features, είναι και θέμα σειράς που συμβαίνει ένα γεγονός. Αν μια ακολουθία γεγονότων είναι εντελώς παράξενη χρονικά, αυτό δεν φαίνεται πάντα σε μια στατική αναπαράσταση. Για αυτό η εργασία δεν περιορίζεται σε ένα μόνο είδος μοντέλου.

- Ο Isolation Forest κοιτάζει κάθε γραμμή ξεχωριστά σε έναν πολυδιάστατο χώρο και μετρά πόσο εύκολα απομονώνεται.
- Ο One Class SVM προσπαθεί να χαράξει ένα λεπτό σύνορο γύρω από τα κανονικά σημεία, σε μειωμένη διάσταση.
- Το νευρωνικό με GRU κοιτάζει την ακολουθία των templates και μαθαίνει ποια σειρά γεγονότων είναι τυπική και ποια σπάνια.

Πέμπτο κριτήριο, είναι οι υπολογιστικοί πόροι και η υλοποίηση. Όλα τα μοντέλα έπρεπε να υλοποιηθούν σε λογικό χρόνο, σε περιβάλλον Python, με βιβλιοθήκες όπως scikit learn και

TensorFlow, χωρίς πρόσβαση σε εξειδικευμένο cluster. Αυτό αποκλείει πιο βαριές λύσεις που απαιτούν χρονοβόρα εκπαίδευση ή πολύπλοκο tuning για να αποδώσουν.
Συνδυάζοντας τα παραπάνω, χρειαζόταν ένα μοντέλο γρήγορο και ανθεκτικό σε μεγάλο όγκο logs, ένα μοντέλο που χαράζει όρια γύρω από την κανονικότητα και ένα μοντέλο ακολουθιών που βλέπει τη ροή των γεγονότων. Μέσα σε αυτά τα όρια επιλέχθηκαν ο Isolation Forest, ο One Class SVM πάνω σε μειωμένη αναπαράσταση και το νευρωνικό GRU, που μαζί καλύπτουν το βασικό φάσμα αναγκών της εργασίας.

3.3.2 Isolation Forest

Στο στάδιο αυτό εφαρμόζεται ο αλγόριθμος Isolation Forest πάνω στο τελικό σύνολο χαρακτηριστικών που προέκυψε από την επεξεργασία των αρχείων καταγραφής. Σκοπός είναι να παραχθεί μια λίστα εγγραφών που ξεφεύγουν από την τυπική συμπεριφορά του συστήματος. Ο συγκεκριμένος αλγόριθμος αντιμετωπίζει κάθε γραμμή log ως σημείο σε πολυδιάστατο χώρο. Δεν αξιοποιεί το ακατέργαστο κείμενο, αλλά ένα σύνολο από αριθμητικά και κατηγορικά χαρακτηριστικά που συμπυκνώνουν τις βασικές ιδιότητες του μηνύματος: πόσο συχνά εμφανίζεται, πότε, σε ποιον host, σε ποιο αρχείο και με ποιο βάρος σε παραμέτρους όπως IP και HEX τιμές.

Επιλογή και προετοιμασία χαρακτηριστικών

Η είσοδος του μοντέλου είναι το αρχείο CSV, όπου κάθε log εγγραφή έχει ήδη μετατραπεί σε δομημένη μορφή με τη διαδικασία που αναφέραμε στο προηγούμενο κεφάλαιο. Από αυτό το αρχείο επιλέγονται τα εξής πεδία:

- template_freq: πλήθος εμφανίσεων του συγκεκριμένου template στο σύνολο των δεδομένων,
- hour: ώρα καταγραφής του γεγονότος,
- day_of_week: ημέρα της εβδομάδας,
- msg_len: μήκος του καθαρισμένου μηνύματος,
- param_IP, param_HEX, param_NUM: αριθμός διευθύνσεων IP, δεκαεξαδικών tokens και μεγάλων αριθμών στο κείμενο,
- host: ταυτότητα του host,
- file: αρχείο καταγραφής από το οποίο προήλθε η γραμμή,

- template_id: σταθερό αναγνωριστικό του προτύπου,
- log_level: επίπεδο σοβαρότητας, όπως INFO, WARN ή ERROR

Τα χαρακτηριστικά χωρίζονται σε δύο ομάδες. Η πρώτη ομάδα είναι τα αριθμητικά πεδία, δηλαδή τα template_freq, hour, day_of_week, msg_len, param_IP, param_HEX και param_NUM. Για αυτή την ομάδα προηγείται αντικατάσταση των κενών τιμών με μηδενικά και στη συνέχεια εφαρμόζεται κλιμάκωση με StandardScaler, ώστε όλες οι μεταβλητές να βρίσκονται σε παρόμοια κλίμακα. Η δεύτερη ομάδα αφορά τα κατηγορικά πεδία, δηλαδή τα host, file, template_id και log_level. Σε αυτά εφαρμόζεται One-Hot Encoding, ώστε κάθε διαφορετική τιμή να μετατρέπεται σε ξεχωριστή δυαδική στήλη.

Εκπαίδευση και ρυθμίσεις του μοντέλου

Η υλοποίηση βασίζεται στην κλάση IsolationForest της scikit-learn, με τις ακόλουθες ρυθμίσεις:

- n_estimators = 300
- contamination = 0.01
- max_samples = 0.8
- random_state = 42
- n_jobs = -1

Ο αριθμός των δέντρων ορίστηκε σε 300 ώστε τα αποτελέσματα να είναι σταθερά χωρίς υπερβολικό χρόνο εκπαίδευσης. Η παράμετρος contamination = 0.01 εκφράζει την υπόθεση ότι περίπου 1% των εγγραφών είναι ανώμαλες. Με βάση αυτή την τιμή το μοντέλο τοποθετεί το κατώφλι που χωρίζει τις φυσιολογικές παρατηρήσεις από τις ύποπτες. Η επιλογή max_samples = 0.8 σημαίνει ότι κάθε δέντρο εκπαιδεύεται σε τυχαίο υποσύνολο του 80 % των διαθέσιμων δειγμάτων, κάτι που αυξάνει την ποικιλία των δέντρων και αναδεικνύει καλύτερα τα ακραία σημεία.
Στόχος είναι να χαρτογραφηθεί η περιοχή όπου συγκεντρώνεται η κανονική συμπεριφορά του συστήματος και να εντοπιστούν τα σημεία που βρίσκονται εκτός αυτής της περιοχής.

Μετά την εκπαίδευση, για κάθε log εγγραφή υπολογίζονται δύο τιμές:

- **if_flag**: παίρνει τιμή 1 όταν η εγγραφή θεωρείται κανονική και −1 όταν θεωρείται ανωμαλία.
- **if_score**: αριθμητικό score από τη συνάρτηση απόφασης του αλγορίθμου. Όσο μικρότερη είναι η τιμή, τόσο πιο «απομονωμένο» θεωρείται το σημείο.

Επεξεργασία scores και δημιουργία λίστας ανωμαλιών

Αν κρατιόνταν όλες οι εγγραφές με τιμή if_flag = -1, το αποτέλεσμα θα ήταν ένας πολύ μεγάλος όγκος alerts με πολλές επαναλήψεις, που δεν θα βοηθούσε έναν αναλυτή. Για να προκύψει μια πραγματικά χρήσιμη και διαχειρίσιμη λίστα, εφαρμόζεται μια σειρά από φίλτρα και βήματα ομαδοποίησης.

Αρχικά διατηρούνται μόνο οι εγγραφές όπου το μοντέλο έχει δώσει if_flag = -1. Αυτές αποτελούν το αρχικό σύνολο υποψήφιων ανωμαλιών. Από αυτό το σύνολο αφαιρούνται όλες οι γραμμές με template_freq μεγαλύτερο από 20. Ένα μήνυμα που εμφανίζεται δεκάδες φορές σε όλο το dataset δύσκολα θεωρείται σπάνιο γεγονός, ακόμη και αν μια μεμονωμένη εμφάνιση έχει οριακό score. Με αυτό το φιλτράρισμα παραμένουν κυρίως templates που δεν κυριαρχούν στα δεδομένα.

Στη συνέχεια χρειάζεται αντιμετώπιση της επανάληψης παρόμοιων καταγραφών. Για αυτό κατασκευάστηκε ένα κλειδί ομαδοποίησης pattern_key, το οποίο προκύπτει από τον συνδυασμό του file, του log_level και των πρώτων 80 χαρακτήρων του template. Όσες εγγραφές μοιράζονται το ίδιο pattern_key θεωρούνται μέλη της ίδιας κατηγορίας γεγονότων. Από κάθε τέτοια κατηγορία επιλέγεται μόνο μία εγγραφή, αυτή με το μικρότερο if_score. Έτσι για κάθε μοτίβο παραμένει η πιο ακραία γραμμή, αυτή που το μοντέλο θεωρεί πιο απομονωμένη από τη μάζα των υπόλοιπων.

Οι εγγραφές που απομένουν ταξινομούνται σε αύξουσα σειρά με βάση το if_score. Από αυτή τη σειρά επιλέγονται οι εκατό εγγραφές με τα χαμηλότερα scores και αποθηκεύονται στο ξεχωριστό αρχείο CSV. Το αρχείο αυτό αποτελεί το βασικό σύνολο εξόδου του Isolation Forest και χρησιμοποιείται στο επόμενο κεφάλαιο για ποιοτική αξιολόγηση. Με αυτό το σύνολο βημάτων ο Isolation Forest προσφέρει μια λίστα καταγραφών που συνδυάζουν σπανιότητα, ασυνήθιστο συνδυασμό χαρακτηριστικών και υψηλό βαθμό απομόνωσης σε σχέση με το συνολικό σύνολο των logs.

3.3.3 One-Class SVM με μείωση διαστάσεων

Στο επόμενο βήμα χρησιμοποιείται ο One-Class SVM πάνω στα ίδια δεδομένα. Το μοντέλο προσπαθεί να χαράξει ένα όριο γύρω από την περιοχή όπου συγκεντρώνεται η κανονική συμπεριφορά και να ξεχωρίσει όσα σημεία πέφτουν έξω από αυτό το όριο. Ο στόχος είναι πάλι εντοπιστεί ένα σύνολο εγγραφών που διαφέρουν καθαρά από τη μάζα των υπολοίπων.

Μείωση διαστάσεων και προετοιμασία δεδομένων

Ο πίνακας χαρακτηριστικών είναι υψηλής διάστασης. Αυτή η αναπαράσταση είναι βολική για δέντρα, αλλά δυσκολεύει τον SVM. Για να γίνει το πρόβλημα πιο διαχειρίσιμο, πριν από τον One-Class SVM εφαρμόζεται μείωση διαστάσεων με Truncated SVD. Τα αριθμητικά χαρακτηριστικά, όπως template_freq, hour, day_of_week, msg_len, param_IP, param_HEX και param_NUM, καθαρίζονται από κενές τιμές και κανονικοποιούνται ενώ τα κατηγορικά χαρακτηριστικά host, file, template_id και log_level κωδικοποιούνται με One-Hot Encoding. Από αυτή τη διαδικασία προκύπτει ξανά ο πίνακας χαρακτηριστικών.
Πάνω σε αυτό τον πίνακα εφαρμόζεται TruncatedSVD με 50 συνιστώσες. Κάθε γραμμή log, που αρχικά περιγραφόταν από πολλά δεκάδες ή και εκατοντάδες χαρακτηριστικά, συμπυκνώνεται σε ένα διάνυσμα 50 διαστάσεων. Η συμπίεση αυτή δεν κρατά όλες τις λεπτομέρειες αλλά κρατά όμως το ισχυρότερο σήμα του dataset και απομακρύνει μέρος του θορύβου, έτσι μειώνεται το κόστος και αποφεύγονται αριθμητικές δυσκολίες που θα εμφανίζονταν σε έναν πολύ πιο μεγάλο χώρο.

Εκπαίδευση και ρυθμίσεις του One-Class SVM

Στην υλοποίηση χρησιμοποιείται γραμμικός kernel. Μετά τη μείωση διαστάσεων τα δεδομένα βρίσκονται ήδη σε έναν πιο συμπαγή χώρο, όπου ένα γραμμικό όριο αρκεί για να ξεχωρίσει τον πυρήνα της κανονικότητας από τα πιο απομακρυσμένα σημεία. Ταυτόχρονα ο γραμμικός kernel είναι σαφώς πιο γρήγορος σε σχέση με τους πιο σύνθετους πυρήνες. Με απλά λόγια, το μοντέλο επιτρέπει περίπου το 1% των σημείων να βρεθούν έξω από την κανονική περιοχή. Αν προσπαθήσει να χωρέσει όλα τα σημεία, δεν θα μπορέσει να ξεχωρίσει τίποτα ως ανωμαλία.
Το ζητούμενο είναι να βρεθεί ένα σχετικά μικρό κέλυφος που να περικλείει τη μεγάλη πλειονότητα των παρατηρήσεων. Μετά την εκπαίδευση, για κάθε log εγγραφή υπολογίζονται δύο ποσά, ο δείκτης ocsvm_flag και το score ocsvm_score. Ο δείκτης παίρνει

τιμή 1 όταν το σημείο θεωρείται κανονικό και τιμή −1 όταν βρίσκεται εκτός του ορίου. Το score δείχνει πόσο μέσα ή πόσο έξω βρίσκεται η εγγραφή σε σχέση με το όριο που χάραξε ο αλγόριθμος. Όσο πιο μικρό είναι το ocsvm_score, τόσο πιο ύποπτο θεωρείται το σημείο. Τα αποτελέσματα αποθηκεύονται σε νέο αρχείο CSV, όπου κάθε γραμμή συνδυάζει το αρχικό μήνυμα, τα χαρακτηριστικά του και τα scores του SVM.

Επεξεργασία scores και δημιουργία λίστας ανωμαλιών

Όπως και στον Isolation Forest, τα raw scores από μόνα τους δεν αρκούν. Αν κρατηθούν όλες οι εγγραφές με ocsvm_flag ίσο με −1, η λίστα θα είναι μεγάλη και γεμάτη επαναλήψεις. Για να αποκτήσει πρακτική αξία το αποτέλεσμα, εφαρμόζονται αντίστοιχα βήματα φιλτραρίσματος και ομαδοποίησης.

Πρώτα διατηρούνται μόνο οι εγγραφές με ocsvm_flag ίσο με −1. Αυτές αποτελούν το αρχικό σύνολο των υποψήφιων ανωμαλιών που εντόπισε ο One-Class SVM. Από αυτό το σύνολο αφαιρούνται οι γραμμές με template_freq μεγαλύτερο από 20, για τον ίδιο λόγο που εφαρμόστηκε και στον Isolation Forest. Ένα template που επαναλαμβάνεται δεκάδες φορές σπανίως είναι πραγματικά σπάνιο γεγονός, ακόμη και αν σε μία συγκεκριμένη εμφάνιση το μοντέλο το βλέπει οριακά εκτός ορίου.

Στη συνέχεια κατασκευάζεται ξανά ένα κλειδί pattern_key, το οποίο προκύπτει από τον συνδυασμό του file, του log_level και των πρώτων 80 χαρακτήρων του template. Όσες εγγραφές έχουν ίδιο pattern_key αντιμετωπίζονται ως παραλλαγές του ίδιου τύπου γεγονότος. Από κάθε τέτοια ομάδα επιλέγεται η εγγραφή με το μικρότερο ocsvm_score. Έτσι για κάθε μοτίβο μένει μόνο η πιο ακραία εκδοχή, αυτή που το μοντέλο θεωρεί πιο μακριά από την περιοχή κανονικότητας.

Τέλος, οι εγγραφές που απομένουν ταξινομούνται σε αύξουσα σειρά ως προς το ocsvm_score. Από αυτή τη σειρά επιλέγονται πάλι οι εκατό πρώτες γραμμές με τα χαμηλότερα scores και αποθηκεύονται σε αρχείο CSV. Αυτό το αρχείο αποτελεί το συμπυκνωμένο αποτέλεσμα του One-Class SVM και χρησιμοποιείται στο επόμενο κεφάλαιο, όπου τα ευρήματα συγκρίνονται με τις ανωμαλίες που προέκυψαν από τον Isolation Forest και το νευρωνικό δίκτυο.

Με αυτή τη διαδικασία ο One-Class SVM λειτουργεί ως δεύτερη, ανεξάρτητη γνώμη πάνω στα ίδια δεδομένα. Δεν βασίζεται σε τυχαία δέντρα αλλά σε ένα όριο στον χώρο των συνιστωσών του SVD. Όπου και τα δύο μοντέλα συμφωνούν ότι μια γραμμή είναι

ανώμαλη, η πιθανότητα να πρόκειται για πραγματικά ενδιαφέρον γεγονός αυξάνεται. Εκεί ακριβώς στρέφεται και η προσοχή στην ανάλυση του επόμενου κεφαλαίου.

3.3.4 Νευρωνικό μοντέλο ακολουθιών με GRU

Το τρίτο μοντέλο που χρησιμοποιείται στην εργασία είναι ένα νευρωνικό δίκτυο ακολουθιών με μονάδες GRU. Σε αντίθεση με τον Isolation Forest και τον One-Class SVM, που βλέπουν κάθε γραμμή log ως ανεξάρτητο σημείο στον χώρο των χαρακτηριστικών, το συγκεκριμένο μοντέλο κοιτάζει τη σειρά με την οποία εμφανίζονται τα πρότυπα μηνυμάτων. Στόχος είναι να μάθει ποια ακολουθία προτύπων θεωρείται φυσιολογική και να σημάνει συναγερμό όταν η πραγματική συνέχεια των γεγονότων αποκλίνει έντονα από αυτό που περιμένει.

Αναπαράσταση των logs ως ακολουθίες προτύπων

Η είσοδος του νευρωνικού είναι τα πρότυπα των μηνυμάτων. Αρχικά τα logs ταξινομούνται χρονικά. Αν υπάρχει στήλη host, η ταξινόμηση γίνεται ανά host και στη συνέχεια ανά timestamp, ώστε να διατηρηθεί η φυσική ροή των γεγονότων σε κάθε μηχάνημα. Στη συνέχεια αφαιρούνται τυχόν κενά templates, ώστε κάθε γραμμή να αντιστοιχεί σε ένα καθαρό πρότυπο.

Κάθε διαφορετικό template μετατρέπεται σε ακέραιο κωδικό μέσω LabelEncoder. Με αυτόν τον τρόπο, η στήλη template αντικαθίσταται από μια στήλη event_id, όπου κάθε τιμή είναι ένας αριθμός από το μηδέν μέχρι τον συνολικό αριθμό των διαφορετικών προτύπων. Το μέγεθος αυτού του λεξιλογίου αποθηκεύεται ως vocab_size και χρησιμοποιείται αργότερα στο embedding layer του μοντέλου.

Με βάση τη σειρά των event_id δημιουργούνται ακολουθίες σταθερού μήκους. Στην υλοποίηση χρησιμοποιείται μήκος πέντε γεγονότων (SEQUENCE_LENGTH = 5). Για κάθε θέση i στο dataset, σχηματίζεται ένα διάνυσμα εισόδου με τα πέντε προηγούμενα γεγονότα και ως στόχος τίθεται το έκτο, δηλαδή το γεγονός στη θέση i + 5. Έτσι προκύπτουν ζεύγη (X, y), όπου το X περιέχει σειρές από πέντε event_id και το y περιέχει το επόμενο event_id που ακολουθεί στην πραγματικότητα.

Η διαίρεση σε train και test γίνεται με βάση τον χρόνο. Περίπου το 80% των ακολουθιών χρησιμοποιείται για εκπαίδευση και το υπόλοιπο 20% για έλεγχο. Αυτή η επιλογή αποφεύγει να μπερδευτούν στο ίδιο σύνολο παλαιότερα και νεότερα γεγονότα και

αντικατοπτρίζει καλύτερα τη ρεαλιστική χρήση του μοντέλου, όπου το δίκτυο εκπαιδεύεται στο παρελθόν και καλείται να αξιολογήσει νέα δεδομένα.

Αρχιτεκτονική του νευρωνικού μοντέλου

Η αρχιτεκτονική του μοντέλου είναι απλή αλλά προσαρμοσμένη στο πρόβλημα πρόβλεψης του επόμενου event. Το δίκτυο αποτελείται από τρία βασικά επίπεδα.

- Στο πρώτο επίπεδο χρησιμοποιείται ένα Embedding layer. Η είσοδος στο δίκτυο είναι ακολουθίες από ακέραιους event_id, που δεν έχουν καμία άμεση γεωμετρική σημασία. Το Embedding μετατρέπει κάθε event_id σε ένα πυκνό διάνυσμα μικρής διάστασης. Στην υλοποίηση, το embedding έχει μέγεθος 16. Έτσι κάθε διαφορετικό template αναπαρίσταται ως σημείο σε δεκαεξαδιάστατο χώρο, όπου το δίκτυο μπορεί να μάθει ομοιότητες ή διαφορές μεταξύ προτύπων.
- Στο δεύτερο επίπεδο βρίσκεται ένα GRU layer με 32 units. Το GRU είναι τύπος επαναλαμβανόμενου νευρώνα, σχεδιασμένος να χειρίζεται ακολουθίες. Διαβάζει βήμα-βήμα τα embeddings των 5 γεγονότων και διατηρεί μια εσωτερική κατάσταση που συνοψίζει την πληροφορία της ακολουθίας. Στο τέλος του βήματος παράγει ένα διάνυσμα σταθερού μεγέθους, που λειτουργεί ως συμπυκνωμένη αναπαράσταση της πρόσφατης ιστορίας των logs.
- Στο τρίτο επίπεδο, το δίκτυο χρησιμοποιεί ένα πλήρως συνδεδεμένο dense layer με vocab_size εξόδους και συνάρτηση ενεργοποίησης softmax. Κάθε έξοδος αντιστοιχεί σε ένα πιθανό επόμενο template. Το softmax επιστρέφει μια κατανομή πιθανοτήτων πάνω σε όλα τα δυνατά templates. Η υψηλότερη πιθανότητα δείχνει ποιο event θεωρεί το μοντέλο πιο πιθανό ως συνέχεια της ακολουθίας που είδε στην είσοδο.

Για την εκπαίδευση χρησιμοποιείται συνάρτηση κόστους sparse categorical cross-entropy και Adam optimizer. Το δίκτυο εκπαιδεύεται για 26 epochs, με μέγεθος batch 512 και με χρήση EarlyStopping. Ο μηχανισμός EarlyStopping σταματά την εκπαίδευση αν δεν βελτιωθεί η απόδοση στο validation set για δύο συνεχόμενες εποχές και επαναφέρει τα καλύτερα βάρη που βρέθηκαν. Έτσι αποφεύγεται το overfitting και μειώνεται ο συνολικός χρόνος εκπαίδευσης.

Ορισμός ανωμαλίας με βάση την πρόβλεψη επόμενου γεγονότος

Μετά την εκπαίδευση, το μοντέλο χρησιμοποιείται ως εργαλείο ανίχνευσης ανωμαλιών πάνω στο test set. Η βασική ιδέα είναι η εξής: αν η πραγματική συνέχεια μιας ακολουθίας δεν ανήκει στις προβλέψεις υψηλής πιθανότητας του δικτύου, τότε κάτι ασυνήθιστο συνέβη.

Για κάθε δείγμα του test set, το μοντέλο υπολογίζει την κατανομή πιθανοτήτων πάνω σε όλα τα templates. Από αυτή την κατανομή κρατούνται οι κορυφαίες τρεις προβλέψεις (TOP_K = 3). Αν το πραγματικό επόμενο event_id δεν βρίσκεται μέσα σε αυτή την τριάδα, η συγκεκριμένη χρονική στιγμή χαρακτηρίζεται ως ανωμαλία. Παράλληλα καταγράφεται και η πιθανότητα που έδωσε το μοντέλο στο πραγματικό event. Όσο χαμηλότερη είναι αυτή η πιθανότητα, τόσο πιο απίθανη θεώρησε το δίκτυο τη συνέχεια που παρατηρήθηκε.

Τα αποτελέσματα αυτά ευθυγραμμίζονται ξανά με το αρχικό dataframe των logs. Οι ανωμαλίες αντιστοιχίζονται στις πραγματικές γραμμές καταγραφής, με βάση τη χρονική θέση της κάθε ακολουθίας. Έτσι για κάθε ύποπτη περίπτωση υπάρχουν όλα τα πεδία: host, file, timestamp, template, raw_line, καθώς και δύο επιπλέον στήλες, μία που δείχνει αν το σημείο χαρακτηρίστηκε ως ανωμαλία από το νευρωνικό (nn_is_anomaly) και μία που δίνει την πιθανότητα του πραγματικού event (nn_prob_true_event).

Στην τελική φάση, όλες οι εγγραφές που το μοντέλο χαρακτήρισε ως ανωμαλίες ταξινομούνται με βάση την πιθανότητα nn_prob_true_event σε αύξουσα σειρά. Όσες έχουν τη χαμηλότερη πιθανότητα θεωρούνται πιο ύποπτες, αφού το δίκτυο τις θεωρεί σχεδόν αδύνατη συνέχεια της ακολουθίας που προηγήθηκε. Το πλήρες σύνολο αποθηκεύεται σε αρχείο CSV, ενώ οι εκατό πιο ακραίες εγγραφές αποθηκεύονται σε διαφορετικό αρχείο. Αυτές οι εκατό γραμμές αποτελούν την κύρια λίστα ανωμαλιών που προκύπτει από το νευρωνικό μοντέλο και χρησιμοποιούνται στο επόμενο κεφάλαιο για σύγκριση με τα αποτελέσματα του Isolation Forest και του One-Class SVM.

Με αυτή την προσέγγιση, το νευρωνικό με GRU συμπληρώνει τα άλλα δύο μοντέλα. Δεν βασίζεται σε στατικά χαρακτηριστικά αλλά στη χρονική ροή των γεγονότων. Εκεί όπου ο Isolation Forest και ο One-Class SVM βλέπουν απλώς ένα σημείο στον χώρο των features, το GRU βλέπει μια μικρή ιστορία πέντε γεγονότων και κρίνει αν η συνέχεια της ιστορίας αυτής φαίνεται φυσιολογική ή όχι.

4 Πειραματική Αξιολόγηση

Η αξιολόγηση των μοντέλων έγινε πάνω στο ίδιο σύνολο δεδομένων που προέκυψε από τα αρχεία καταγραφής του vSphere, μετά από όλα τα στάδια καθαρισμού και δημιουργίας χαρακτηριστικών του Κεφαλαίου 3. Το dataset περιλαμβάνει όλα τα logs που συλλέχθηκαν από τον ESXi host esx-vshere3.kos.gr, ενοποιημένα από 13 διαφορετικά αρχεία καταγραφής και μετασχηματισμένα σε ενιαίο αρχείο CSV. Κάθε γραμμή αντιστοιχεί σε ένα γεγονός του συστήματος και συνοδεύεται από host, αρχείο προέλευσης, χρονική σήμανση, πρότυπο μηνύματος και τα αριθμητικά χαρακτηριστικά που δημιουργήθηκαν στο στάδιο του feature engineering.

Ο Isolation Forest και ο One-Class SVM δουλεύουν πάνω στον πολυδιάστατο πίνακα χαρακτηριστικών που προέκυψε με χρήση του ColumnTransformer. Τα αριθμητικά πεδία, όπως η συχνότητα εμφάνισης του προτύπου, η ώρα, η ημέρα της εβδομάδας, το μήκος του μηνύματος και τα πλήθη IP, HEX και NUM, περνούν από συμπλήρωση κενών τιμών και κανονικοποίηση. Τα κατηγορικά πεδία, όπως host, file, template_id και log_level, κωδικοποιούνται σε δυαδικές μεταβλητές μέσω One-Hot Encoding. Παράλληλα, το κείμενο του μηνύματος στο πεδίο clean_line μετατρέπεται σε διανύσματα TF-IDF. Όλες αυτές οι ομάδες χαρακτηριστικών συνενώνονται σε μία κοινή αναπαράσταση, πάνω στην οποία εφαρμόζονται στη συνέχεια ο Isolation Forest και ο One-Class SVM, ώστε να είναι άμεση η σύγκριση των αποτελεσμάτων τους.

Στην περίπτωση του One-Class SVM για να περιοριστεί η διάσταση, εφαρμόζεται Truncated SVD με 50 συνιστώσες. Κάθε log εγγραφή, που αρχικά περιγραφόταν από ένα πολύ μεγάλο διάνυσμα, συμπυκνώνεται σε ένα διάνυσμα πενήντα διαστάσεων. Η μέθοδος αυτή κρατά το πιο ισχυρό σήμα του dataset και ταυτόχρονα μειώνει τον θόρυβο και το κόστος εκπαίδευσης. Ο Isolation Forest, αντίθετα, εκπαιδεύεται απευθείας στην αρχική αναπαράσταση, καθώς τα δέντρα χειρίζονται πιο φυσικά την παρουσία πολλών δυαδικών και συνεχών χαρακτηριστικών.

Το νευρωνικό μοντέλο με GRU δεν αξιοποιούνται τα χαρακτηριστικά του πίνακα, αλλά η ίδια η ακολουθία των προτύπων. Τα logs ταξινομούνται χρονικά (ανά host και timestamp) και κάθε template μετατρέπεται σε event_id με χρήση LabelEncoder. Με βάση αυτή τη στήλη δημιουργούνται ακολουθίες σταθερού μήκους: κάθε δείγμα περιέχει πέντε συνεχόμενα γεγονότα και στόχος είναι η πρόβλεψη του έκτου. Ο διαχωρισμός σε εκπαίδευση και έλεγχο γίνεται χρονικά, με περίπου ογδόντα τοις εκατό των ακολουθιών να

χρησιμοποιούνται για training και το υπόλοιπο για test. Το GRU εκπαιδεύεται να προβλέπει το επόμενο template και στη συνέχεια χρησιμοποιείται για ανίχνευση ανωμαλιών, όταν η πραγματική συνέχεια δεν ανήκει στις top-k προβλέψεις του μοντέλου.

Κοινό στοιχείο για όλα τα μοντέλα είναι ότι δεν υπάρχουν labels που να δηλώνουν ποια γραμμή είναι κανονική και ποια είναι ανωμαλία. Δεν υπολογίζονται κλασικοί δείκτες ταξινόμησης, όπως accuracy ή F1 score, γιατί δεν υπάρχει έδαφος για άμεση σύγκριση με ground truth. Η αξιολόγηση βασίζεται στην υπόθεση ότι η μεγάλη πλειονότητα των logs αντιστοιχεί σε φυσιολογική λειτουργία και ότι οι πραγματικές ανωμαλίες είναι λίγες και σπάνιες. Με βάση αυτή την παραδοχή, κάθε μοντέλο παράγει ένα συνεχές score ανωμαλίας για όλες τις εγγραφές και στη συνέχεια εξάγεται ένα μικρό σύνολο εγγραφών με τις πιο ακραίες τιμές.

Τα αποτελέσματα κάθε μοντέλου αποθηκεύονται σε πίνακες μορφής CSV, ώστε για κάθε γραμμή καταγραφής να υπάρχει το αντίστοιχο score ανωμαλίας και το δυαδικό flag. Πάνω σε αυτούς τους πίνακες εφαρμόζονται τα φίλτρα συχνότητας των templates και η ομαδοποίηση παρόμοιων εγγραφών, με στόχο να προκύψει στο τέλος μια μικρή λίστα με τις πιο ακραίες ανωμαλίες για κάθε αλγόριθμο.

Για το νευρωνικό με GRU αποθηκεύονται σε ανάλογο πίνακα τόσο η πληροφορία για το αν η πραγματική συνέχεια της ακολουθίας βρέθηκε μέσα στις top-k προβλέψεις, όσο και η πιθανότητα που έδωσε το μοντέλο στο πραγματικό επόμενο γεγονός. Οι χρονικές στιγμές όπου η συνέχεια δεν προβλέπεται σωστά και συνοδεύεται από πολύ χαμηλή πιθανότητα χαρακτηρίζονται ως ανωμαλίες, και από αυτές επιλέγεται ένα υποσύνολο με τις πιο ακραίες τιμές.

Σε όλα τα μοντέλα εφαρμόζεται η ίδια λογική μετα-επεξεργασίας όπως είναι η απομάκρυνση πολύ συχνών templates, ώστε να φιλτράρονται καταγραφές ρουτίνας καθώς και ομαδοποίηση παρόμοιων γραμμών, ώστε να μην εμφανίζονται πολλές μικρές παραλλαγές του ίδιου γεγονότος. Στόχος είναι να μείνει, για κάθε μοντέλο, ένα σύνολο από λίγες αλλά αντιπροσωπευτικές ανωμαλίες που μπορούν να εξεταστούν χειροκίνητα και να αποτελέσουν τη βάση για την ποιοτική ανάλυση των επόμενων υποενοτήτων.

4.1 Αποτελέσματα Isolation Forest

Στο στάδιο αυτό εξετάζονται οι 100 εγγραφές που ο Isolation Forest ξεχώρισε ως πιο ύποπτες, μετά από όλα τα φίλτρα συχνότητας και την ομαδοποίηση παρόμοιων γραμμών.

Με τον όρο ανωμαλία δεν εννοούμε απαραίτητα κάποιο σφάλμα του συστήματος, αλλά γραμμές καταγραφής που, με βάση τα χαρακτηριστικά τους, βρίσκονται μακριά από τη γενική συμπεριφορά των υπόλοιπων logs. Είναι σπάνιες ως προς το template, έχουν ασυνήθιστο συνδυασμό πεδίων και απομονώθηκαν γρήγορα στα δέντρα του μοντέλου.

Κατανομή ανωμαλιών στα αρχεία καταγραφής

Οι 100 ανωμαλίες δεν μοιράζονται ομοιόμορφα σε όλα τα αρχεία. Υπάρχει συγκέντρωση σε λίγες πηγές.

- Οι 63 από τις 100 γραμμές προέρχονται από το vobd.log.
- Ακολουθεί το vmkernel.log με 14 γραμμές.
- Το esxupdate.log συνεισφέρει 12 ανωμαλίες.
- Το vmkwarning.log εμφανίζεται με 7 γραμμές.
- Τέλος, υπάρχουν μεμονωμένες εγγραφές από dhclient.log, syslog.log και hostd.log.

Αυτό σημαίνει ότι ο Isolation Forest βλέπει πιο ύποπτη συμπεριφορά κυρίως σε γεγονότα επιπέδου vobd και vmkernel, που σχετίζονται με υποσύστημα δικτύου, αποθήκευση και λειτουργία του host.

Δεν εμφανίζονται καταγραφές που να έχουν πιαστεί με κλασική ετικέτα ERROR ή WARN από την απλή κανονική έκφραση που χρησιμοποιήθηκε. Αυτό δείχνει ότι το μοντέλο εντοπίζει ύποπτα μοτίβα ακόμη και σε μηνύματα που, με βάση το level, δεν φαίνονται σοβαρά με την πρώτη ματιά.

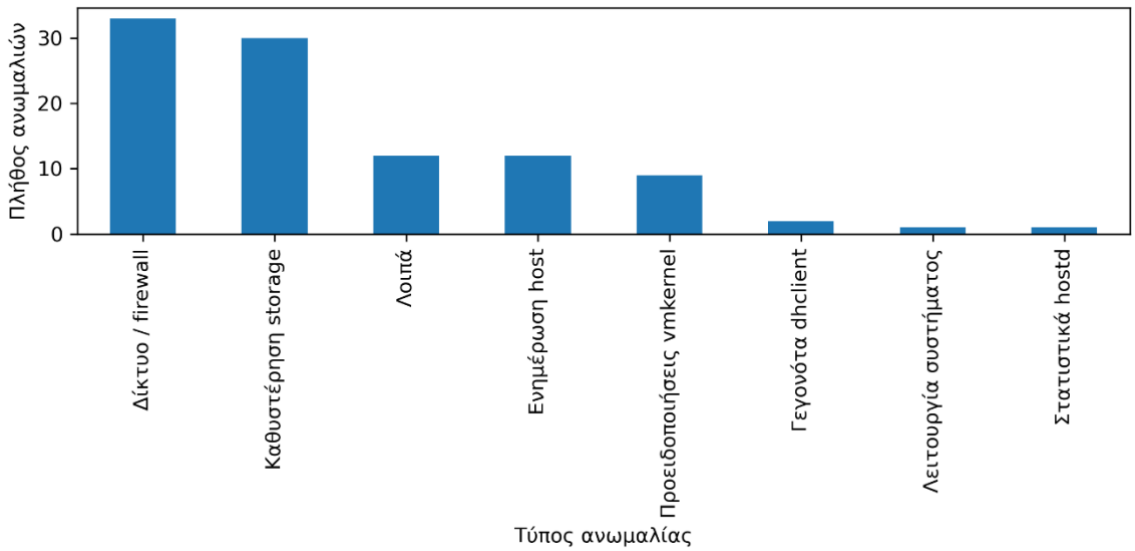
Η μεταβλητή template_freq στο top-100 κυμαίνεται από 2 έως 20 εμφανίσεις, με τυπική τιμή γύρω στις 12 επαναλήψεις ανά template. Οι πολύ σπάνιες γραμμές με μία μόνο εμφάνιση έχουν ήδη αποκλειστεί στα προηγούμενα στάδια, όπως και οι πολύ συχνές. Το αποτέλεσμα είναι ότι οι ανωμαλίες που εξετάζονται εδώ ανήκουν σε templates που εμφανίζονται αρκετές φορές ώστε να έχουν νόημα, αλλά όχι τόσο συχνά ώστε να θεωρηθούν καθαρή ρουτίνα.

Ο μέσος αριθμός IP διευθύνσεων ανά μήνυμα είναι περίπου 1,7, με τιμές από 0 μέχρι 8. Το μήκος του μηνύματος msg_len είναι σχετικά μεγάλο, με μέση τιμή περίπου 186 χαρακτήρες και μέγιστο 350. Πολλά από τα logs που επιλέγονται από τον Isolation Forest είναι εκτενή, με πλούσιο κείμενο και πολλές παραμέτρους, κάτι που τα κάνει να ξεχωρίζουν σε σχέση με πιο σύντομες καταγραφές.

Τύποι ανωμαλιών που εντοπίστηκαν

Από την ποιοτική εξέταση των 100 γραμμών προκύπτουν ορισμένες σαφείς κατηγορίες ανωμαλιών.

- **Αλλαγές συνδεσιμότητας δικτύου και firewall**: Τέτοιες εγγραφές δεν είναι απαραίτητα σφάλματα. Δείχνουν όμως μεταβάσεις σε κρίσιμα στοιχεία, όπως uplinks, δίκτυα διαχείρισης και firewall. Σε επίπεδο ανωμαλίας, αυτά είναι σημεία όπου το δίκτυο αλλάζει κατάσταση ή οι κανόνες προστασίας τροποποιούνται.
- **Προειδοποιήσεις καθυστέρησης σε συσκευές αποθήκευσης**: Άλλη μεγάλη ομάδα ανωμαλιών στο vobd.log έχει ετικέτες scsiCorrelator με περιγραφές τύπου vob.scsi.device.io.latency.high και esx.clear.scsi.device.io.latency.increased. Τα μηνύματα αναφέρονται σε συσκευές με αναγνωριστικά τύπου naa.6d0... και περιγράφουν αυξημένη καθυστέρηση I/O, σύγκριση με προηγούμενη μέση τιμή και τρέχουσες μετρήσεις.
- **Προειδοποιήσεις από vmkernel και vmkwarning**: Πρόκειται για χαμηλού επιπέδου προειδοποιήσεις που αφορούν storage pathing, PCI τοπολογία, file system υπηρεσίες και διαχείριση πόρων.
- **Δραστηριότητα ενημέρωσης host και ρυθμίσεων**: Στο esxupdate.log οι ανωμαλίες αφορούν εγγραφές με λεπτομέρειες για το εργαλείο ενημέρωσης. Εμφανίζονται μηνύματα με πολύ μεγάλες λίστες επιλογών, λεξικά Python με options όπως updateonly, dryrun, depot, profile, force, rebooting, και καταγραφές του HostImage για installers που ξεκινούν ή ολοκληρώνονται.
- **Μεμονωμένα γεγονότα δικτύου και συστήματος**: Γεγονότα που εμφανίζονται σπάνια στο συνολικό dataset και έχουν σαφές τεχνικό νόημα. Το μοντέλο τα θεωρεί ανωμαλίες επειδή διαφέρουν έντονα από την τυπική ροή των logs



Σχήμα 4.1: Κατηγοριοποίηση των top-100 ανωμαλιών του Isolation Forest ανά τύπο γεγονότος

Συνοψίζοντας τα παραπάνω, ο Isolation Forest δεν εντοπίζει ένα μόνο είδος ανωμαλίας. Αναδεικνύει κυρίως τρεις οικογένειες γεγονότων: μεταβολές δικτύου και firewall, αυξημένη καθυστέρηση σε συσκευές αποθήκευσης και δραστηριότητα ενημέρωσης του host, πλαισιωμένες από χαμηλού επιπέδου προειδοποιήσεις του vmkernel. Η έννοια της ανωμαλίας εδώ δεν περιορίζεται σε σφάλματα με τη στενή έννοια, αλλά επεκτείνεται σε ασυνήθιστες φάσεις συμπεριφοράς του συστήματος, που ξεχωρίζουν στατιστικά από τη μάζα των υπόλοιπων καταγραφών.

4.2 Αποτελέσματα One-Class SVM

Ο One-Class SVM εκπαιδεύτηκε πάνω στο ίδιο σύνολο χαρακτηριστικών με τον Isolation Forest, μετά τη μείωση διαστάσεων με Truncated SVD. Το μοντέλο παρήγαγε score για κάθε γραμμή log και χαρακτήρισε ως ανωμαλίες τις εγγραφές με ocsvm_flag = −1 και αρνητικό ocsvm_score. Από αυτές επιλέχθηκαν οι εκατό πιο ακραίες περιπτώσεις, πάνω στις οποίες έγινε η ποιοτική ανάλυση.

Οι ανωμαλίες που εντόπισε ο One-Class SVM κατανέμονται σε επτά αρχεία καταγραφής. Η πλειονότητα εμφανίζεται στα esxupdate.log, vobd.log και vmkernel.log, τα οποία μαζί συγκεντρώνουν σχεδόν οκτώ στις δέκα εγγραφές της λίστας. Τα esxupdate.log αντιστοιχούν σε λειτουργίες ενημέρωσης του host, εγκατάσταση ή τροποποίηση image profiles και εκτέλεση εντολών μέσω του μηχανισμού esxupdate. Τα vobd.log σχετίζονται με το scsiCorrelator και την πρόσθεση ή συσχέτιση SCSI paths προς δίσκους, ενώ τα

vmkernel.log καταγράφουν χαμηλού επιπέδου γεγονότα του πυρήνα του hypervisor. Η εικόνα αυτή δείχνει ότι ο One-Class SVM τραβάει την προσοχή προς δύο βασικές οικογένειες γεγονότων:

- ενημερώσεις host
- συμπεριφορά storage.

Όσον αφορά το επίπεδο καταγραφής, τα logs που επιλέγονται ως ανωμαλίες δεν είναι κυρίως σφάλματα. Περίπου το 44% εμφανίζονται με επίπεδο OTHER, το 34% ως INFO, το 18% ως DEBUG και μόνο ένα μικρό ποσοστό, γύρω στο 4%, ως ERROR. Το μοντέλο δεν ακολουθεί απλά τη σήμανση severity που δίνει το σύστημα, αλλά εντοπίζει γραμμές που είναι γεωμετρικά ακραίες στον χώρο των χαρακτηριστικών, ακόμη και αν παρουσιάζονται ως πληροφοριακές. Αυτό φαίνεται και στην κατανομή του template_freq: στο μισό των ανωμαλιών το αντίστοιχο template εμφανίζεται τουλάχιστον 86 φορές στο σύνολο των logs, ενώ στο ανώτερο 10% των περιπτώσεων ξεπερνά τις 2.000 εμφανίσεις, με ακραίες τιμές που φθάνουν σε δεκάδες ή και εκατοντάδες χιλιάδες γραμμές. Ο One-Class SVM δεν περιορίζεται σε σπάνια templates, αλλά συχνά επιλέγει περίεργες εκδοχές πολύ συχνών μηνυμάτων.

Ένα χαρακτηριστικό παράδειγμα είναι οι γραμμές του esxupdate που περιγράφουν την εκτέλεση σύνθετων εντολών με πολλές παραμέτρους. Στο generated dataset αυτές οι καταγραφές ανήκουν σε templates με μεγάλη συχνότητα, όμως ο συνδυασμός host, ώρας, μήκους μηνύματος και πλήθους διευθύνσεων IP τις κάνει να ξεχωρίζουν. Στις top-100 ανωμαλίες, ο μέσος αριθμός διευθύνσεων IP ανά γραμμή είναι περίπου 2,8, με τιμές που φθάνουν μέχρι και τις 13 IP σε μία μόνο εγγραφή. Αυτό παραπέμπει σε συσχετίσεις πολλαπλών endpoints, όπως συμβαίνει σε λειτουργίες multipath storage ή σε διαδικασίες ενημέρωσης που επηρεάζουν πολλά δίκτυα και διαδρομές ταυτόχρονα.

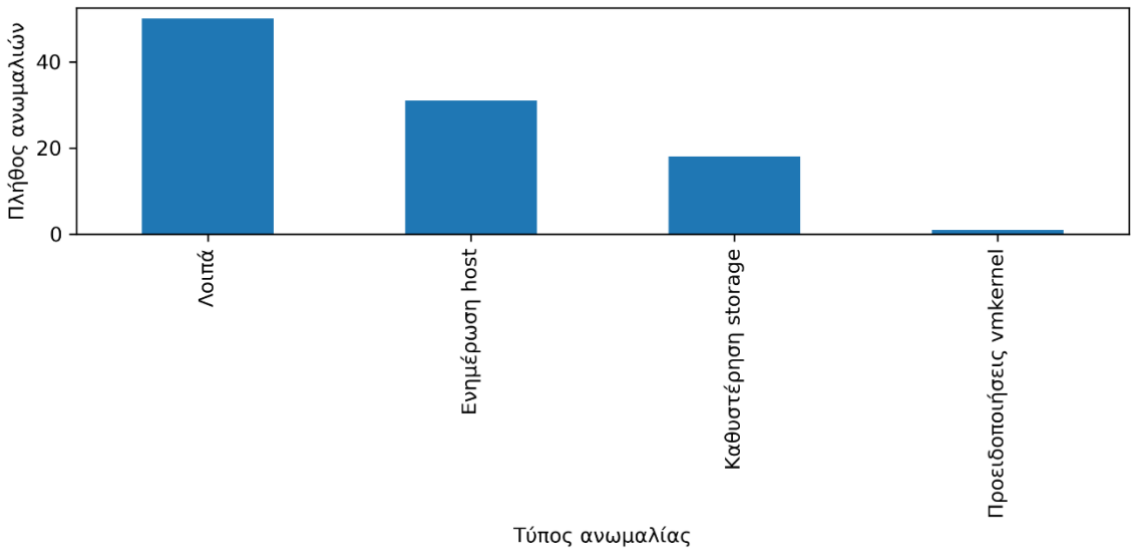
Για να υπάρξει καλύτερη εικόνα των μοτίβων που αναδεικνύει ο One-Class SVM, οι ανωμαλίες κατηγοριοποιήθηκαν χειροκίνητα, με βάση λέξεις-κλειδιά στο clean_line. Οι κύριες ομάδες που προκύπτουν είναι «Ενημέρωση host», «Καθυστέρηση / γεγονότα storage» και «Προειδοποιήσεις vmkernel», ενώ ένα μεγάλο μέρος συγκεντρώνεται σε μια γενικότερη κατηγορία «Λοιπά». Η ομάδα «Ενημέρωση host» περιλαμβάνει κυρίως εγγραφές από esxupdate.log, όπου καταγράφονται image profiles, κλήσεις του vmware.runcommand και διαδικασίες που αλλάζουν την εικόνα του host. Η ομάδα «Καθυστέρηση storage» προέρχεται κατά βάση από vobd.log και αφορά μηνύματα του

scsiCorrelator για προσθήκη paths (π.χ. vmhba0:C2:T0:L0), που συνδέονται με τη διαχείριση διαδρομών προς τους δίσκους. Στις «Προειδοποιήσεις vmkernel» εντάσσονται σπάνιες αλλά κρίσιμες γραμμές, όπως προειδοποιήσεις για PCI resources που «ήδη υπάρχουν» και δεν έγινε σωστά η προσθήκη κάποιου BAR.

Η κατηγορία «Λοιπά» περιλαμβάνει εγγραφές όπου ο vmkernel αναφέρει «unrecognised disk label» για συγκεκριμένους δίσκους, καθώς και μηνύματα από το syslog όπου ο μηχανισμός Host Profiles αδυνατεί να διαβάσει αρχεία ρυθμίσεων, όπως το /etc/security/login.map ή ορισμένα features εμφανίζονται ως μη ενεργά.

Ο One-Class SVM συμπληρώνει τον Isolation Forest. Ο πρώτος εντοπίζει γραμμές που είναι ακραίες μέσα σε πολύ πυκνές περιοχές του χώρου χαρακτηριστικών, συχνά σε επαναλαμβανόμενα templates που υπό άλλες συνθήκες θα θεωρούνταν «βαρετά». Ο δεύτερος εστιάζει περισσότερο σε σπάνιες μορφές μηνυμάτων. Η λίστα των top-100 ανωμαλιών που προκύπτει από τον One-Class SVM δείχνει ότι υπάρχουν γεγονότα ενημέρωσης, διαχείρισης storage και ελέγχου host profiles τα οποία δεν φωνάζουν με σαφές ERROR, αλλά ξεχωρίζουν στατιστικά μέσα στην κανονική συμπεριφορά των logs.

Ένα ενδεικτικό διάγραμμα της κατανομής των ανωμαλιών ανά κατηγορία παρουσιάζεται στο Σχήμα 4.2, όπου φαίνεται καθαρά η ενίσχυση των γεγονότων ενημέρωσης host και storage, καθώς και το ποσοστό των «Λοιπών» που συγκεντρώνουν κρίσιμα αλλά πιο διάσπαρτα σφάλματα.



Σχήμα 4.2: Κατηγοριοποίηση των top-100 ανωμαλιών του One-Class SVM ανά τύπο γεγονότος

4.3 Αποτελέσματα νευρωνικού μοντέλου ακολουθιών (GRU)

Το νευρωνικό μοντέλο ακολουθιών με GRU units δεν κοιτάζει κάθε γραμμή log μόνη της. Χρησιμοποιεί ακολουθίες πέντε προτύπων και προσπαθεί να προβλέψει ποιο template θα εμφανιστεί στη συνέχεια. Για κάθε θέση του test set υπολογίζει μια κατανομή πιθανοτήτων πάνω σε όλα τα templates. Αν το πραγματικό επόμενο template δεν βρίσκεται μέσα στις τρεις πιο πιθανές επιλογές, η χρονική στιγμή χαρακτηρίζεται ως ανωμαλία. Η πιθανότητα που δίνει το μοντέλο στο πραγματικό event αποθηκεύεται στο πεδίο nn_prob_true_event και παίζει ρόλο ως δείκτης «έκπληξης». Οι εκατό περιπτώσεις με τη χαμηλότερη πιθανότητα συγκροτούν τη λίστα των top-100 ανωμαλιών που εξετάζονται εδώ.

Οι τιμές του nn_prob_true_event στις εκατό αυτές εγγραφές είναι εξαιρετικά χαμηλές. Κυμαίνονται από περίπου $5 \cdot 10^{-14}$ μέχρι 10^{-11}, με διάμεσο κοντά στα $4 \cdot 10^{-12}$. Για το GRU αυτά τα γεγονότα είναι σχεδόν αδύνατα με βάση το μοτίβο που έχει μάθει. Δεν πρόκειται για μικρές αποκλίσεις, αλλά για σημεία όπου η πραγματική συνέχεια της ακολουθίας βρίσκεται πολύ μακριά από τις προβλέψεις του μοντέλου.

Η κατανομή των ανωμαλιών στα αρχεία καταγραφής είναι διαφορετική σε σχέση με τα άλλα δύο μοντέλα. Από τις 100 εγγραφές, οι 35 ανήκουν στο syslog.log, οι 25 στο vmkernel.log, οι 19 στο vmkwarning.log, οι 12 στο vobd.log, οι 5 στο esxupdate.log, ενώ υπάρχουν 3 γραμμές από το hostd.log και μία από το dhclient.log. Το νευρωνικό εστιάζει έντονα στο syslog.log, κάτι που δεν συνέβη στον Isolation Forest, και ταυτόχρονα κρατάει ισχυρή παρουσία από τα logs του πυρήνα και των προειδοποιήσεων (vmkernel και vmkwarning).

Ως προς το log_level, οι ανωμαλίες μοιράζονται σε 60 εγγραφές με level OTHER και 40 με level INFO. Δεν εμφανίζονται γραμμές με ERROR ή WARN. Το μοντέλο δεν κυνηγά τη δηλωμένη σοβαρότητα του μηνύματος, αλλά τη θέση του μέσα στο χρονικό μοτίβο. Ένα τυπικό ενημερωτικό log μπορεί να θεωρηθεί εξαιρετικά ύποπτο αν εμφανιστεί σε πλαίσιο που το GRU δεν αναγνωρίζει.

Η συχνότητα των templates ενισχύει αυτή την εικόνα. Η μεταβλητή template_freq για τα top-100 έχει ελάχιστη τιμή 4 εμφανίσεις, μέση τιμή περίπου 129 και μέγιστη κοντά στις 2.000. Το 50% των ανωμαλιών αντιστοιχεί σε templates με τουλάχιστον 58 εμφανίσεις, ενώ στο ανώτερο 10% η συχνότητα ξεπερνά τις 200. Συχνά σηματοδοτεί templates που εμφανίζονται πολλές φορές στο σύστημα, αλλά σε αυτή τη συγκεκριμένη θέση της ακολουθίας το θεωρεί πολύ απίθανο να εμφανιστούν. Σε μέσο όρο, κάθε γραμμή έχει

περίπου δύο IP διευθύνσεις (param_IP ≈ 2,09), με μέγιστο τις 8, ενώ τα πεδία param_HEX και param_NUM είναι μηδενικά. Οι ανωμαλίες του GRU βασίζονται κυρίως στη σειρά των γεγονότων και λιγότερο σε μεγάλους αριθμούς ή hex τιμές.

Από την ποιοτική εξέταση του περιεχομένου, προκύπτουν ορισμένες χαρακτηριστικές ομάδες ανωμαλιών.

Η πρώτη και πιο πολυπληθής ομάδα αφορά τα Host Profiles. Εδώ ανήκουν και οι 35 εγγραφές από το syslog.log, με μηνύματα τύπου «Host Profiles: Calling GatherData()» ή «Calling GenerateProfileFromConfig» για διάφορους τύπους profile, όπως PsaDeviceConfigurationProfile, SystemSwapConfigProfile και KernelModuleParamProfile. Τα templates αυτά εμφανίζονται αρκετές φορές στο σύνολο των logs, όμως το GRU τα έχει «συνηθίσει» σε συγκεκριμένες ακολουθίες κλήσεων. Οι γραμμές που βρέθηκαν στις top-100 αντιστοιχούν σε στιγμές όπου η εμφάνιση ενός τέτοιου μηνύματος δεν ταιριάζει με τη συνηθισμένη σειρά των Host Profiles. Για το μοντέλο, η διαδικασία συλλογής και δημιουργίας προφίλ εκτελείται σε κάπως διαφορετικό ρυθμό ή σειρά από αυτή που είδε στη φάση εκπαίδευσης.

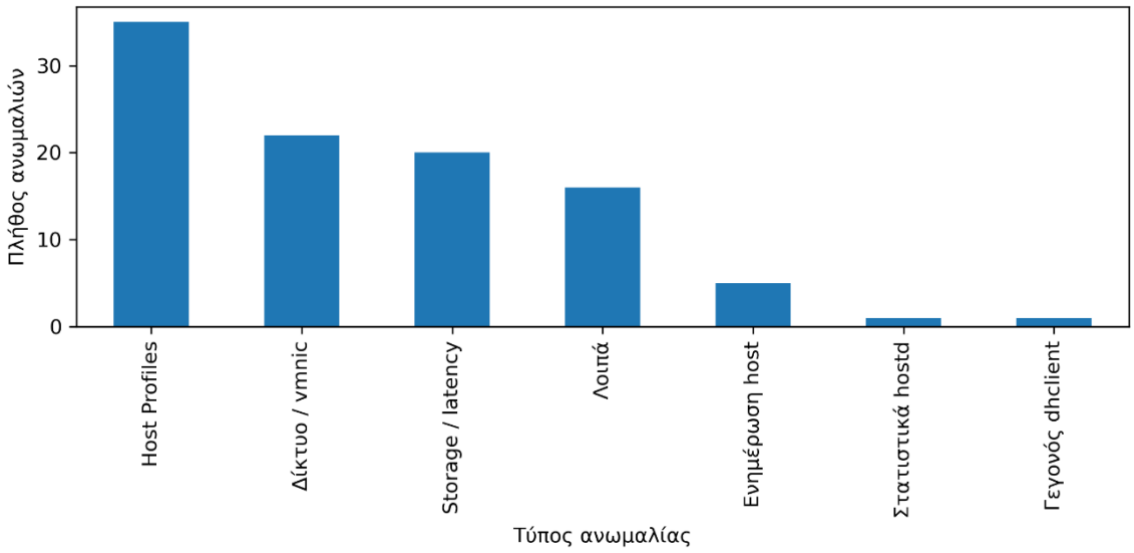
Η δεύτερη μεγάλη ομάδα αφορά storage και latency. Εδώ εντάσσονται 20 εγγραφές από τα αρχεία vmkernel.log, vmkwarning.log και vobd.log. Περιλαμβάνουν προειδοποιήσεις ScsiDeviceIO για συσκευές naa.* με αυξημένη καθυστέρηση I/O, μηνύματα vob.scsi.device.io.latency.high από το vobd.log, αλλά και καταγραφές του CBT στο vmkernel, όπου δημιουργούνται νέες συσκευές για τον changed block tracking driver. Τα μηνύματα αυτά δεν είναι πάντα μοναδικά. Το GRU τα θεωρεί ανωμαλίες όταν εμφανίζονται σε ακολουθίες όπου δεν περίμενε να δει απότομα αυξημένη latency ή δραστηριότητα CBT, σε σχέση με τις προηγούμενες πέντε γραμμές.

Ακολουθεί η οικογένεια των προειδοποιήσεων δικτύου και vmnic. Σε αυτήν μπαίνουν μηνύματα από το vmkwarning.log και το vobd.log που σχετίζονται με τις κάρτες δικτύου, την κατάσταση WOL και τις αλλαγές σε firewall και uplinks. Χαρακτηριστικά παραδείγματα είναι οι προειδοποιήσεις ntg3 για αλλαγές στο Wake-on-LAN, τα logs που δηλώνουν ότι ένας uplink vmnic0 ενεργοποιείται ή ότι ενεργοποιούνται συγκεκριμένα firewall rule sets. Ο One-Class SVM και ο Isolation Forest βλέπουν παρόμοιες γραμμές ως ανωμαλίες. Το νευρωνικό δίκτυο όμως τις εντοπίζει όταν σπάνε το κανονικό μοτίβο δικτυακής δραστηριότητας, για παράδειγμα όταν εισβάλλει ξαφνικά ένα firewall change ανάμεσα σε Host Profiles ή storage μηνύματα.

Μια ακόμη ενδιαφέρουσα ομάδα είναι η ενημέρωση του host και οι κλήσεις του esxupdate. Τα 5 αντίστοιχα logs στο esxupdate.log περιγράφουν εντολές με λεπτομερείς επιλογές, όπως dictionaries με flags dryrun, oktoremove, nomaintmode, depot, καθώς και κλήσεις vmware.runcommand με παραμέτρους timeout. Τα templates αυτά είναι αρκετά συχνά, όμως ο συνδυασμός τους με το άμεσο παρελθόν της ακολουθίας τα καθιστά ασυνήθιστα για το μοντέλο. Συνδέονται συνήθως με φάσεις ενημέρωσης ή αλλαγής image profile.

Τέλος, υπάρχουν μερικές πιο διάσπαρτες κατηγορίες, όπως τα στατιστικά του hostd και ένα μεμονωμένο γεγονός από το dhclient.log. Στο hostd.log εμφανίζονται γραμμές όπου ο μηχανισμός Statssvc αναφέρει ακραίες τιμές για read I/O ή μηνύματα του τύπου «message = <unset>» και «fetch_main_sel: got reservation». Στο dhclient.log το μοντέλο σημαδεύει μια γραμμή dhclient-uw που στέλνει κίνηση σε PF_INET6/vmk0. Αυτά τα μηνύματα δεν ξεχωρίζουν τόσο από το ίδιο τους το περιεχόμενο, όσο από το σημείο στο οποίο παρεμβαίνουν μέσα στην ροή των υπόλοιπων γεγονότων.

Η συνολική εικόνα δείχνει ότι το νευρωνικό μοντέλο με GRU συμπεριφέρεται συμπληρωματικά σε σχέση με τον Isolation Forest και τον One-Class SVM. Δεν κάνει μόνο έλεγχο σπανιότητας ή γεωμετρικής απόστασης στον χώρο των χαρακτηριστικών. Εντοπίζει κυρίως ακολουθίες όπου η σειρά των γεγονότων δεν ταιριάζει με τα μοτίβα που έμαθε από το ιστορικό των logs.



Σχήμα 4.3: Κατηγοριοποίηση των top-100 ανωμαλιών του GRU ανά τύπο γεγονότος

4.4 Συγκριτική αξιολόγηση των μοντέλων ανίχνευσης ανωμαλιών

Στην ενότητα αυτή συγκρίνονται τα τρία μοντέλα που εφαρμόστηκαν στα logs του vSphere. Και τα τρία δουλεύουν πάνω στο ίδιο σύνολο χαρακτηριστικών και παράγουν λίστες με τις εκατό πιο ακραίες ανωμαλίες, αλλά η ματιά τους στο ίδιο υλικό είναι διαφορετική.

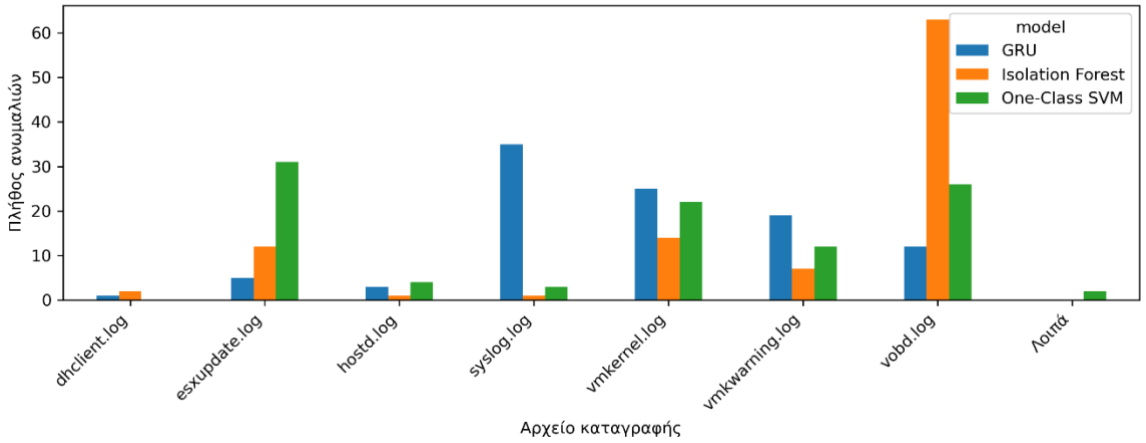
Ο Isolation Forest τείνει να αναδεικνύει γραμμές που είναι σπάνιες και απομονωμένες στον χώρο χαρακτηριστικών. Δίνει βάρος σε μηνύματα από vobd.log και vmkernel.log, με έντονη παρουσία προειδοποιήσεων για latency σε συσκευές storage, αλλαγές σε uplinks δικτύου και τροποποιήσεις firewall. Πολλά από αυτά τα templates εμφανίζονται λίγες φορές σε όλο το dataset, οπότε το μοντέλο τα «κόβει» γρήγορα ως outliers. Η λίστα του Isolation Forest είναι χρήσιμη όταν ο διαχειριστής θέλει να ξεκινήσει από γεγονότα που είναι και σπάνια και τεχνικά φορτωμένα.

Το One-Class SVM, αντίθετα, δουλεύει πάνω στη μειωμένη αναπαράσταση που προκύπτει από το Truncated SVD και ψάχνει λεπτά όρια ανάμεσα στο κανονικό και στο ύποπτο. Οι ανωμαλίες του συγκεντρώνονται κυρίως σε esxupdate.log, vobd.log και vmkernel.log. Πολλές γραμμές αφορούν ενημέρωση host, αλλαγές σε image profiles και εντολές esxupdate με πολλά ορίσματα. Άλλες σχετίζονται με διαδρομές SCSI και διαχείριση storage. Τα templates αυτά δεν είναι απαραίτητα σπάνια. Συχνά εμφανίζονται δεκάδες ή εκατοντάδες φορές, όμως σε συγκεκριμένους συνδυασμούς host, ώρας και περιεχομένου ξεφεύγουν από τα όρια του μοντέλου. Ο One-Class SVM είναι πιο χρήσιμος όταν ο στόχος είναι να βρεθούν «περίεργες» περιπτώσεις μέσα σε γεγονότα που, με το μάτι, μοιάζουν ρουτίνα.

Το νευρωνικό GRU φέρνει μια τρίτη οπτική. Δεν στέκεται μόνο στο τι γράφει κάθε γραμμή, αλλά στο πού εμφανίζεται μέσα στην ακολουθία των templates. Οι ανωμαλίες του δεν είναι πάντα σπάνια μηνύματα. Συχνά πρόκειται για πολύ συνηθισμένα templates που όμως εμφανίζονται σε εντελώς διαφορετικό πλαίσιο από αυτό που έχει μάθει το μοντέλο στην εκπαίδευση. Η λίστα του GRU γεμίζει με γεγονότα Host Profiles στο syslog.log, προειδοποιήσεις για latency σε storage και μηνύματα που αφορούν κάρτες δικτύου και uplinks. Κοινό στοιχείο είναι ότι το πραγματικό επόμενο template έχει εξαιρετικά χαμηλή πιθανότητα σύμφωνα με το GRU, άρα η σειρά των γεγονότων είναι ασυνήθιστη, ακόμη κι αν το ίδιο το μήνυμα εμφανίζεται αρκετές φορές σε άλλα σημεία των logs.

Η εικόνα αυτή αποτυπώνεται και στην κατανομή των ανωμαλιών ανά αρχείο καταγραφής, όπως φαίνεται στο Σχήμα 4.4. Για την ανάγκη του σχήματος, όλα τα αρχεία που δεν

ανήκουν στα dhclient.log, esxupdate.log, hostd.log, syslog.log, vmkernel.log, vmkwarning.log και vobd.log ομαδοποιούνται στην κατηγορία «Λοιπά». Στο δείγμα των top-100 ανωμαλιών μόνο ο One-Class SVM εντοπίζει δύο γραμμές σε αυτή την κατηγορία, ενώ ο Isolation Forest και το GRU συγκεντρώνουν όλες τις ανωμαλίες στα βασικά αρχεία log.



Σχήμα 4.4: Κατανομή των ανωμαλιών ανά αρχείο log και μοντέλο

Το GRU συγκεντρώνει 35 από τις 100 ανωμαλίες στο syslog.log και στη συνέχεια μοιράζει τις υπόλοιπες κυρίως στα vmkernel.log (25 εγγραφές), vmkwarning.log (19 εγγραφές) και vobd.log. Ο Isolation Forest, αντίθετα, φορτώνει έντονα το vobd.log, από όπου προέρχονται 63 ανωμαλίες, ενώ ακολουθούν τα vmkernel.log και vmkwarning.log με σαφώς χαμηλότερα πλήθη. Ο One-Class SVM κινείται ενδιάμεσα: εντοπίζει 31 ανωμαλίες στο esxupdate.log, 26 στο vobd.log, 22 στο vmkernel.log και λιγότερες στο syslog.log και στο vmkwarning.log. Η κατανομή αυτή δείχνει ότι κάθε μοντέλο τραβάει την προσοχή σε διαφορετικό σύνολο αρχείων, παρότι δουλεύουν πάνω στο ίδιο dataset.

Πέρα από τα αρχεία, τα μοντέλα μοιράζονται παρόμοιες θεματικές περιοχές. Και τα τρία αναδεικνύουν γεγονότα που σχετίζονται με storage και latency, δικτυακές διεπαφές και uplinks, αλλαγές firewall, ενημερώσεις host και λειτουργία Host Profiles. Η διαφορά βρίσκεται στον τρόπο με τον οποίο φτάνουν σε αυτά τα γεγονότα. Ο Isolation Forest ξεκινά από τη σπανιότητα και την ευκολία απομόνωσης. Ο One-Class SVM βασίζεται στη γεωμετρική απόσταση σε έναν χώρο χαμηλότερης διάστασης. Το GRU κοιτάζει την ιστορία, δηλαδή ποια γεγονότα συνήθως προηγούνται και ποια ακολουθούν. Έτσι, ένα συμβάν μπορεί να εμφανιστεί μόνο στην μία λίστα, σε δύο ή και στις τρεις, ανάλογα με το

αν είναι σπάνιο, αν βγαίνει έξω από το όριο του SVM ή αν «σπάει» την αναμενόμενη χρονική ακολουθία.

Ο Isolation Forest είναι πιο ελαφρύς υπολογιστικά και προσφέρει μια γρήγορη πρώτη ματιά στα πιο απομονωμένα logs. Ο One-Class SVM απαιτεί προεπεξεργασία με SVD, αλλά εμφανίζει τις ακραίες περιπτώσεις μέσα σε πυκνές περιοχές δεδομένων. Το GRU είναι πιο απαιτητικό σε εκπαίδευση, αλλά εισάγει πληροφορία για τη ροή των γεγονότων. Ο συνδυασμός των τριών δίνει μια πιο πλήρη εικόνα της συμπεριφοράς του συστήματος: σπανιότητα, γεωμετρία και χρονική σειρά μπαίνουν στο ίδιο τραπέζι και βοηθούν τον αναλυτή να ξεχωρίσει τα πραγματικά ενδιαφέροντα συμβάντα από τον θόρυβο της καθημερινής λειτουργίας.

Με βάση τα παραπάνω ευρήματα, η υπεροχή κάθε μοντέλου εξαρτάται από το ζητούμενο της ανάλυσης. Αν ο στόχος είναι μια γρήγορη και πρακτική πρώτη διαλογή των πιο ασυνήθιστων εγγραφών με χαμηλό υπολογιστικό κόστος, τότε ο Isolation Forest υπερέχει, επειδή εντοπίζει αποτελεσματικά σπάνια και απομονωμένα συμβάντα και δίνει μια άμεση λίστα για διερεύνηση. Αν, αντίθετα, ο στόχος είναι να εντοπιστούν περίεργες περιπτώσεις μέσα σε γεγονότα που εμφανίζονται συχνά (δηλαδή ανωμαλίες που κρύβονται σε πυκνές περιοχές), τότε ο One-Class SVM υπερέχει ως προς αυτή τη λεπτή διάκριση, αν και παρουσιάζει μεγαλύτερη ευαισθησία σε παραμετροποίηση των δεδομένων.

Για σενάρια όπου το ζητούμενο είναι η προγνωστική ικανότητα (δηλαδή αν οι ανωμαλίες είναι αποτέλεσμα ασυνήθιστης ροής γεγονότων και όχι απλώς σπάνιων μηνυμάτων), το GRU υπερέχει, επειδή αξιοποιεί τη χρονική πληροφορία και εντοπίζει αποκλίσεις στη σειρά των templates ακόμη κι όταν τα ίδια μηνύματα είναι συνηθισμένα. Αυτό το χαρακτηριστικό το καθιστά πιο κατάλληλο όταν ο αναλυτής θέλει να αναγνωρίσει προδρομικά μοτίβα (precursor patterns) που προηγούνται προβλημάτων.

Κλείνωντας, ως καλύτερη γενική λύση για πρώτη εφαρμογή σε logs προτείνεται ο Isolation Forest λόγω σταθερότητας και χαμηλού κόστους, ενώ ως καλύτερη λύση για πρόβλεψη/πρόδρομα μοτίβα προτείνεται το GRU, επειδή ενσωματώνει τη χρονική δομή των γεγονότων. Η τελική επιλογή εξαρτάται από το αν η ανάλυση στοχεύει περισσότερο σε άμεσο triage ή σε πρόγνωση μέσω ακολουθιακών μοτίβων.

Μοντέλο	**Καλύτερο σε**	**Πλεονεκτήματα**	**Μειονεκτήματα**
Isolation Forest	Σπάνια συμβάντα	Γρήγορο, σταθερό	Χάνει ακολουθιακές ανωμαλίες
One-Class SVM	Λεπτές αποκλίσεις σε πυκνά δεδομένα	Καλή οριοθέτηση	Ευαίσθητο σε ρύθμιση/διάσταση
GRU	Ασυνήθιστη σειρά γεγονότων	Πρόδρομα μοτίβα/ πρόβλεψη	Βαρύ υπολογιστικά

Πίνακας 4.1: Σύγκριση των μοντέλων ανίχνευσης ανωμαλιών

5 Συζήτηση

Η ανάλυση των αποτελεσμάτων δείχνει ότι τα τρία μοντέλα δεν βλέπουν τα logs με τον ίδιο τρόπο. Το vSphere host esx-vshere3.kos.gr παράγει χιλιάδες επαναλαμβανόμενα γεγονότα ρουτίνας και, μέσα σε αυτά, λίγες μόνο γραμμές που μοιάζουν να ξεφεύγουν από τον κανόνα. Το πώς ορίζεται αυτή η απόκλιση είναι ακριβώς το σημείο όπου διαφοροποιούνται ο Isolation Forest, ο One-Class SVM και το νευρωνικό GRU.

Ο Isolation Forest, που στηρίζεται στην ιδέα της στατιστικής απομόνωσης, τείνει να εντοπίζει κυρίως σπάνια και βαριά τεχνικά μηνύματα. Στη λίστα των 100 πιο έντονων ανωμαλιών κυριαρχούν καταγραφές από το vobd.log και το vmkernel.log, με θέματα όπως αυξημένη καθυστέρηση σε συσκευές αποθήκευσης, αλλαγές σε paths SCSI, μεταβολές δικτυακών διεπαφών και κανόνων firewall. Πρόκειται για γεγονότα που δεν εμφανίζονται διαρκώς στα logs, αλλά όταν εμφανιστούν συνοδεύονται από πλούσιο τεχνικό περιεχόμενο (πολλές IP, μεγάλα identifiers, σύνθετα μονοπάτια). Ο αλγόριθμος αυτά τα ακραία σημεία τα απομονώνει εύκολα στα τυχαία δέντρα του, ακόμη κι όταν το επίπεδο log δεν είναι τυπικό ERROR ή WARN.

Ο One-Class SVM, αντίθετα, δρα περισσότερο σαν μαθηματικό κέλυφος γύρω από την κανονικότητα. Μετά τη μείωση διαστάσεων, επιχειρεί να περικλείσει τη μεγάλη μάζα των εγγραφών σε έναν γεωμετρικό όγκο και να απορρίψει ό,τι πέφτει εκτός. Έτσι δεν περιορίζεται στα σπάνια templates αφού πολλές από τις ανωμαλίες του προέρχονται από ιδιαίτερα συχνά μηνύματα του esxupdate.log και του vobd.log, τα οποία όμως, σε συγκεκριμένους συνδυασμούς host, ώρας, πλήθους IP και μήκους, βρίσκονται ακριβώς στο χείλος της κανονικής περιοχής.

Το GRU δεν ενδιαφέρεται τόσο για το πόσο ασυνήθιστο είναι ένα μεμονωμένο μήνυμα, αλλά για το αν η συνέχεια μιας μικρής ιστορίας πέντε γεγονότων είναι αυτή που περιμένει. Τα top-100 συμβάντα που σηματοδοτεί έχουν εξαιρετικά χαμηλή πιθανότητα σύμφωνα με την κατανομή εξόδου του μοντέλου. Πολλά ανήκουν στο syslog.log και συνδέονται με τον μηχανισμό Host Profiles, με κλήσεις τύπου GatherData() και GenerateProfileFromConfig() που εμφανίζονται σε ακολουθίες διαφορετικές από αυτές που είχε μάθει το δίκτυο στην εκπαίδευση. Άλλα σχετίζονται με spikes latency σε storage και ξαφνικές αλλαγές σε uplinks ή firewall. Εδώ η ανωμαλία δεν είναι τόσο το περιεχόμενο του μηνύματος, όσο το ότι μπαίνει σε μια σειρά γεγονότων όπου δεν θα έπρεπε να βρίσκεται.

Ένα από τα βασικά συμπεράσματα της συζήτησης είναι ότι η έννοια της ανωμαλίας στα logs του vSphere δεν είναι μονοδιάστατη. Άλλες φορές σημαίνει σπάνιο τεχνικό γεγονός (όπως μια μεμονωμένη αύξηση καθυστέρησης σε SCSI συσκευή), άλλες φορές σημαίνει ασυνήθιστος συνδυασμός χαρακτηριστικών (π.χ. μια εντολή ενημέρωσης με ασυνήθιστο αριθμό παραμέτρων και IP), και άλλες φορές σημαίνει απλώς λάθος σειρά σε μια κατά τα άλλα συνηθισμένη ροή Host Profiles.

Εξίσου σημαντικό είναι ότι σε όλα τα μοντέλα αναδύονται οι ίδιες περίπου θεματικές περιοχές:

- υποσύστημα αποθήκευσης
- δικτυακές διεπαφές και uplinks
- firewall και κανόνες πρόσβασης
- κύκλοι ενημέρωσης host
- και λειτουργίες Host Profiles

Με άλλα λόγια, τα logs επανέρχονται σταθερά σε συγκεκριμένα σημεία ευαισθησίας του vSphere. Αν συνδεθεί αυτό με τη βιβλιογραφία της Log-based Anomaly Detection, όπου αντίστοιχες περιοχές (storage, network, configuration drift) εμφανίζονται ως βασικές πηγές σφαλμάτων σε μεγάλες υποδομές, τότε τα ευρήματα της εργασίας ευθυγραμμίζονται με όσα έχουν βρεθεί σε μεγαλύτερες μελέτες.

Ένα δεύτερο κεντρικό μήνυμα της συζήτησης είναι η βαρύτητα της προ-επεξεργασίας. Δίχως τον καθαρισμό των γραμμών, την εξαγωγή προτύπων, τη μετατροπή σε TF-IDF και τη δημιουργία δεικτών σπανιότητας ακολουθιών, τα μοντέλα θα έβλεπαν απλώς ακατέργαστο κείμενο με τεράστια ποικιλία και θα κατέληγαν σε άχρηστα, θορυβώδη

αποτελέσματα. Το γεγονός ότι οι τρεις πολύ διαφορετικές μέθοδοι καταλήγουν να εντοπίζουν παρεμφερείς ζώνες ανωμαλιών δείχνει ότι ο πίνακας χαρακτηριστικών αποτυπώνει, σε ικανοποιητικό βαθμό, τη λειτουργική δομή του συστήματος. Ουσιαστικά, η έξυπνη ανίχνευση ξεκινά από το πώς μεταφράζουμε τις γραμμές logs σε αριθμούς.

Παράλληλα, υπάρχουν και σαφείς περιορισμοί. Ο πιο προφανής είναι η απουσία των labels. Δεν γνωρίζουμε εκ των προτέρων ποιες καταγραφές αντιστοιχούν σε πραγματικά περιστατικά βλάβης ή ασφάλειας, άρα δεν μπορούμε να υπολογίσουμε κλασικούς δείκτες απόδοσης. Η αξιολόγηση στηρίζεται σε ποιοτική επιθεώρηση των top-100 ανωμαλιών ανά μοντέλο και σε συσχετίσεις με γνωστές ευαίσθητες περιοχές (storage, network κ.λπ.). Αυτό σημαίνει ότι το σύστημα, όπως υλοποιήθηκε, είναι καλύτερα προσανατολισμένο στη διερευνητική ανάλυση και στην υποβοήθηση του διαχειριστή, παρά στην πλήρως αυτόματη λήψη αποφάσεων (π.χ. αυτόματη διακοπή VM).

Δεύτερος περιορισμός είναι η εξάρτηση από τις επιλεγμένες υπερπαραμέτρους. Η υπόθεση ότι περίπου 1% των εγγραφών είναι ανώμαλες (contamination), το μήκος της ακολουθίας (πέντε γεγονότα), ο αριθμός των συνιστωσών SVD, το μέγεθος του embedding, όλα αυτά επηρεάζουν το ποια logs θα αναδειχθούν τελικά. Με διαφορετικές επιλογές, ο κατάλογος των top-100 θα άλλαζε. Η παρούσα εργασία δεν εξερευνά συστηματικά όλο τον χώρο των παραμέτρων, αλλά επιλέγει ρεαλιστικές τιμές που ισορροπούν μεταξύ ακρίβειας και υπολογιστικού κόστους.

Τρίτος περιορισμός αφορά την αντιπροσωπευτικότητα των δεδομένων. Η ανάλυση βασίστηκε σε έναν συγκεκριμένο host και σε ένα συγκεκριμένο χρονικό παράθυρο. Αν αλλάξει ριζικά ο τρόπος λειτουργίας της υποδομής (νέες υπηρεσίες, αναβαθμίσεις, διαφορετικό φορτίο), τότε τόσο τα templates όσο και οι συχνότητές τους θα μεταβληθούν. Τα μοντέλα, ειδικά το GRU, θα χρειαστούν επανεκπαίδευση, αλλιώς θα αρχίσουν να βλέπουν ως ανωμαλία συμπεριφορές που απλώς είναι καινούργια κανονικότητα.

Παρόλα αυτά, η συνολική εικόνα είναι ενθαρρυντική. Με σχετικά τυπικά εργαλεία της Python και της Μηχανικής Μάθησης, και χωρίς την πολυτέλεια πλήρως επισημασμένων δεδομένων, κατέστη δυνατό να αναδειχθούν συγκεκριμένες ζώνες κινδύνου στο vSphere: αποθήκευση, δίκτυο, κύκλοι ενημέρωσης και Host Profiles. Η συζήτηση αυτή δεν κλείνει το θέμα της ανίχνευσης ανωμαλιών στα logs αλλά δείχνει ότι η προσέγγιση που ακολουθήθηκε μπορεί να λειτουργήσει ως σκελετός πάνω στον οποίο μπορούν να χτιστούν πιο πλούσιες λύσεις, με ενσωμάτωση feedback από διαχειριστές, προσθήκη επιβλεπόμενων στοιχείων εκεί όπου υπάρχουν labels και προσαρμογή σε μεγαλύτερες, multi-host

εγκαταστάσεις. Ως εκ τούτου, μια ιδιαίτερα αποτελεσματική προσέγγιση πρακτικής επαλήθευσης συνίσταται στη χρήση των «Top-100» ανωμαλιών υψηλότερης βαθμολογίας ως αφετηρία ποιοτικής ανάλυσης. Συγκεκριμένα, ένας διαχειριστής συστήματος δύναται να διασταυρώσει χρονικά τα παραγόμενα alerts με τα καταγεγραμμένα IT Incident Tickets της αντίστοιχης ημέρας ή χρονικής περιόδου, μέσω συστημάτων διαχείρισης συμβάντων όπως το ServiceNow ή το GLPI, προκειμένου να εξετάσει εάν κάθε ανωμαλία αντιστοιχεί σε μια τεκμηριωμένη αναφορά βλάβης ή διακοπής υπηρεσίας. Αυτές οι προεκτάσεις αποτελούν το φυσικό επόμενο βήμα και συνδέονται άμεσα με τα συμπεράσματα και τις προτάσεις μελλοντικής εργασίας που παρουσιάζονται στο επόμενο κεφάλαιο.

6 Συμπέρασμα και Μελλοντική Εργασία

Τα logs του vSphere γεμίζουν μέρα με τη μέρα με γραμμές που μοιάζουν όλες ίδιες. Ένας διαχειριστής καλείται να ξεχωρίσει μέσα σε αυτό τον όγκο τις λίγες στιγμές που αξίζει να ανησυχήσει. Το ερώτημα ήταν απλό στη διατύπωση, μπορεί μια σειρά από αλγορίθμους ανίχνευσης ανωμαλιών να φιλτράρει αυτό το χάος και να προσφέρει μια συμπυκνωμένη, χρήσιμη εικόνα για τη συμπεριφορά ενός ESXi host;

Το πρώτο βήμα ήταν να μετατραπεί το αρχικό κείμενο σε μια ολοκληρωμένη δομή. Τα logs δεν είναι φυσικά πίνακας αλλά είναι προτάσεις, timestamps, τεχνικοί όροι, identifiers, μονοπάτια. Με την εξαγωγή templates, τη διάσπαση του timestamp, τη δημιουργία χαρακτηριστικών όπως template_freq, hour, day_of_week, msg_len και την καταμέτρηση IP, HEX και μεγάλων αριθμών, το υλικό πήρε αριθμητική μορφή. Η προσθήκη TF-IDF πάνω στο καθαρισμένο μήνυμα εμπλούτισε ακόμη περισσότερο την αναπαράσταση. Από ένα σύνολο απλών γραμμών κειμένου προέκυψε ένας πολυδιάστατος πίνακας, όπου κάθε καταγραφή είχε θέση μέσα σε έναν χώρο χαρακτηριστικών που συνδέεται με τη λειτουργία του συστήματος.

Η κεντρική διαπίστωση είναι ότι η ποιότητα αυτού του πίνακα καθόρισε σε μεγάλο βαθμό την ποιότητα των αποτελεσμάτων. Ο Isolation Forest, ο One-Class SVM και το GRU πάτησαν πάνω σε μια αναπαράσταση που αναδεικνύει τις περιοχές της υποδομής όπου συσσωρεύεται η ουσία. Storage, δίκτυο, firewall, ενημερώσεις host, Host Profiles. Αυτά τα θέματα επανέρχονται σταθερά στις λίστες ανωμαλιών και των τριών μοντέλων. Αυτό

σημαίνει ότι, παρά τους περιορισμούς, ο τρόπος που χτίστηκε ο πίνακας χαρακτηριστικών περιγράφει αρκετά πιστά την κανονική και την περίεργη συμπεριφορά του vSphere host.

Στην πορεία φάνηκε καθαρά και ο ρόλος του κάθε μοντέλου. Ο Isolation Forest σημάδεψε σπάνιες και τεχνικά φορτωμένες γραμμές, όπου η καθυστέρηση I/O, οι ασυνήθιστες SCSI διαδρομές και οι αλλαγές σε firewall και uplinks ξεχώριζαν. Ο One-Class SVM πήγε πιο βαθιά στις πυκνές περιοχές και έφερε στην επιφάνεια περιπτώσεις όπου εντολές esxupdate και γεγονότα ενημέρωσης host δεν ήταν σπάνια, αλλά η συγκεκριμένη εκδοχή τους έμοιαζε οριακή ως προς το πώς τα έβλεπε το μοντέλου. Το GRU λειτούργησε σαν παρατηρητής της ροής αφού δεν ανησυχούσε τόσο για το αν ένα μήνυμα είναι σπάνιο, αλλά για το αν έρχεται στη σωστή θέση μέσα στην ακολουθία. Έτσι αναδείχθηκαν Host Profiles που εκτελούνται σε διαφορετική σειρά, latency peaks που πετάγονται σε αταίριαστο σημείο και δικτυακές προειδοποιήσεις που κόβουν τον ρυθμό των συνηθισμένων γεγονότων.

Από αυτή τη σύγκριση προκύπτει το συμπέρασμα ότι δεν υπάρχει ένα μοντέλο που τα κάνει όλα. Η επικαλυπτόμενη περιοχή ανάμεσα στις τρεις λίστες είναι σχετικά μικρή. Αυτό δεν δηλώνει αποτυχία, αλλά διαφορετικό ορισμό της ανωμαλίας αφού άλλο πράγμα είναι η σπανιότητα, άλλο η γεωμετρική απόσταση σε χώρο χαρακτηριστικών, άλλο η απρόσμενη χρονική ακολουθία. Ο συνδυασμός τους, μαζί με τα γραφήματα κατανομής ανά αρχείο log και ανά τύπο ανωμαλίας, δίνει μια πιο στέρεη βάση για διερεύνηση.

Η εργασία, βέβαια, δεν στέκεται έξω από τους περιορισμούς της αφού δεν υπάρχουν labels. Η αξιολόγηση γίνεται έμμεσα, μέσα από την τεχνική κατανόηση των μηνυμάτων, τη σύγκριση των λιστών μεταξύ τους και την ευθυγράμμιση με όσα αναφέρει η βιβλιογραφία για αντίστοιχα περιβάλλοντα. Αυτό σημαίνει ότι δεν προκύπτουν εύκολοι δείκτες, όπως ποσοστό ανίχνευσης ή false positives.

Ένας ακόμη περιορισμός αφορά την κλίμακα. Όλα έγιναν πάνω σε έναν host και σε ένα συγκεκριμένο χρονικό παράθυρο. Η συμπεριφορά ενός συστήματος όμως δεν είναι στατική. Νέες υπηρεσίες, αλλαγές σε workload, αναβαθμίσεις λογισμικού μπορούν να μεταφέρουν την κανονικότητα σε άλλη θέση. Τα templates θα αλλάξουν, οι συχνότητες θα μετακινηθούν, οι ακολουθίες θα πάρουν νέα μορφή. Τα μοντέλα, και ειδικά το GRU, θα χρειαστούν εκ νέου εκπαίδευση, αλλιώς θα αρχίσουν να βλέπουν τη νέα κανονικότητα σαν ανωμαλία. Αυτή η ανάγκη προσαρμογής είναι κομμάτι της πραγματικότητας σε κάθε ζωντανό σύστημα και δεν λύνεται με μία και μόνο πειραματική μελέτη.

Παρά τα όρια αυτά, η συνολική εικόνα είναι ενθαρρυντική. Με απλά εργαλεία Python, χωρίς labeled δεδομένα, κατασκευάστηκε ένα pipeline που καταφέρνει να απομονώσει

συγκεκριμένες ζώνες κινδύνου σε ένα vSphere host. Ο διαχειριστής δεν μένει πλέον μόνο με τη δυνατότητα του grep και της χειροκίνητης αναζήτησης. Μπορεί να έχει μπροστά του μια λίστα με γραμμές, ταξινομημένες κατά ύποπτη συμπεριφορά, και να βλέπει πάνω σε αυτές προβλήματα storage, δικτυακές ανωμαλίες, ασυνήθιστους κύκλους ενημέρωσης και περίεργες ροές Host Profiles. Αυτό από μόνο του αποτελεί ένα βήμα προς πιο σοβαρή παρακολούθηση της υποδομής.

Όσο για τη μελλοντική εργασία, ένα πρώτο προφανές βήμα είναι η εφαρμογή της ίδιας μεθοδολογίας σε περισσότερους hosts και σε μεγαλύτερα χρονικά διαστήματα. Ένα δεύτερο βήμα είναι η ενσωμάτωση ανθρώπινου feedback. Ο διαχειριστής μπορεί να χαρακτηρίζει ορισμένες ανωμαλίες ως ουσιαστικά συμβάντα και άλλες ως αθώο θόρυβο. Αυτές οι κρίσεις μπορούν να τροφοδοτήσουν ένα δεύτερο, επιβλεπόμενο επίπεδο μάθησης, που θα μειώσει σταδιακά τους ψευδείς συναγερμούς και θα ενισχύσει τα πραγματικά χρήσιμα alerts.

Υπάρχει, τέλος, χώρος για πειραματισμό με νέα μοντέλα και πιο στενή σύνδεση με εργαλεία παραγωγής. Autoencoders, πιο σύνθετα δίκτυα ακολουθιών, ακόμα και μοντέλα που ενσωματώνουν πληροφορία από διαφορετικά συστήματα μαζί. Παράλληλα, ολόκληρο το pipeline μπορεί να ενωθεί με συστήματα παρακολούθησης και ειδοποίησης που ήδη χρησιμοποιούνται σε παραγωγή, ώστε οι λίστες ανωμαλιών να μη μένουν μόνο στο χαρτί, αλλά να μετατρέπονται σε ζωντανά alerts με ιστορικό, σχολιασμό και συνεχή βελτίωση. Αν κάτι έδειξε καθαρά η εργασία αυτή, είναι ότι τα logs δεν είναι απλώς θόρυβος, αλλά μια πηγή πληροφορίας και το επόμενο βήμα είναι να περάσει αυτή η λογική από τα πειράματα στην καθημερινή λειτουργία.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

[1] VMware. (n.d.). VMware vSphere | Virtualization Technology. VMware.

[2] Broadcom. (2025, March 16). Location of ESXi log files (Knowledge Base Article). Broadcom Support.

[3] Broadcom. (2025, January 28). Locating virtual machine log files on an ESXi host (Knowledge Base Article). Broadcom Support.

[4] Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2022). Deep Learning for Anomaly Detection in Log Data: A Survey (arXiv:2207.03820). arXiv.

[5] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). ACM.

[6] He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. In Proceedings of the IEEE International Conference on Web Services (ICWS 2017).

[7] Khan, Z. A., Shin, D., Bianculli, D., & Briand, L. (2024). Impact of log parsing on deep learning-based anomaly detection: A comprehensive empirical study. Empirical Software Engineering.

[8] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. 2008 Eighth IEEE International Conference on Data Mining, 413–422.

[9] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471.

[10] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). ACM.

[11] He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. In Proceedings of the IEEE International Conference on Web Services (ICWS 2017). IEEE.

[12] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.

[13] Halko, N., Martinsson, P.-G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review, 53(2), 217–288.

[14] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734.

[15] Xu, W., Huang, L., Fox, A., Patterson, D., & Jordan, M. (2009). Detecting large-scale system problems by mining console logs. Proceedings of the ACM SIGOPS Symposium on Operating Systems Principles (SOSP).

[16] Oliner, A. J., Ganapathi, A., & Xu, W. (2012). Advances and challenges in log analysis. Communications of the ACM, 55(2), 55–61.

[17] Landauer, M., Skopik, F., Wurzenberger, M., Rauber, A., & Schrittwieser, S. (2023). Deep learning for anomaly detection in log data: A survey. Software Impacts, 15, 100463.

[18] Zhu, J., He, P., Fu, Q., Zhang, H., Lyu, M. R., & Zhang, D. (2015). Learning to parse log messages. Proceedings of the International Conference on Software Engineering (ICSE).

[19] He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. Proceedings of the IEEE International Conference on Web Services (ICWS).

[20] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3).

[21] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS ’17).

[22] Le, V. H., Zhang, H., & others. (2022). Log-based anomaly detection: A survey. ACM Computing Surveys.

[23] Khan, Z. A., Shin, D., Bianculli, D., & Briand, L. (2024). Impact of log parsing on deep learning-based anomaly detection: An empirical study. Empirical Software Engineering.

[24] Zhang, Q., Chen, M., Li, L., & Zhou, Z. (2020). Anomaly detection for cloud systems: A survey. Journal of Cloud Computing.

[25] Broadcom. (2025). Location of ESXi and virtual machine log files. Broadcom Support Documentation.

[26] Casas, Pedro & Fiadino, Pierdomenico & D'Alconzo, Alessandro. (2016). Machine-Learning Based Approaches for Anomaly Detection and Classification in Cellular Networks.

[27] Fulp, Errin & Fink, Glenn & Haack, Jereme. (2008). Predicting Computer System Failures Using Support Vector Machines.

[28] Farzad, A., & Gulliver, T. A. (2020). Oversampling log messages using a sequence generative adversarial network for anomaly detection and classification. In 9th International Conference on Information Technology Convergence and Services (ITCSE 2020) (pp. 163–175). AIRCC Publishing Corporation. https://doi.org/10.5121/csit.2020.100515

[29] Naeem, Samreen & Ali, Aqib & Anam, Sania & Ahmed, Munawar. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. IJCDS Journal. 13. 911-921. 10.12785/ijcds/130172.

[30] Yin, H., Aryani, A., Petrie, S., Nambissan, A., Astudillo, A., & Cao, S. (2024). A rapid review of clustering algorithms. arXiv. https://arxiv.org/abs/2401.07389

[31] Πήγη: https://www.geeksforgeeks.org/machine-learning/k-means-clustering-introduction/

[32] Πηγή: https://www.kaggle.com/discussions/questions-and-answers/422077)

[33] Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). Computers & Geosciences, 19(3), 303–342. https://doi.org/10.1016/0098-3004(93)90090-R

[34] Juvonen, Antti & Hamalainen, Timo. (2014). An Efficient Network Log Anomaly Detection System Using Random Projection Dimensionality Reduction. 1-5. 10.1109/NTMS.2014.6814006.

[35] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. 2013. Enhancing one-class support vector machines for unsupervised anomaly detection. In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description (ODD '13). Association for Computing Machinery, New York, NY, USA, 8–15. https://doi.org/10.1145/2500853.2500857

[36] Chua, W., Pajas, A. L. D., Castro, C. S., Panganiban, S. P., Pasuquin, A. J., Purganan, M. J., Malupeng, R., Pingad, D. J., Orolfo, J. P., Lua, H. H., & Velasco, L. C. (2024). Web Traffic Anomaly Detection Using Isolation Forest. Informatics, 11(4), 83. https://doi.org/10.3390/informatics11040083

[37] Arshi, M & Nasreen, MD & Karanam, Madhavi. (2020). A Survey of DDOS Attacks Using Machine Learning Techniques. E3S Web of Conferences. 184. 01052. 10.1051/e3sconf/202018401052.

[38] Dani, Mohamed & Doreau, Henri & Alt, Samantha. (2017). K-means Application for Anomaly Detection and Log Classification in HPC. 201-210. 10.1007/978-3-319-60045-1_23.

[39] Ali, S., Boufaied, C., Bianculli, D. et al. A comprehensive study of machine learning techniques for log-based anomaly detection. Empir Software Eng 30, 129 (2025). https://doi.org/10.1007/s10664-025-10669-3

[40] Lv, D., Luktarhan, N., & Chen, Y. (2021). ConAnomaly: Content-Based Anomaly Detection for System Logs. Sensors, 21(18), 6125. https://doi.org/10.3390/s21186125

[41] Zihan Wang, Yujia Zhou, Yiteng Tu, and Zhicheng Dou. 2023. NOVO: Learnable and Interpretable Document Identifiers for Model-Based IR. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23). Association for Computing Machinery, New York, NY, USA, 2656–2665. https://doi.org/10.1145/3583780.3614993

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.