

School of Science and Technology

Collective Artificial Intelligence and Machine Learning with resource constraints

Ph.D. Thesis

Zoi Lygizou

Supervisor: Prof. Dimitris Kalles

Patras, Greece

June, 2026

Theses / Dissertations remain the intellectual property of students (“authors/creators”), but in the context of open access policy they grant to the HOU a non-exclusive license to use the right of reproduction, customization, public lending, presentation to an audience and digital dissemination thereof internationally, in electronic form and by any means for teaching and research purposes, for no fee and throughout the duration of intellectual property rights. Free access to the full text for studying and reading does not in any way mean that the author/creator shall allocate his/her intellectual property rights, nor shall he/she allow the reproduction, republication, copy, storage, sale, commercial use, transmission, distribution, publication, execution, downloading, uploading, translating, modifying in any way, of any part or summary of the dissertation, without the explicit prior written consent of the author/creator. Creators retain all their moral and property rights.



Collective Artificial Intelligence and Machine Learning with resource constraints

Zoi Lygizou

Supervising Committee

Supervisor:

Dimitris Kalles

Professor

School of Science and Technology,

Hellenic Open University

Co-Supervisor:

Ioannis Hatzilygeroudis

Professor Emeritus

Dept. of Computer Engineering and Informatics,

University of Patras

Co-Supervisor:

Ioannis Stamatiou

Professor

Dept. of Business Administration

University of Patras

Patras, Greece, June, 2026

Examination Committee

1. **Dimitris Kalles**, Professor, School of Science and Technology, Hellenic Open University (**supervisor**)
2. **Ioannis Hatzilygeroudis**, Professor Emeritus, Department of Computer Engineering and Informatics, University of Patras (member of the **advisory committee**)
3. **Ioannis Stamatiou**, Professor, Department of Business Administration, University of Patras (member of the **advisory committee**)
4. **Vassilios S. Verykios,** Professor, School of Science and Technology, Hellenic Open University
5. **Sarantis Mitropoulos**, Professor, School of Science and Technology, Hellenic Open University
6. **Nikolaos Karousos**, Assistant Professor, Department of Management Science and Technology, University of Patras
7. **Chairi Kiourt**, Principal Researcher, ATHENA – Research and Innovation Centre in Information, Communication and Knowledge Technologies

Acknowledgements

The completion of this doctoral dissertation marks the end of a long and demanding journey, which would not have been possible without the support, guidance, and encouragement of several people.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Dimitris Kalles. Throughout the years of this research, he provided invaluable scientific guidance, constructive criticism, continuous encouragement, and unwavering support. His trust in my abilities, his willingness to discuss new ideas, and the academic freedom he offered me were essential for the successful completion of this work.

I would also like to thank the members of my advisory committee for their time, interest and valuable comments.

My sincere appreciation extends to the academic community of the Hellenic Open University, which provided a stimulating environment for research and learning throughout my doctoral studies.

Finally, and most importantly, I would like to thank my family for their unconditional love, patience, encouragement, and support. Their confidence in me gave me the strength to continue during the most challenging moments of this journey.

This dissertation is dedicated to my parents, my family, and my teachers, whose guidance and influence have shaped both my academic path and my life.

To my parents, my family, and my teachers

The prudent person considers what is appropriate to the particular circumstances.
Aristotle, *Nicomachean Ethics (paraphrased)*

Abstract

Trust constitutes a fundamental mechanism for decision making in open and dynamic Multi-Agent Systems, where autonomous agents must identify reliable partners under conditions of uncertainty, incomplete information, and continuously changing environments. Although computational trust and reputation models have been extensively studied and successfully applied in numerous domains, important challenges remain unresolved, including the cold-start problem, agent mobility, behavioral volatility, communication overhead, vulnerability to disinformation attacks, and privacy concerns, and resource constraints in terms of computational, communication, and decision-making costs.

This dissertation investigates alternative approaches to trust management in dynamic Multi-Agent Systems and introduces a trustee-centric perspective on computational trust. Inspired by the biological mechanisms of synaptic plasticity and neural assembly formation, a novel trust model, named *Create Assemblies* (CA), is proposed. Unlike conventional trust-management approaches, where trustors evaluate and select trustees, the proposed model enables trustees to actively participate in the trust-formation process by assessing their own suitability for performing requested tasks.

The proposed approach is evaluated through a series of simulation experiments and compared against FIRE, a well-known trust and reputation model for open Multi-Agent Systems. The results demonstrate that the trustee-centric approach exhibits distinct advantages under specific environmental conditions, particularly in scenarios characterized by changes in trustor populations, while conventional trustor-centric approaches remain advantageous in other situations. These findings indicate that no single trust-management mechanism consistently outperforms all alternatives across all operating conditions.

To address this limitation, the dissertation further investigates adaptive trust management. First, a self-assessment mechanism is incorporated into the CA model, enabling trustees to detect changes in their own performance and adjust their behavior accordingly. Second, the problem of trust model selection is formulated as a reinforcement learning task. A Deep Q-Learning-based adaptable trustor is developed, capable of dynamically selecting between alternative trust mechanisms according to observed

environmental conditions. Experimental results demonstrate that adaptive trust model selection can achieve robust performance across a wide range of dynamic environments and often achieves or exceeds the performance of the best individual trust model.

Beyond performance evaluation, the dissertation examines the broader properties of the proposed trustee-centric paradigm, including privacy preservation, resistance to reputation manipulation attacks, communication efficiency, explainability, and scalability. Furthermore, the applicability of the approach beyond traditional trust-management problems is explored through the development of a simulation framework for emergency department operations, where trust-informed self-assessment and task allocation mechanisms are employed as decision-support tools.

The findings of this dissertation suggest that trust management should not be viewed as a competition between alternative trust models seeking a universally superior solution. Instead, different trust mechanisms exhibit complementary strengths under different environmental conditions. The proposed trustee-centric approach provides a complementary perspective to conventional trustor-centric models, while reinforcement learning offers a mechanism for dynamically selecting the most appropriate trust strategy according to prevailing circumstances. Together, these findings contribute towards the development of adaptive, robust, and intelligent trust-management frameworks capable of operating effectively in complex, open, and continuously evolving environments under resource-constrained conditions where decision quality must be balanced against computational and communication cost.

Keywords

Computational Trust, Open Multi-Agent Systems, Trustee-Centric Trust Management, Biologically Inspired Algorithms, Adaptive Trust Management, Reinforcement Learning, Deep Q-Learning, Decision Support Systems, Simulation-Based Decision Support, Healthcare Management, Emergency Department (ED)

Συλλογική Τεχνητή Νοημοσύνη και Μηχανική Μάθηση προσανατολισμένη σε συνθήκες έλλειψης πόρων

Ζωή Λυγίζου

Περίληψη

Η εμπιστοσύνη αποτελεί θεμελιώδη μηχανισμό λήψης αποφάσεων σε ανοικτά και δυναμικά πολυπρακτορικά συστήματα, όπου αυτόνομοι πράκτορες καλούνται να εντοπίσουν αξιόπιστους συνεργάτες υπό συνθήκες αβεβαιότητας, ελλιπούς πληροφόρησης και συνεχώς μεταβαλλόμενων περιβαλλόντων. Παρά τη σημαντική πρόοδο που έχει σημειωθεί στον τομέα των υπολογιστικών μοντέλων εμπιστοσύνης και φήμης, εξακολουθούν να υφίστανται σημαντικές προκλήσεις όπως το πρόβλημα της ψυχρής εκκίνησης (cold-start problem), η κινητικότητα των πρακτόρων, η μεταβλητότητα της συμπεριφοράς τους, το επικοινωνιακό και υπολογιστικό κόστος, οι επιθέσεις παραπληροφόρησης και οι ανησυχίες που σχετίζονται με την προστασία της ιδιωτικότητας.

Η παρούσα διατριβή διερευνά εναλλακτικές προσεγγίσεις στη διαχείριση της εμπιστοσύνης σε δυναμικά πολυπρακτορικά συστήματα και εισάγει μια νέα οπτική γωνία, προσανατολισμένη στον trustee. Εμπνευσμένο από τους βιολογικούς μηχανισμούς της συναπτικής πλαστικότητας και του σχηματισμού συναθροίσεων νευρικών κυττάρων, προτείνεται ένα νέο υπολογιστικό μοντέλο εμπιστοσύνης με την ονομασία *Create Assemblies* (CA). Σε αντίθεση με τα συμβατικά μοντέλα, όπου ο trustor αξιολογεί και επιλέγει τον trustee, το προτεινόμενο μοντέλο επιτρέπει στους trustees να συμμετέχουν ενεργά στη διαδικασία σχηματισμού εμπιστοσύνης, αξιολογώντας οι ίδιοι την καταλληλότητά τους για την εκτέλεση μιας ζητούμενης εργασίας.

Η προτεινόμενη προσέγγιση αξιολογείται μέσω εκτεταμένων πειραματικών προσομοιώσεων και συγκρίνεται με το FIRE, ένα καθιερωμένο μοντέλο εμπιστοσύνης και φήμης για ανοικτά πολυπρακτορικά συστήματα. Τα αποτελέσματα δείχνουν ότι η trustee-

centric προσέγγιση παρουσιάζει σημαντικά πλεονεκτήματα σε συγκεκριμένες περιβαλλοντικές συνθήκες, ιδιαίτερα όταν μεταβάλλεται ο πληθυσμός των trustors, ενώ οι συμβατικές trustor-centric προσεγγίσεις παραμένουν πιο αποτελεσματικές σε άλλες περιπτώσεις. Τα ευρήματα αυτά καταδεικνύουν ότι δεν υπάρχει ένα μοναδικό μοντέλο εμπιστοσύνης που να υπερέχει συστηματικά σε όλα τα περιβάλλοντα λειτουργίας. Επιπλέον, το πρόβλημα εξετάζεται υπό συνθήκες περιορισμένων πόρων, όπου η λήψη αποφάσεων πραγματοποιείται με περιορισμένη διαθεσιμότητα υπολογιστικών, επικοινωνιακών και επιχειρησιακών πόρων.

Στη συνέχεια, η διατριβή εξετάζει μηχανισμούς προσαρμοστικής διαχείρισης της εμπιστοσύνης. Αρχικά, ενσωματώνεται στο μοντέλο CA ένας μηχανισμός αυτοαξιολόγησης, ο οποίος επιτρέπει στους trustees να ανιχνεύουν μεταβολές στην απόδοσή τους και να προσαρμόζουν ανάλογα τη συμπεριφορά τους. Επιπλέον, το πρόβλημα της επιλογής μοντέλου εμπιστοσύνης διατυπώνεται ως πρόβλημα ενισχυτικής μάθησης. Για το σκοπό αυτό αναπτύσσεται ένας προσαρμοστικός trustor βασισμένος σε Deep Q-Learning, ο οποίος είναι σε θέση να επιλέγει δυναμικά μεταξύ διαφορετικών μηχανισμών εμπιστοσύνης ανάλογα με τις συνθήκες του περιβάλλοντος. Τα πειραματικά αποτελέσματα δείχνουν ότι η προσαρμοστική επιλογή μοντέλου εμπιστοσύνης επιτυγχάνει σταθερά υψηλή απόδοση σε ένα ευρύ φάσμα δυναμικών περιβαλλόντων και συχνά προσεγγίζει ή υπερβαίνει την απόδοση του καλύτερου επιμέρους μοντέλου.

Πέρα από την αξιολόγηση της απόδοσης, η διατριβή εξετάζει ευρύτερες ιδιότητες του προτεινόμενου trustee-centric παραδείγματος, όπως η προστασία της ιδιωτικότητας, η ανθεκτικότητα σε επιθέσεις χειραγώγησης φήμης, η αποδοτικότητα επικοινωνίας, η επεξηγησιμότητα και η δυνατότητα για κλιμάκωση. Τέλος, διερευνάται η εφαρμογή της προσέγγισης πέρα από τα παραδοσιακά προβλήματα διαχείρισης εμπιστοσύνης, μέσω της ανάπτυξης ενός πλαισίου προσομοίωσης για τη λειτουργία τμημάτων επειγόντων περιστατικών, όπου μηχανισμοί αυτοαξιολόγησης και ανάθεσης εργασιών αξιοποιούνται ως εργαλεία υποστήριξης αποφάσεων.

Τα αποτελέσματα της διατριβής υποδεικνύουν ότι η διαχείριση της εμπιστοσύνης δεν θα πρέπει να αντιμετωπίζεται ως αναζήτηση ενός καθολικά βέλτιστου μοντέλου. Αντίθετα, διαφορετικοί μηχανισμοί εμπιστοσύνης εμφανίζουν διαφορετικά πλεονεκτήματα ανάλογα με τις συνθήκες του περιβάλλοντος. Η trustee-centric προσέγγιση λειτουργεί συμπληρωματικά προς τις συμβατικές trustor-centric προσεγγίσεις,

ενώ η ενισχυτική μάθηση παρέχει ένα αποτελεσματικό μηχανισμό δυναμικής επιλογής της καταλληλότερης στρατηγικής εμπιστοσύνης. Συνολικά, η διατριβή συμβάλλει στην ανάπτυξη προσαρμοστικών, ανθεκτικών και ευφυών πλαισίων διαχείρισης εμπιστοσύνης για σύνθετα και συνεχώς εξελισσόμενα υπολογιστικά περιβάλλοντα.

Λέξεις – Κλειδιά

Υπολογιστική Εμπιστοσύνη, Ανοικτά Πολυπρακτορικά Συστήματα, Trustee-Centric Διαχείριση Εμπιστοσύνης, Βιολογικά Εμπνευσμένοι Αλγόριθμοι, Προσαρμοστική Διαχείριση Εμπιστοσύνης, Ενισχυτική Μάθηση, Deep Q-Learning, Συστήματα Υποστήριξης Αποφάσεων, Διαχείριση Υπηρεσιών Υγείας, Τμήμα Επειγόντων Περιστατικών

Dissemination

Based on this PhD thesis, the following papers have been published:

Papers in Journals

1. Lygizou, Z.; Kalles, D. Using Deep Q-Learning to Dynamically Toggle between Push/Pull Actions in Computational Trust Mechanisms. *Machine Learning and Knowledge Extraction* **2024**, *6*, 1413–1438.
2. Lygizou, Z.; Kalles, D. A Biologically Inspired Trust Model for Open Multi-Agent Systems That Is Resilient to Rapid Performance Fluctuations. *Applied Sciences* **2025**, *15*, 6125.
3. Lygizou, Z.; Kalles, D. From Coordination to Personalization: A Trust-Aware Simulation Framework for AI-Driven Personalized Decision Support in Emergency Departments. *Journal of Personalized Medicine* **2025**, *15*, 574.

Papers in Conferences

1. Lygizou, Z.; Kalles, D. A Biologically Inspired Computational Trust Model Based on the Perspective of the Trustee. In Proceedings of the Proceedings of the 12th Hellenic Conference on Artificial Intelligence; ACM: Corfu Greece, September 7 2022; pp. 1–10.
2. Lygizou, Z.; Kalles, D. A Biologically Inspired Computational Trust Model for Open Multi-Agent Systems Which Is Resilient to Trustor Population Changes. In Proceedings of the Proceedings of the 13th Hellenic Conference on Artificial Intelligence; ACM: Piraeus Greece, September 11 2024; pp. 1–9.

Table of Contents

Abstract viii
Περίληψη x
Dissemination xiii
Table of Contents xiv
List of Figures xxiii
List of Tables xxviii
1. Introduction 1
1.1. Computational Trust in Open Multi-Agent Systems 1
1.1.1. Open Multi-Agent Systems and the Need for Trust 1
1.1.2. Conceptual Foundations of Computational Trust 3
1.1.3. Trust Representation and Trust Formation 4
1.1.4. Computational Trust Models and Research Challenges 6
1.2. Limitations of Existing Trust Management Approaches 8
1.2.1. The Cold Start Problem 8
1.2.2. Highly dynamic environments and Agent Mobility 9
1.2.3. Behavioral volatility 10
1.2.4. Communication and computational overhead 12
1.2.5. Disinformation attacks 13
1.2.5.1. Mechanisms for Promoting Honest Behavior 14
1.2.6. Privacy concerns 20
1.2.7. Summary of Limitations and Research Motivation 20
1.3. Trustee-Centric Perspective on Trust Management 21
1.4. Adaptive Trust Management 24
1.5. Beyond Trust Management: Decision Support and Healthcare Applications 25
1.5.1. Simulation of Emergency Departments 27
1.5.2. Decision Support Systems in Healthcare 29
1.5.3. Computational Trust Models 30
1.5.4. Personalized Medicine and Adaptive Staffing 31
1.6. Research Objectives and Research Questions 33

1.7. Dissertation Structure 35
2. The Create Assemblies (CA) Trust Model 37
2.1. Biological Inspiration 37
2.2. From Synaptic Plasticity to Trustee-Centric Trust Modeling 38
2.3. Mathematical Formulation 40
2.3.1. Task and Task Categories 40
2.3.2. Trust Connections 41
2.3.3. Decision Threshold 42
2.3.4. Trust Update Mechanism 42
2.4. Reward-Punishment Mechanism 43
2.5. The Create Assemblies Algorithm 44
2.6. Key Advantages of Trustee-Centric Trust Management 47
2.6.1. Mobility and Reusability of Trust Information 47
2.6.2. Reduced Communication Overhead 47
2.6.3. Resistance to Disinformation Attacks 47
2.6.4. Enhanced Privacy Preservation 48
3. Comparative Evaluation of Trustor-Centric and Trustee-Centric Trust Models 49
3.1. The FIRE Trust and Reputation model 49
3.2. Extending CA for Dynamic Trustee Profiles 50
3.3. Experimental Environment 52
3.4. Experimental Methodology 56
3.4.1. Performance Measure 57
3.4.2. Statistical Analysis 57
3.4.3. Presentation of Results 58
3.4.4. Experimental Parameters 58
3.5. Experimental Scenarios and Results 60
3.5.1. Baseline Comparison in a Static Environment 60
3.5.2. CA's vs. FIRE's Performance under Provider Population Changes 61
3.5.3. CA's vs. FIRE's Performance under Consumer Population Changes 65
3.5.4. The Effect of Concurrent Consumer and Provider Population Changes 68
3.5.5. CA's vs. FIRE's performance in dynamic trustee profiles 69

3.6. Explaining the Performance Differences Between CA and FIRE 71
3.7. Conclusions 72
3.8. Research Directions 73
4. Adaptive Trust Model Selection in Dynamic Multi-Agent Systems 75
4.1. Adaptive Trust Model Selection as a Reinforcement Learning Problem 76
4.2. Deep Q-Network Architecture 77
4.3. State Representation and Environmental Features 78
4.3.1. Available Local Information in a Partially Observable Environment 79
4.3.2. Environmental Features 80
4.3.3. Features Related to Provider Population Changes 81
4.3.3.1. ProvidersPopulationDirectChangeEstimate 81
4.3.3.2. MeanProvidersPopulationDirectChangeEstimate 82
4.3.3.3. ProvidersPopulationIndirectChangeEstimate 82
4.3.3.4. MeanProvidersPopulationIndirectChangeEstimate 83
4.3.4. Features Related to Consumer Population Changes 83
4.3.4.1. NewcomerEstimate 84
4.3.4.2. ConsumersPopulationIndirectChangeEstimate 84
4.3.4.3. MeanConsumersPopulationIndirectChangeEstimate 85
4.3.5. Features Related to Performance and Mobility Changes 85
4.3.5.1. ProvidersPerformanceChangeEstimate 85
4.3.5.2. ConsumersLocationChangeEstimate 86
4.4. Hyperparameter Configuration and Training Setup 87
4.5. Experimental Methodology 89
4.5.1. Consumer Groups 89
4.5.2. Number of Independent Simulation Runs 89
4.5.3. Simulation Rounds 90
4.6. Experimental Scenarios and Results 90
4.6.1. Adaptable's Performance under Single Environmental Changes 91
4.6.2. Adaptable's Performance under Combined Environmental Changes 98

4.6.3. Adaptable's Performance under Time-Varying Environmental Changes 103
4.7. Explaining the Performance of the Adaptable Trustor 104
4.7.1. Why the Adaptable Trustor Typically Performs Between FIRE and CA 105
4.7.2. Cases where the Adaptable Trustor Outperforms Both FIRE and CA 106
4.7.3. The Impact of Severe Environmental Dynamics 107
4.8. Conclusions 107
4.9. Research Directions 109
4.9.1. Improved Exploration Strategies 109
4.9.2. Multi-Agent Reinforcement Learning 109
4.9.3. Hyperparameter Optimization and Resource Constraints 109
4.9.4. Trust Disinformation and Malicious Information Sources 110
4.9.5. Adaptive Selection Among Multiple Trust Models 110
4.9.6. Real-World Validation and Application Domains 110
5. Improving CA for Dynamic Trustee Performance Changes 112
5.1. Semi-Formal Analysis of the Original CA Algorithm 112
5.1.1. Analysis under Different Task Difficulty Levels 113
5.1.1.1. Phase A: Provider $\boldsymbol{BP1}$ learns its incapability in performing $\boldsymbol{task1}$ 115
5.1.1.2. Phase B: Provider $\boldsymbol{BP1}$ learns its incapability in performing $\boldsymbol{task2}$. 116
5.1.1.3. Phase C: Provider $\boldsymbol{BP1}$ learns its incapability in performing $\boldsymbol{task3}$. 116
5.1.1.4. Phase D: Provider $\boldsymbol{BP1}$ learns its incapability in performing $\boldsymbol{task4}$ 118
5.1.1.5. Phase E: Provider $\boldsymbol{BP1}$ learns its capability in performing $\boldsymbol{task5}$. 118
5.2. The Proposed CA Algorithm for the Early Detection of Low-Performance Providers 118
5.3. Experimental Methodology 122

5.3.1. Consumer Groups....122
5.3.2. Statistical Evaluation....123
5.4. Experimental Scenarios and Results....123
5.4.1. The Performance of the New CA Algorithm in Dynamic Trustee Profiles....124
5.4.2. The performance of the New CA Algorithm in the Static Setting....125
5.4.3. The Performance of the New CA Algorithm in Provider Population Changes....125
5.4.4. The Performance of the New CA Algorithm in Consumer Population Changes....130
5.4.5. The Performance of the New CA Algorithm in Consumer and Provider Location Changes....132
5.4.6. The Performance of the New CA Algorithm Under the Effects of All Dynamic Factors....133
5.4.7. An Overview of the Results....134
5.5. Conclusions....135
5.6. Research Directions....136
6. Towards a Comprehensive Evaluation of the Create Assemblies Trust Model 138
6.1. Evaluation Criteria....138
6.1.1. Decentralization....139
6.1.2. Subjectivity....139
6.1.3. Context Awareness....140
6.1.4. Dynamicity....140
6.1.5. Availability....140
6.1.6. Integrity and Transparency....140
6.1.7. Scalability....141
6.1.8. Overhead....141
6.1.9. Accuracy....142
6.1.10. Robustness....142
6.1.11. Privacy Preservation....142
6.1.12. Explainability....143
6.1.13. User Acceptance....143

6.2. CA's Resilience to Trust-Related Attacks 144
6.2.1. Malicious with Everyone (ME) 144
6.2.2. Bad-mouthing Attack (BMA) 144
6.2.3. Ballot Stuffing (BSA) or Good-mouthing Attack 145
6.2.4. Self-promoting Attack 146
6.2.5. Opportunistic Service Attack (OSA) 146
6.2.6. Sybil Attack (SA) 146
6.2.7. Whitewashing Attack (WA) 147
6.2.8. On-Off Attack (OOA) 147
6.2.9. Discrimination or Selective Misbehavior Attack 147
6.2.10. Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks 148
6.2.11. Storage Attack 148
6.3. Potential Mechanisms for Enhancing Attack Resilience 148
6.3.1. Incentive Mechanisms 148
6.3.2. Penalty Mechanisms 149
6.3.3. Contract-Theoretic Mechanisms 149
6.3.4. Consistency-Checking Mechanisms 150
6.4. Conclusions 150
7. Trust-Based Decision Support for Emergency Department Operations: A Simulation Framework Case Study Based on the CA Model 152
7.1. Simulation Framework 152
7.1.1. Overview of the Framework 152
7.1.2. Agents and Roles 154
7.1.2.1. Doctors 154
7.1.2.2. Nurses 154
7.1.2.3. Trainer Nurses 155
7.1.2.4. Patients 155
7.1.3. Modeling Staff Heterogeneity 155
7.1.3.1. Doctor Evaluation Styles 156
7.1.3.2. Nurse Quality Attributes 156

7.1.3.3. Role of Heterogeneity in the Simulation 157
7.1.4. Emergency Department Layout 157
7.1.4.1. Patient Treatment Area 158
7.1.4.2. Staff Waiting Area 158
7.1.4.3. Department Exit 159
7.1.4.4. Agent Navigation and Visualization 159
7.1.4.5. Simplifications and Future Extensions 159
7.1.5. Workflow Description 159
7.1.5.1. Base Workflow 160
7.1.5.2. Replacement Workflow 160
7.1.5.3. Training Workflow 161
7.1.6. Logged Metrics 162
7.1.6.1. Number of patients served 162
7.1.6.2. Total time damage 162
7.1.6.3. Total delay 163
7.1.6.4. Task success/failure counts 163
7.1.6.5. Utility values for each nurse 164
7.1.6.6. Doctors' delays and evaluation accuracy 164
7.1.7. Customizability and Reproducibility 164
7.2. Decision-Support Experimentation Methodology 164
7.2.1. Objectives and Rationale 164
7.2.2. Experimental Design 165
7.2.2.1. Scenarios 165
7.2.2.2. Experimental Variables 165
7.2.3. Experimental Procedure 166
7.2.4. Data Analysis Approach 167
7.3. Results 167
7.3.1. Case Study Example 167

7.3.2. Key Findings 168
7.3.2.1. Baseline comparison (CA trust vs. FIFO) 168
7.3.2.2. Replacement Scenario 168
7.3.2.3. Training Scenario 168
7.3.2.4. Replacement vs. Training 169
7.3.3. Interpretation and Limitations 169
7.3.4. Summary of Findings 169
7.4. Conclusions 170
7.5. Research Directions 171
7.5.1. Learning Agent Behavior Models from Logged Performance Data 171
7.5.1.1. Inferring Doctor Evaluation Styles 171
7.5.1.2. Inferring Nurse Quality Attributes 171
7.5.2. Environment and Workflow Customization 172
7.5.3. Integrating a Triage Mechanism for Task Prioritization 172
7.5.4. Expanding the Range of Nursing Tasks 172
7.5.5. Towards AI-Driven Personalized Healthcare Operations 173
8. Conclusions and Future Research 174
8.1. Overview of the Dissertation 174
8.2. Main Research Contributions 175
8.3. Limitations of the Research 178
8.4. Future Research Directions 180
8.4.1. Adaptive and Self-Aware Trustees 180
8.4.2. Expanding Reinforcement Learning-Based Trust Selection 181
8.4.3. Learning-Based Trustee Self-Assessment 181
8.4.4. Real-World Validation and Deployment 182
8.4.5. Trust-Informed Decision Support Systems 182
8.4.6. Toward Integrated Trust Management Frameworks 182
Appendix A. Synaptic Plasticity 184
A.1. Action Potential 184
A.2. Synaptic Plasticity 186
Appendix B. Reinforcement Learning Background 191

B.1. Reinforcement Learning 191
B.2. Q-Learning 192
B.3. Deep Q-Learning 193
B.4. Relation to Chapter 4 194
Appendix C. Selected Proof Sketches 195
Appendix D. Sketch of Computational Complexity Analysis of CA Algorithm 196
Appendix E. Algorithms 198
E.1. Doctor's performance level estimation based on evaluation style 198
E.2. Nurse task execution time determination based on quality and training status 198
Appendix F. Decision Support Case Study 200
F.1. Comparative Evaluation of Nurse Task Selection Methods in the Baseline Scenario 200
F.2. Baseline (CA trust model) vs. Replacement (CA trust model) 204
F.3. Baseline (CA trust model) vs. Training (CA trust model) 207
F.4. Replacement (CA trust model) vs. Training (CA trust model) 212
References 217

List of Figures

Figure 1. Synaptic plasticity. An action potential in a presynaptic neuron leads to neurotransmitter release and receptor activation in a postsynaptic neuron. Coordinated firing of the two neurons strengthens the synapse by increasing the number of postsynaptic receptors. As a result, stronger synaptic connections are more likely to produce coordinated activation in the future. 38

Figure 2. The Trustee stores, for each Trustor and for each requested task, a connection with a weight representing the trust value 41

Figure 3. Experiment 1: CA versus FIRE performance in a static environment. ... 61

Figure 4, Experiment 2. Provider population change $\boldsymbol{pppc = 2\%}$ 62

Figure 5. Experiment 3. Provider population change $\boldsymbol{pppc = 5\%}$ 63

Figure 6. Experiment 4. Provider population change $\boldsymbol{pppc = 10\%}$ 63

Figure 7. Impact of increasing provider population change on CA performance ... 64

Figure 8. Impact of increasing provider population change on FIRE performance 64

Figure 9. Experiment 5. Consumer population change $\boldsymbol{pCPC = 2\%}$ 66

Figure 10. Experiment 6. Consumer population change $\boldsymbol{pCPC = 5\%}$ 66

Figure 11. Experiment 7. Consumer population change $\boldsymbol{pCPC = 10\%}$ 67

Figure 12. CA performance under increasing consumer population change rates .. 67

Figure 13. Experiment 8. Concurrent population changes $\boldsymbol{pPPC = 2\%, pCPC = 5\%}$ 68

Figure 14. Experiment 9. Concurrent population changes $\boldsymbol{pPPC = 10\%, pCPC = 10\%}$ 69

Figure 15. Experiment 10. Providers change their performance $\boldsymbol{p\mu C = 10\%, M = 1.0}$ 70

Figure 16. Experiment 11. Providers switch performance profiles $\boldsymbol{pProfileSwitch = 2\%}$ 71

Figure 17. Experiment 1: performance of the adaptable consumer in the static setting 92

Figure 18. Experiment 2: provider population change; $\boldsymbol{pPPC = 2\%}$ 93

Figure 19. Experiment 3: provider population change; $\boldsymbol{pPPC = 5\%}$ 93

Figure 20. Experiment 4: provider population change; $\boldsymbol{pPPC = 10\%}$ 94

Figure 21. Experiment 5: consumer population change; $\boldsymbol{pCPC = 2\%}$ 94
Figure 22. Experiment 6: consumer population change; $\boldsymbol{pCPC = 5\%}$ 95
Figure 23. Experiment 7: consumer population change; $\boldsymbol{pCPC = 10\%}$.....95
Figure 24. Experiment 8: providers change their performance; $\boldsymbol{pCPC = 10\%, M = 1.0}$.....96
Figure 25. Experiment 9: providers switch their profiles; $\boldsymbol{pProfileSwitch = 2\%}$ 96
Figure 26. Experiment 10: consumers change their locations; $\boldsymbol{pCLC = 10\%}$, $\boldsymbol{\Delta\Phi = \pi/20}$ 97
Figure 27. Experiment 11: providers change their locations; $\boldsymbol{pPLC = 10\%}$, $\boldsymbol{\Delta\Phi = \pi/20}$ 97
Figure 28. Experiment 12: $\boldsymbol{pPPC = 2\%}$ and $\boldsymbol{pCPC = 5\%}$.....100
Figure 29. Experiment 13: $\boldsymbol{pPPC = 10\%}$ and $\boldsymbol{pCPC = 10\%}$.....100
Figure 30. Experiment 14: $\boldsymbol{pPPC = 2\%}$, $\boldsymbol{pCPC = 5\%}$, $\boldsymbol{pCLC = 10\%}$, $\boldsymbol{\Delta\Phi = \pi/20}$. 101
Figure 31. Experiment 15: $\boldsymbol{pPPC = 2\%}$, $\boldsymbol{pCPC = 5\%}$, $\boldsymbol{pCLC = 10\%}$, $\boldsymbol{pPLC = 10\%}$, $\boldsymbol{\Delta\Phi = \pi/20}$.....101
Figure 32. Experiment 16: $\boldsymbol{pPPC = 2\%}$, $\boldsymbol{pCPC = 5\%}$, $\boldsymbol{pCLC = 10\%}$, $\boldsymbol{pPLC = 10\%}$, $\boldsymbol{\Delta\Phi = \pi/20}$, $\boldsymbol{p\mu C = 10\%}$, $\boldsymbol{M = 1.0}$.....102
Figure 33. Experiment 17: $\boldsymbol{pPPC = 2\%}$, $\boldsymbol{pCPC = 5\%}$, $\boldsymbol{pCLC = 10\%}$, $\boldsymbol{pPLC = 10\%}$, $\boldsymbol{\Delta\Phi = \pi/20}$, $\boldsymbol{p\mu C = 10\%}$, $\boldsymbol{M = 1.0}$, $\boldsymbol{pProfileSwitch = 2\%}$. 102
Figure 34. Experiment 18: Performance of Adaptable, FIRE, and CA consumer groups under time-varying environmental changes. 104
Figure 35. (a) A performance comparison of CA_NEW and CA_OLD, when providers change performance with 10% probability per round: CA_NEW achieves better performance, especially in early interactions. (b) A performance comparison of CA_NEW and FIRE, when providers change performance with 10% probability per round: CA_NEW excels in early interactions (except for the first interaction), while FIRE adapts and eventually achieves slightly better performance. 124
Figure 36. (a) A performance comparison of CA_NEW and CA_OLD when providers switch performance profiles with 2% probability per round: CA_NEW consistently outperforms CA_OLD by an average of approximately 2UG units. (b) A

performance comparison of CA_NEW and FIRE when providers switch performance profiles with 2% probability per round: CA_NEW outperforms FIRE in all interactions except for the first one. 125

Figure 37. (a) A performance comparison of CA_NEW and CA_OLD in a static environment: CA_NEW achieves better performance in early interactions. (b) A performance comparison of CA_NEW and FIRE in a static environment: CA_NEW outperforms FIRE in all interactions except for the first interaction. 125

Figure 38. (a) A performance comparison of CA_NEW and CA_OLD when the provider population changes with a 2% probability per round: CA_NEW outperforms CA_OLD in all interactions. (b) A performance comparison of CA_NEW and FIRE when the provider population changes with a 2% probability per round. 126

Figure 39. (a) A performance comparison of CA_NEW and CA_OLD when the provider population changes with a 5% probability per round: CA_NEW outperforms CA_OLD in all interactions. (b) A performance comparison of CA_NEW and FIRE when the provider population changes with 5% probability per round. 126

Figure 40. (a) A performance comparison of CA_NEW and CA_OLD when the provider population changes with a 10% probability per round: CA_NEW outperforms CA_OLD in all interactions. (b) A performance comparison of CA_NEW and FIRE when the provider population changes with a 10% probability per round. 127

Figure 41. A performance comparison of CA_NEW and FIRE when the provider population changes with a 20% probability per round. 127

Figure 42. A performance comparison of CA_NEW and FIRE when the provider population changes with a 30% probability per round. 128

Figure 43. (a) Performance comparison of CA_OLD when the provider population changes with probabilities of 2%, 5%, and 10%; performance drops by 4 UG units. (b) Performance comparison of CA_NEW when provider population changes with probabilities of 2%, 5%, and 10%; performance drops by 2 UG units. (c) Performance comparison of FIRE when provider population changes with probabilities of 2%, 5%, and 10%; FIRE is more resilient than CA_NEW to this environmental change. 129

Figure 44. (a) Performance comparison of CA_NEW and CA_OLD when consumer population changes with a probability of 2% per round. (b) Performance

comparison of CA_NEW and FIRE when consumer population changes with a probability of 2% per round. 130

Figure 45. (a) Performance comparison of CA_NEW and CA_OLD when the consumer population changes with a probability of 5% per round. (b) Performance comparison of CA_NEW and FIRE when the consumer population changes with a probability of 5% per round. 130

Figure 46. (a) Performance comparison of CA_NEW and CA_OLD when the consumer population changes with a probability of 10% per round. (b) Performance comparison of CA_NEW and FIRE when the consumer population changes with a probability of 10% per round. 131

Figure 47. (a) Performance comparison of CA_NEW in consumer population changes of 2%, 5%, and 10%. (b) Performance comparison of CA_OLD in consumer population changes of 2%, 5%, and 10%. 132

Figure 48. (a) A performance comparison of CA_NEW and CA_OLD when consumers change locations with a probability of 10% per round. (b) A performance comparison of CA_NEW and FIRE when consumers change locations with a probability of 10% per round. 132

Figure 49. (a) A performance comparison of CA_NEW and CA_OLD when providers change locations with a probability of 10% per round. (b) A performance comparison of CA_NEW and FIRE when providers change locations with a probability of 10% per round. 133

Figure 50. (a) Performance comparison of CA_NEW and CA_OLD when all dynamic factors are in effect: CA_NEW outperforms. (b) Performance comparison of CA_NEW and FIRE when all dynamic factors are in effect: CA_NEW outperforms. 134

Figure 51. CA Approach with Incentives 149

Figure 52. CA Approach with Penalties 149

Figure 53. CA Approach with Contracts 150

Figure 54. CA Approach with Consistency Checks 150

Figure 55. Simulation snapshot showing the bed zone (green area), staff waiting room (orange area), and the movement of agents. Red arrows indicate movement directions, while labels describe the current activity of agents. Dynamically updated statistics are displayed on the left side. 158

Figure 56: At rest, the concentration of positively charged ions is higher outside the cell than inside 184

Figure 57: When an action potential reaches the end of a presynaptic cell, it causes the release of neurotransmitters, which cross the synaptic cleft and bind to receptors on the postsynaptic cell 187

Figure 58: In the stronger connection, the number of receptors on the postsynaptic cell is higher, and therefore the degree of voltage change in the postsynaptic cell is greater, compared to the weaker connection 188

List of Tables

Table 1. Profiles of provider agents (performance constants defined in Table 2) ..53
Table 2. Performance level constants.....54
Table 3. Number of Independent Simulation Runs (NISR).....57
Table 4. General Simulation Parameters.....58
Table 5. Default FIRE parameters 59
Table 6. Default CA parameters.....59
Table 7: Hyperparameters and their values.....87
Table 8: Number of Independent Simulation Runs (NISR) per experiment.....89
Table 9: Number of simulation rounds 90
Table 10. Environmental changes applied in each simulation round. PPC stands for provider population change, and CPC stands for consumer population change. 103
Table 11. Explanation of requirements. 114
Table 12. Evolution of average connection weights after each execution of $\boldsymbol{task3}$ for consumer $\boldsymbol{C1}$ to $\boldsymbol{C4}$.....117

1. Introduction

1.1. Computational Trust in Open Multi-Agent Systems

1.1.1. Open Multi-Agent Systems and the Need for Trust

The increasing adoption of distributed and autonomous computational systems has led to the emergence of large-scale, open, and highly dynamic environments in which heterogeneous entities interact to achieve individual or collective goals. Examples include Multi-Agent Systems (MAS), Internet of Things (IoT) ecosystems, service-oriented architectures, cloud computing platforms, cyber-physical systems, microservice architectures, and future-generation heterogeneous networks integrating IoT, cloud services, and mobile infrastructures [1–6]. In such environments, autonomous agents continuously cooperate, compete, exchange services, and make decisions under uncertainty. Consequently, the ability to identify reliable interaction partners becomes a fundamental requirement for successful system operation [7–9]. This requirement is even more critical in Open Multi-Agent Systems (OMAS), where autonomous agents dynamically join and leave the environment and frequently interact with previously unknown entities. In these settings, weak or unreliable collaborative relationships may lead to task failure, degraded system performance, or even coalition breakdown, highlighting the importance of trust and reputation mechanisms for enabling robust cooperation [10].

Recent research has further refined the notion of openness in multi-agent environments. Open Multi-Agent Systems (MAS) are traditionally defined as environments in which agents may continuously enter and leave the system [11]. However, several researchers have argued that openness extends beyond agent population changes and also includes the ability of a system to accommodate new tasks and resources [12–14]. Under this broader perspective, openness can be viewed as the capability of a system to cope with the addition, removal, and evolution of its constituent entities, including agents, tasks, and resources [12–14]. Consistent with this view, Eck et al. [15] distinguish three complementary forms of openness in Open Agent Systems (OASYS): agent openness, where agents dynamically enter and leave the environment; task openness, where the set of tasks evolves over time; and type openness, where agents may acquire,

lose, or modify capabilities and behavioral characteristics during operation. These dimensions introduce additional uncertainty in decision-making, as agents must reason not only about the future behavior of others but also about the evolving availability of agents, tasks, and capabilities. As uncertainty increases, mechanisms supporting decentralized coordination, cooperation, and trust establishment become essential for maintaining effective system operation.

To address these challenges, researchers have often drawn inspiration from biological systems, where complex collective behavior emerges from the interactions among relatively simple entities. Examples include swarm intelligence, social insect colonies, and biologically inspired coordination mechanisms, which have been successfully applied to task allocation, resource management, distributed decision making, and trust management in large-scale distributed systems [16–25]. In particular, biologically inspired trust-management approaches have been proposed for dynamic and decentralized network environments, where local observations and neighborhood interactions are used to support trust assessment without requiring global system knowledge [26,27]. The appeal of these approaches lies in their scalability, adaptability, and ability to operate without centralized control. In trust management, this perspective is especially relevant, as it enables trust assessments to emerge from local interactions and distributed information processing. These characteristics align closely with the objectives of this dissertation and constitute an important source of inspiration for the trust-management framework developed in the following chapters.

Traditional security mechanisms, such as cryptography, authentication, and access control, provide protection against external threats and unauthorized access. However, they do not address scenarios in which legitimate entities behave unreliably, incompetently, or maliciously after authentication. For this reason, trust has been widely proposed as a complementary security mechanism in distributed and mobile-agent environments [28]. Rather than relying solely on identity verification, trust-management mechanisms aim to assess whether an authenticated entity is sufficiently reliable to participate in future interactions, thereby extending security considerations beyond authentication and authorization [29,30]. Consequently, the problem extends beyond secure communication to include the selection of appropriate interaction partners and the assessment of interaction reliability. In task-oriented multi-agent systems, where agents

frequently delegate tasks to others to improve utility, partner selection becomes a central coordination challenge.

Trust and reputation mechanisms have therefore been widely adopted as decision-support tools that help agents estimate the expected behavior of potential counterparts before engaging in interactions [9,31–33], thereby facilitating effective cooperation within agent societies and other collective environments [34]. This raises a fundamental question: how should trust be conceptualized and represented in computational environments?

1.1.2. Conceptual Foundations of Computational Trust

Trust remains a difficult concept to define precisely. The literature contains numerous definitions originating from disciplines such as sociology, psychology, economics, and computer science, reflecting diverse perspectives on expectations of behavior, reliability, risk, and the willingness to depend on others despite the possibility of adverse outcomes [35,36]. Several authors have emphasized that trust remains inherently ambiguous, as its interpretation varies across disciplines and application domains [37]. This lack of consensus has motivated efforts to develop more formal and domain-independent definitions of computational trust. For instance, Saputra [38] defines trust as a trustor's level of confidence in the ability of a trustee to produce the expected outcome of an interaction. Such definitions highlight the relational nature of trust and explicitly connect trust assessments to expectations of future behavior under uncertainty.

Nevertheless, trust is commonly characterized as a complex, abstract, subjective, context-dependent, multidimensional, and dynamic concept that evolves through experience and interaction [39–43]. A key property of trust is its subjectivity: different agents may form different evaluations of the same trustee even when exposed to similar evidence, as trust judgments are shaped by individual experiences, preferences, evaluation criteria, and contextual knowledge [2].

A widely adopted perspective models trust as a relationship between two entities: the **trustor**, who places trust, and the **trustee**, who is evaluated. From this viewpoint, trust is not an intrinsic property of an agent but a relational construct shaped by interaction context, prior experiences, expectations about future behavior, and the objectives of the trusting entity [39,44,45].

Trust assessments are often complemented by reputation, which incorporates social information obtained from other members of a community. Early reputation models such

as REGRET [46] distinguish between an individual dimension, derived from direct experience, and a social dimension, reflecting the influence of community structure and social relationships on trust formation. These models suggest that trustworthiness is influenced by multiple contextual and behavioral factors that contribute differently to an agent's overall evaluation.

While many computational trust models focus primarily on observed behavior and interaction outcomes, other researchers argue that trust assessments should also incorporate broader social and normative dimensions. Carter and Ghorbani [47] propose a value-centric trust model in which trust emerges from the interaction between reputation, self-esteem, familiarity, and the fulfillment of social values such as honesty, responsibility, helpfulness, capability, and cost-efficiency. From this perspective, trust is not only an estimate of expected future behavior but also a multidimensional social construct shaped by the extent to which agents conform the norms and expectations of the societies in which they operate.

1.1.3. Trust Representation and Trust Formation

Given the conceptual complexity of trust, a central challenge in computational trust research is determining how trust can be represented and quantified in a form suitable for autonomous decision making. Trust is commonly modeled as a numerical estimate of the confidence that a trustor places in a trustee. To capture the multifaceted nature of cooperative behavior, many computational trust models adopt multidimensional representations in which trustworthiness is evaluated across multiple attributes rather than being reduced to a single scalar value [48,49]. For example, an agent may be considered highly trustworthy with respect to task quality while being less reliable regarding execution time or cost [50]. Similar approaches have been applied in service-oriented environments, where trust is derived from multiple Quality of Service (QoS) attributes, such as reliability, availability, response time, and cost, allowing trust evaluations to reflect the preferences and priorities of service consumers [51].

Although multidimensional representations are widely adopted, they often assume that trust dimensions are independent and can be aggregated into a trust vector or a single trust score. Recently, Messina et al. [52] challenged this assumption by arguing that trust dimensions may exhibit dependency relationships. For example, an agent may infer honesty from reliability or competence from expertise, even when direct evidence

regarding these dimensions is unavailable. To capture such interdependencies, they proposed representing trust as a graph of interrelated trust aspects rather than as a simple vector of attributes, enabling agents to infer unknown dimensions from known ones through explicitly modeled dependency relationships.

Trust values are typically updated over time based on interaction outcomes, increasing after successful interactions and decreasing following failures or undesirable behavior. Most computational trust models rely on multiple sources of evidence, including **direct trust**, derived from the trustor's own interaction history, and **indirect trust** or **reputation**, obtained from recommendations and opinions provided by other agents [53]. By combining these sources, trustors estimate the trustworthiness of potential partners and select those expected to maximize utility or minimize risk [31,32,54].

The manner in which trust evidence is combined varies considerably across computational trust models. For example, Beltzer et al. [55] proposed new aggregation operators for a bi-lattice-based trust model, extending a traditionally local trust representation to support global reputation computation. Their approach integrates both trust and distrust information and propagates evidence through trust-network relationships, enabling reputation estimation even when only partial knowledge of the network is available. Similar principles have been applied in Social Internet of Things (SIoT) environments, where trust assessments often combine direct experiences, indirect recommendations, and social relationships among devices. Sagar et al. [56], for instance, proposed a trust quantification model that integrates direct trust, indirect trust, and social similarities such as friendship, community-of-interest, and co-work relationships into a unified trust score. Their results suggest that combining behavioral and social evidence can improve the identification of trustworthy and untrustworthy entities in highly decentralized environments.

Beyond evidence aggregation and reputation-based approaches, alternative trust-formation mechanisms rely on continuous behavioral analysis. Arazzi et al. [57] proposed a fully distributed trust model for IoT environments that evaluates device trustworthiness through behavioral fingerprinting. In their approach, devices are characterized by behavioral profiles describing their expected communication and interaction patterns. Trust assessments are generated by comparing observed behavior against these profiles, while distributed consensus and blockchain mechanisms are employed to maintain a

decentralized view of device reliability. This work illustrates how trust can emerge from behavioral monitoring and anomaly detection rather than from explicit reputation exchanges alone.

The diversity of these approaches demonstrates that computational trust can be established through multiple forms of evidence, including direct experience, reputation, social relationships, and behavioral observations. Similar principles have also been explored in other decentralized environments, including UAV networks, where trust and reputation systems are increasingly recognized as important mechanisms for enhancing network security communication reliability, and the trustworthiness of autonomous entities operating under dynamic conditions [58,59]. Collectively, these studies highlight the central role of trust representation, evidence integration, and trust formation mechanisms in supporting reliable decision making in open and distributed systems.

1.1.4. Computational Trust Models and Research Challenges

Numerous computational trust and reputation models have been proposed for open Multi-Agent Systems [60]. Early approaches, such as SPORAS, HISTOS, REGRET, and FIRE, focused on aggregating direct experiences and reputation information to estimate the trustworthiness of potential interaction partners. Later probabilistic and Bayesian models introduced more sophisticated mechanisms for handling uncertainty and incomplete information. Beyond partner selection, trust models have also been applied to resource allocation, task scheduling, and access-control mechanisms in distributed and IoT environments, where trust estimates are used to improve reliability, security, and execution success rates [61,62]. Surveys of the field have shown that existing approaches differ substantially in their underlying assumptions, trust representations, uncertainty-handling mechanisms, reputation-management strategies, and support for decision making in open environments [6,63–65]. This growing diversity has motivated the development of conceptual frameworks that facilitate the systematic analysis, classification, and comparison of heterogeneous trust models across different applications domains [66–68].

Despite their methodological differences, most computational trust models share a common underlying assumption: trustors evaluate available trustees and select those considered most trustworthy according to the information currently available. This view is consistent with the widely adopted distinction between trust evaluation, during which agents estimate the trustworthiness of potential partners, and trust-aware decision making,

during which these estimates are used to guide future actions and partner-selection decisions [69–71]. Consequently, computational trust research has traditionally focused on supporting trustor-side decision making, while comparatively less attention has been devoted to mechanisms that help trustees understand, establish, maintain, or improve their trustworthiness over time [72,73]. Similar observations were made by Sen [74], who argued that trust establishment, trust maintenance, and the strategic use of trust relationships remain underexplored compared with trustworthiness assessment. As a result, the trustor-centric perspective has emerged as the dominant paradigm in computational trust research.

At the same time, the increasing openness, scale, and decentralization of modern computational environments continue to expose important limitations of existing trust approaches. Agents may join or leave the system at any time, exhibit heterogeneous and evolving behaviors, and provide information that is incomplete, unavailable, or misleading [7,9,11,75–77]. These characteristics make the prediction of future behavior inherently difficult and contribute to significant variations in trust-model performance across different environments. Consequently, the literature consistently emphasizes that no single trust model performs equally well under all conditions, since different approaches rely on different assumptions regarding information availability, communication patterns, social structures, and agent behavior [63].

These observations suggest that future research should not focus solely on improving the accuracy of trust assessment, but also on understanding how trust relationships are established, maintained, adapted, and strategically utilized under changing environmental conditions. The research presented in this dissertation is situated at the intersection of computational trust, Multi-Agent Systems, and machine learning, and investigates alternative approaches to trust formation and trust-based decision making in open and dynamic environments. Particular emphasis is placed on resource-constrained settings, where conventional trust mechanisms may become inefficient or ineffective. More specifically, this dissertation challenges the traditional trustor-centric perspective that dominates the computational trust literature and explores how trustees themselves can actively contribute to the development and maintenance of trust relationships. By investigating trustee-centric trust models, adaptive trust mechanisms, and reinforcement-learning approaches for trust-model selection, this work seeks to contribute to the

development of more robust, flexible, and scalable trust frameworks capable of operating effectively under uncertainty, limited information availability, and continuously evolving environmental conditions.

1.2. Limitations of Existing Trust Management Approaches

Despite significant advances in computational trust research and the large number of trust and reputation models proposed in the literature, existing approaches continue to have important limitations when deployed in open and dynamic environments. Many of these limitations stem from a common assumption underlying conventional trust models: trust is calculated by the trustor, who evaluates available trustees and selects the one considered most trustworthy for a particular interaction [74].

1.2.1. The Cold Start Problem

One of the most persistent challenges in computational trust research is the cold-start problem. Trust and reputation models typically perform poorly when evaluating newcomer service providers because service requesters have limited means of assessing agents with whom they have no prior interactions and for whom little or no reputation information is available. Newly arrived agents often lack interaction history and community feedback, preventing conventional trust models from producing reliable trust estimates. Under such conditions, trustors may be unable to accumulate sufficient evidence before interaction opportunities disappear, leading either to suboptimal partner-selection decisions or to overly conservative behavior that restricts cooperation [78].

To mitigate these limitations, a variety of approaches have been proposed. Stereotyping, originally introduced by Meyerson et al. [79], is based on the assumption that agents sharing similar observable characteristics are likely to exhibit similar behavior. This approach facilitates the assessment of newcomers and other agents with limited interaction histories and is closely related to more recent techniques that exploit profile similarity and auxiliary contextual information to bootstrap initial trust estimates [80]. More recently, machine-learning methods have been employed to predict the likely behavior or reliability of newcomer agents using contextual information and historical patterns derived from similar entities [81]. In addition, organizational roles and social relationships within virtual communities can serve as alternative sources of trust information [82], particularly when direct or indirect trust evidence is unavailable.

Another common strategy is to assign an initial trust value until sufficient direct experience has been accumulated. Such values may reflect the average level of performance observed within the environment or may be inferred from interactions involving related agents [83]. Although these approaches partially alleviate the cold-start problem, the challenge remains significant in highly dynamic environments characterized by scarce information, short-lived interactions, and substantial uncertainty. Consequently, addressing the cold-start problem remains an important objective of computational trust research and constitutes one of the key motivations underlying the work presented in this dissertation.

1.2.2. Highly dynamic environments and Agent Mobility

Another major limitation of existing trust-management approaches is their difficulty in coping with highly dynamic environments. In open Multi-Agent Systems, agents continuously enter and leave the system, while their capabilities, roles, and behavioral characteristics may evolve over time. Conventional trust models often rely on repeated interactions to construct reliable trust estimates. Consequently, trust information may become outdated before sufficient evidence has been accumulated, reducing the effectiveness of trust-based decision making and limiting the predictive value of previously observed behavior [7,11,75].

Beyond the challenge of maintaining up-to-date trust assessments, dynamic environments also introduce issues related to agent mobility. A longstanding concern in trust and reputation research is the limited interoperability of trust-management systems across different applications and administrative domains. When entities move between environments, their trust and reputation information often remains confined to the original application, forcing them to rebuild their reputation from scratch in each new setting. Enabling the portability of trust information across heterogeneous systems therefore remains an important open challenge.

To address this issue, Jelenc [82] proposed a general mechanism for exchanging trust and reputation information between independent systems. The proposed approach defines a communication protocol that enables trust-management systems to request ratings, provide responses, and communicate error conditions. To facilitate interoperability, the framework also includes a grammar for constructing trust-related queries and procedures for parsing exchanged messages.

A related approach was proposed by Jabeen et al. [84], who developed a hybrid trust-management architecture that combines centralized and distributed trust-evaluation mechanisms to support agent mobility. In their framework, the network is partitioned into multiple sub-networks, each managed by a resource-rich fog node responsible for maintaining and sharing reputation information. Reputation scores are aggregated and stored in the cloud, allowing service providers that migrate between sub-networks to preserve their previously established reputation. As a result, trust information can follow agents as they move across the system, reducing the need to rebuild trust from the beginning after each migration.

Although these approaches improve the portability and availability of trust information, maintaining accurate trust assessments in highly dynamic environments remains a significant challenge. Frequent behavioral changes, limited interaction opportunities, and the continuous movement of agents across environments can still reduce the reliability of trust estimates and complicate trust-based decision making under uncertainty.

1.2.3. Behavioral volatility

A further challenge for trust management in open Multi-Agent Systems is behavioral volatility, namely the tendency of agents to alter their behavior over time. Such changes may arise from malicious intent, resource limitations, environmental conditions, device malfunctions, network congestion, seasonal effects, or evolving social relationships among agents [83,84]. Some behavioral variations occur unpredictably, whereas others follow recurring or cyclical patterns. As a result, trust assessments derived from historical observations may quickly become outdated and fail to accurately reflect an agent's current trustworthiness.

Traditional trust and reputation models typically address this problem through temporal weighting mechanisms such as sliding windows and forgetting factors [83]. A sliding window preserves only the most recent interactions under the assumption that they best represent current behavior, while older observations are discarded. In contrast, forgetting-factor approaches retain all historical interactions but gradually reduce the influence of older evidence over time. Both mechanisms aim to improve responsiveness to behavioral change by emphasizing recent observations.

An example of the forgetting-factor approach is the trust model for pervasive computing environments proposed by kolomvatsos et al. [85]. The model combines direct experience and reputation information using fuzzy-logic techniques and incorporates referral age and referrer reputation into the calculation of social trust. More recent referrals exert greater influence on trust assessments, while older information gradually loses significance and is eventually discarded. Although this approach improves adaptability to changing conditions, it relies on a central authority for storing and managing referrals, introducing a potential single point of failure.

Despite their widespread use, traditional temporal weighting mechanisms often struggle to cope with agents whose behavior changes abruptly or at irregular intervals. Similar concerns have been highlighted in earlier studies showing that many trust models react slowly to sudden behavioral shifts and may be exploited by agents exhibiting oscillatory or strategically changing behavior [86]. To address this limitation, Player and Griffiths [83] propose RaPTaR, a framework that operates alongside existing trust models and dynamically identifies behavioral changes within agent groups. Using the Kolmogorov-Smirnov statistical test, RaPTaR detects shifts in behavioral patterns and supplies trust algorithms with experience data that better reflect current behavior. The framework also records recurring behavioral transitions, enabling the recognition of repetitive patterns that can support future predictions. Experimental results indicate that although RaPTaR improves responsiveness to behavioral change, its detection accuracy remains limited.

A related line of research focuses on trust prediction. Reputation-based trust models often suffer from two interconnected problems: reputation information may no longer reflect an entity's current behavior, and excessive reliance on reputation increases vulnerability to collusion and other forms of manipulation. Trust-prediction techniques attempt to overcome these limitations by estimating current trustworthiness from historical interactions and behavioral trends. Traditional approaches include arithmetic averaging, mean-shift methods, and exponential smoothing. However, exponential smoothing alone often performs poorly when abrupt behavioral transitions occur, such as when a previously trustworthy entity becomes compromised. To improve prediction accuracy under such conditions, Wang et al. [87] proposed a dynamic trust model that combines direct and indirect trust computation with trust prediction. Their approach integrates exponential

smoothing with Markov-chain modeling to capture both long-term trends and sudden fluctuations in behavior.

Behavioral volatility is particularly evident in emerging networked environments such as Fifth-Generation (5G) communication networks. In these systems, forwarding entities may alternate between legitimate and malicious behavior, for example by selectively dropping packets while attempting to avoid detection [88]. Such environments are especially challenging because both legitimate and malicious entities may employ reinforcement-learning techniques to adapt their behavior over time. Ahmad et al. [88] addressed this problem through a hybrid trust model that combines Q-learning-based trust assessment with network-wide trust information to support secure routing decisions. Nevertheless, the effectiveness of the proposed approach has not yet been comprehensively validated experimentally.

Overall, behavioral volatility remains one of the most challenging problems in computational trust research. Although mechanisms such as temporal weighting, behavioral-change detection, and trust prediction can improve adaptability, accurately assessing trustworthiness in environments characterized by rapidly changing and strategically adaptive behavior remains an open research challenge.

1.2.4. Communication and computational overhead

Conventional trust-management approaches often introduce significant communication and computational overhead [89]. To assess the trustworthiness of potential interaction partners, agents frequently need to collect information from multiple sources, exchange recommendations, maintain historical records of interactions, and continuously update trust estimates. In large-scale distributed environments, these activities can generate considerable messaging costs and increase the processing requirements imposed on participating agents [90].

The problem becomes even more pronounced in resource-constrained environments, such as IoT ecosystems, mobile computing platforms, and wireless sensor networks, where processing power, memory capacity, energy resources, and communication bandwidth are inherently limited [91]. In such settings, trust-management mechanisms that rely on extensive information exchange or computationally intensive trust calculations may become inefficient, reduce system scalability, and negatively affect overall performance.

Consequently, minimizing communication and computational overhead remains an important challenge in the design of practical trust-management solutions for open and dynamic environments. Effective trust models must therefore balance the need for accurate trust assessment with the communication and computational costs associated with the collection, dissemination, and evaluation of trust-related information.

1.2.5. Disinformation attacks

The use of indirect trust and reputation information introduces additional vulnerabilities into trust-management systems. Reputation-based mechanisms rely on recommendations provided by other agents, creating opportunities for disinformation attacks. Malicious entities may intentionally provide misleading recommendations to manipulate trust assessments, either by unfairly reducing the reputation of trustworthy agents (bad-mouthing attacks) or by artificially increasing the reputation of malicious agents (ballot-stuffing attacks). Defending against such attacks remains a persistent challenge in computational trust research.

A particularly difficult form of disinformation is collusion, in which groups of malicious agents cooperate by exchanging mutually supportive recommendations to artificially enhance their own reputation or damage that of competitors. Qureshi et al. [92] demonstrated that even sophisticated reputation-based trust models such as FIRE remain vulnerable to coordinated witness manipulation and consequently proposed the FIRE+ framework to detect and mitigate collusive recommendation behavior.

To improve the robustness of reputation systems, several researchers have focused on assessing the reliability of the information sources themselves. For example, the Comprehensive Reputation Model (CRM) proposed by Khosravifar et al. [93] incorporates an ex-post evaluation mechanism in which the actual outcome of an interaction is compared with previously received recommendations. This enables the credibility of recommendation providers to be continuously updated and reduces the influence of inaccurate or deceptive information on future trust assessments.

Related work by Khalid et al. [94] proposed a blockchain-based trust management framework for Multi-Agent Systems in which trust evaluators are assessed using trust-distortion, consistency, and reliability measures. By identifying dishonest or unreliable evaluators, the framework seeks to limit the impact of manipulated recommendations while promoting cooperative behavior through blockchain-supported trust-management

mechanisms. These studies highlight the importance of evaluating not only the target of a trust assessment but also the agents that provide trust-related information.

Beyond the credibility of recommendation providers, the quality of the reputation information itself also affects the reliability of trust assessments. Reputation scores often aggregate multiple aspects of an interaction into a single evaluation, making it difficult to distinguish between different sources of dissatisfaction. Furthermore, reputation systems remain vulnerable to intentionally manipulated reviews. To address these limitations, Oh et al. [95] proposed a reputation framework that separates seller performance from item quality and introduces an integrity-weighting mechanism that estimates the credibility of individual reviews before incorporating them into reputation calculations. Their results demonstrate that accurate reputation estimation depends not only on robust aggregation mechanisms but also on the ability to assess the informational quality of the evidence being aggregated.

Importantly, distortions in reputation information do not necessarily originate from malicious behavior. Indirect trust assessments may also be affected by implicit biases and collective stereotypes embedded in user feedback, leading to unfair or inaccurate reputation estimates even when recommendations are provided honestly. To address this issue, Bangui et al. [96] proposed a deep-learning-based reputation model that employs synthetic data generation to mitigate bias in indirect trust information. Their work demonstrates that reliable trust assessment depends not only on detecting malicious recommendations but also on identifying and mitigating systematic sources of bias within reputation systems.

Consequently, the design of mechanisms capable of filtering deceptive recommendations, identifying collusive behavior, mitigating recommendation bias, and improving the reliability of indirect trust information remains an important research challenge in open agent societies [97,98]. In response, a variety of incentive-based and trust-management mechanisms have been proposed to encourage honest reporting and discourage malicious participation, as discussed in the following subsection.

1.2.5.1. Mechanisms for Promoting Honest Behavior

Several studies [99–107] have proposed mechanisms that encourage honest behavior and discourage malicious actions in decentralized environments. Although these approaches differ in their implementation, most rely on a combination of incentives,

reputation management, punishment mechanisms, recommendation filtering, and adaptive trust assessment. These mechanisms provide useful insights for extending the CA framework to operate in environments where dishonest behavior is prevalent.

A fundamental challenge in large-scale and dynamic IoT environments is motivating nodes to provide accurate trust information during authentication procedures. To address this issue, Fragos et al. [100] propose a contract-theoretic incentive mechanism that encourages nodes to participate honestly in trust establishment. Each node maintains and advertises its trust score, while neighboring nodes assist in the authentication process. The mechanism associates rewards with both the node's trustworthiness and its contribution to authentication activities. As a result, nodes maximize their expected benefit by accurately reporting trust information and actively participating in the protocol. Unlike many blockchain-based approaches, the mechanism relies on localized trust formation and incentive alignment rather than global consensus.

Pan et al. [101] introduce TrustCoin, a trust management framework that promotes honest behavior through a blockchain-supported credit and reputation mechanism. Each participant is assigned a number of credit coins that simultaneously represent its reputation and determine its eligibility to participate in content dissemination activities. Before publishing content, a user must possess a sufficient number of credit coins, thereby preventing low-reputation entities from freely distributing information. The credibility of published content is subsequently evaluated using a Deep Reinforcement Learning model, and the participants' credit balances are updated accordingly. Publishers who provide highly credible content receive additional credit coins, strengthening both their reputation and their future participation opportunities. Conversely, publishers that disseminate low-credibility content are penalized through credit reduction. The mechanism also incentivizes subscribers to actively monitor content quality: users who correctly identify and report malicious or misleading content are awarded, while inaccurate assessments may result in penalties. By continuously adjusting credit balances according to observed behavior and content quality, TrustCoin creates a self-reinforcing ecosystem in which both content producers and consumers are encouraged to act honestly and contribute to the overall trustworthiness of the network.

In many IoT networks, nodes may behave selfishly by refusing to forward packets in order to conserve resources such as energy, bandwidth, and memory. To address this

challenge Muhammad et al. [102] propose the Honesty-Based Social Trust (HBST) framework, which promotes cooperation through community formation, behavioral monitoring, and controlled rehabilitation of selfish nodes. The mechanism operates in two phases. First, nodes are evaluated according to their honesty level, which is derived from factors such as interaction frequency, credibility, and participation within the community. Nodes whose reputation exceeds a predefined threshold are admitted to a "credible community" consisting exclusively of trustworthy participants. In the second phase, the members of this community elect two leaders: a Selected Community Head (SCH) and a Selected Monitoring Head (SMH). The SCH is responsible for managing the community and enforcing penalties against selfish nodes, while the SMH continuously monitors node behavior and reports selfish activities. Nodes identified as selfish are isolated in a separate domain and are prevented from interacting with the credible community. However, unlike traditional approaches that permanently exclude misbehaving nodes, HBST allows isolated nodes to regain membership if they improve their honesty level and satisfy the required reputation threshold. Persistent offenders face progressively stronger penalties, including warning notifications distributed to neighboring communities and eventual expulsion from the network. By combining continuous monitoring, social accountability, temporary isolation, and conditional reintegration, the framework creates strong incentives for nodes to behave cooperatively while still allowing behavioral recovery.

In Internet of Medical Things (IoMT) environments, establishing trustworthy communication among a large number of interconnected medical devices is critical for reliable healthcare monitoring and decision-making. To address this challenge, Ali et al. [103] propose BFT-IoMT, a blockchain-enabled trust management framework that combines clustering, fuzzy logic, and distributed trust evaluation to detect and isolate malicious devices. The mechanism begins by organizing IoMT devices into clusters and assigning Cluster Heads (CHs), which coordinate trust evaluation within their respective groups. Trust-related behavioral parameters are continuously collected from participating devices and processed by a trust calculator that employs fuzzy logic to handle uncertainty and imprecision in the observed data. Based on this analysis, each device is assigned a trust score that reflects its overall trustworthiness. The calculated scores are recorded on the blockchain, providing a tamper-resistant history of trust evaluations that can be shared across the network. Devices whose trust scores fall below a predefined threshold are

identified as potentially malicious, particularly in the case of Sybil attacks, and are subsequently isolated from further communication. The final trust decision is then disseminated throughout the network, ensuring that other devices can avoid interacting with the identified malicious nodes. By continuously monitoring node behavior, maintaining transparent trust records, and excluding devices that exhibit suspicious activity, the framework encourages participants to maintain trustworthy behavior in order to remain active members of the network.

Kouicem et al. [104] propose a decentralized trust management protocol that enables IoT devices to assess the trustworthiness of service providers without relying on pre-trusted entities. The mechanism combines blockchain-based trust storage with a fine-grained recommendation model. Rather than assigning a single trust value to a service provider, the protocol evaluates trust with respect to specific service requirements, allowing IoT devices to obtain recommendations that are tailored to their particular needs. To support trustworthy trust assessments, recommendations and trust evidence are permanently recorded on the blockchain, providing an immutable history that can be accessed by fog nodes during trust computation. The protocol further incorporates an online countermeasure against cooperative recommendation attacks. Whenever trust recommendations are aggregated, the fog node analyzes the distribution of recommendation values stored on the blockchain. Large discrepancies between recommendations indicate the possibility of coordinated attacks, such as self-promotion, bad-mouthing, or ballot stuffing. In such cases, suspicious recommendations are discarded before the final trust value is calculated. By combining fine-grained trust assessment, transparent recommendation histories, and real-time detection of manipulated recommendations, the protocol reduces the influence of dishonest participants and improves the reliability of trust decisions in decentralized IoE environments.

Liu et al. [107] propose BtRaI (Blockchain and trusted Reputation assessment-based Incentive mechanism), a trust management framework designed to promote honest participation in Healthcare IoT environments. The mechanism begins with a multidimensional reputation assessment process, in which each participant is evaluated using multiple behavioral and service-related factors. The resulting reputation values are securely recorded on a consortium blockchain, providing a transparent and tamper-resistant history of participant behavior. Based on these reputation scores, BtRaI employs

a token-based reward and punishment mechanism. Participants that consistently provide reliable services receive rewards, while nodes exhibiting dishonest or malicious behavior incur penalties that reduce their reputation and associated benefits. Beyond economic incentives, the framework further links reputation to the consensus process through an enhanced PBFT algorithm. Nodes with higher reputation scores are granted a greater role in consensus operations, whereas low-reputation nodes have reduced influence on network decisions. Consequently, honest behavior yields both financial rewards and increased participation privileges, while dishonest behavior results in economic losses and diminished authority within the system. By simultaneously combining reputation assessment, incentive allocation, and reputation-aware consensus participation, the framework creates multiple incentives for maintaining trustworthy behavior.

Ouechtati et al. [105] propose a fuzzy logic-based mechanism for filtering dishonest recommendations in the Social Internet of Things (SIoT). Rather than accepting all recommendations equally, the model evaluates their reliability using multiple contextual and social factors, including recommendation values, submission time, geographic location, and the social relationships between objects. The first stage focuses on detecting Sybil attacks. Recommendations that exhibit high similarity in content, are submitted within a short time interval, and originate from nearby locations are considered likely to have been generated by the same malicious entity. Such recommendations are identified through fuzzy classification and removed from the subsequent trust calculations. The remaining recommendations are then evaluated using two complementary metrics: Internal Similarity (IS) and Degree of Social Relationship (DSR). IS measures the extent to which a recommendation agrees with the opinions of trusted members within the same community, while DSR quantifies the strength of the social connection between the recommender and the recommended object based on their previous interactions and relationship type (e.g., ownership, co-location, co-work, or social relationship). Recommendations that exhibit both low agreement with trusted community members and weak social connections are considered unreliable and may indicate good-mouthing or bad-mouthing attacks. By reducing the influence of recommendations that appear inconsistent, socially unsupported, or potentially coordinated, the mechanism limits the ability of malicious agents to manipulate trust values and therefore promotes more reliable and honest trust assessments.

In Vehicular Edge Computing (VEC) environments, Autonomous Vehicles (AVs) can trade data with edge servers in exchange for rewards, creating an economic incentive for data sharing. However, the success of such transactions depends on the ability of edge servers to identify trustworthy data providers. To address this challenge, Mianji et al. [106] propose the Dynamic Selection of Trusted Sellers using Deep Deterministic Policy Gradient (DSTSDPG), a reputation management mechanism that combines trust assessment with deep reinforcement learning. Instead of relying on a fixed trust threshold, the mechanism dynamically adjusts the minimum credibility level required for an AV to participate in data trading. Each vehicle is assigned a credibility score based on three factors: historical reputation, familiarity, and freshness of previous interactions. Historical reputation reflects past behavior, familiarity captures the extent of previous interactions between an edge server and a vehicle, and freshness assigns greater importance to recent observations. Using these factors, the edge server continuously evaluates candidate sellers and selects the most trustworthy vehicles for data transactions. Since only highly credible vehicles are eligible to participate in profitable trading opportunities, AVs are incentivized to maintain a positive reputation by consistently providing reliable data. Furthermore, credibility information is shared across clusters through the cloud infrastructure, allowing trustworthy behavior to be recognized even when vehicles move between different network regions.

Existing trustworthiness models can only detect a subset of known attacks, but none can defend against all types [108]. This is because attack patterns are highly diverse, with malicious nodes strategically exploiting weaknesses in trust algorithms to evade detection. To address this challenge, Marche and Nitti examine in [108] all trust-related attacks documented in the literature that can impact IoT systems. They propose a decentralized trust management model that leverages a machine learning algorithm and introduces three novel parameters: goodness, usefulness, and perseverance scores. These scores enable the model to continuously learn, adapt, and effectively identify and counteract a wide range of malicious attacks.

Overall, the literature suggests that honest behavior can be promoted through several complementary mechanisms, including reward-and-punishment schemes, reputation-based incentives, temporary exclusion and rehabilitation policies, recommendation filtering, transparency through blockchain technologies, and adaptive

machine learning techniques. Since each mechanism addresses different aspects of dishonest behavior, combining multiple approaches appears to be a promising direction for developing robust trust management systems. These observations motivate the discussion in Chapter 6, where several of these mechanisms are considered for integration into the proposed CA framework.

1.2.6. Privacy concerns

The exchange of trust and reputation information introduces important privacy challenges. Trust assessments often reveal information about an agent's preferences, experiences, interaction histories, and relationships with other agents potentially exposing sensitive aspects of its behavior and decision-making processes. As a result, agents may be reluctant to share recommendations, reducing both the availability and reliability of reputation information [109,110].

Beyond the disclosure of interaction information, a related challenge arises from the strong coupling between reputation and identity in many trust-management systems. Reputation scores are frequently associated with persistent identities, making it possible to track behavior over time and potentially exposing participants to profiling or retaliation. Conversely, allowing agents to change identities may facilitate whitewashing and Sybil attacks, enabling malicious entities to evade the consequences of poor reputation. To address this trade-off between accountability and privacy, Bouchiha et al. [111] proposed DARS, a decentralized anonymous reputation system that combines blockchain technology with cryptographic techniques to support anonymous participation while preserving reputation continuity across multiple pseudonyms. Their work demonstrates that privacy preservation and reputation management need not be mutually exclusive and highlights the growing importance of privacy-aware trust mechanisms in decentralized environments.

1.2.7. Summary of Limitations and Research Motivation

These limitations discussed throughout this chapter illustrate the challenges faced by existing trust-management approaches when operating in environments characterized by uncertainty, mobility, limited resources, malicious behavior, and rapidly changing conditions. Despite significant advances in computational trust research, there remains a need for alternative solutions that reduce communication overhead, improve resilience to

deceptive behavior, adapt more effectively to behavioral change, and support trust formation under limited information availability.

These challenges provide a strong motivation for the research presented in this dissertation. In particular, they motivate the investigation of trustee-centric trust management mechanisms as a possible alternative to conventional trustor-centric approaches, with the goal of developing more adaptive, scalable, and robust trust-management solutions for open and dynamic environments.

1.3. Trustee-Centric Perspective on Trust Management

Most computational trust models adopt a trustor-centric perspective, in which the responsibility for trust formation lies entirely with the trustor. In these approaches, trustors collect information about potential trustees, estimate their trustworthiness, and select those expected to provide the highest utility. As discussed in the previous sections, this paradigm has dominated computational trust research for more than two decades and forms the basis of most existing trust and reputation mechanisms [74].

Despite its widespread adoption, the trustor-centric paradigm implicitly assumes that trustors possess sufficient information and resources to identify suitable interaction partners. In practice, however, this assumption is often unrealistic. In open and dynamic environments, trustors frequently operate under conditions of incomplete information, limited computational resources, uncertain behavioral patterns, and rapidly changing agent populations. Consequently, trust decisions may need to be made using sparse, outdated, or unreliable information. At the same time, comparatively little attention has been devoted to mechanisms that assist trustees in actively establishing, maintaining, or improving their trustworthiness.

Several researchers have therefore argued that trust should also be examined from the trustee's perspective. Castelfranchi et al. [73,112] view being trusted as a form of relational[1] capital that increases an agent's opportunities for cooperation and interaction.

[1] Here, relational capital refers to the value derived from being trusted within a network of interacting agents, representing the set of potential interactions, cooperation opportunities, and delegation choices that become available to an agent as a result of its perceived trustworthiness.

Similarly, Aref and Tran [72] distinguish between trust evaluation and trust establishment, arguing that most existing approaches focus on helping trustors assess potential partners rather than guiding trustees toward behaviors that increase the confidence of future interaction partners. More recently, trust-establishment mechanisms have been proposed for Internet of Agents environments, where trustees adapt their behavior based on implicit feedback from trustors in order to increase future trust levels and improve their chances of being selected as interaction partners [113].

These observations motivate the investigation of an alternative perspective on trust management in which trustees actively participate in the trust-formation process. Rather than requiring trustors to identify the most trustworthy service provider, the approach proposed in this dissertation, introduced in Chapter 2, shifts part of the decision-making responsibility to the trustees themselves. Under this view, trustees evaluate their own ability to successfully perform a requested task and decide whether they should undertake it. Consequently, the trustee becomes an active participant in the trust-management process rather than a passive entity being evaluated by others.

This perspective is consistent with the notion of trust establishment introduced by Tran et al [114], who argued that trust management should not only help trustors identify reliable partners but should also provide mechanisms through which trustees can actively develop and maintain trust relationships. While the approach of Tran et al. focuses primarily on behavioral adaptation aimed at increasing future trust levels, the *Create Assemblies* (CA) model approaches the problem from a different perspective by allowing trustees to assess their own capability and willingness to perform tasks before accepting interaction requests.

The idea of transferring part of the decision-making responsibility to service providers has also appeared in other decentralized multi-agent domains. Studies in swarm robotics and multi-agent task allocation have shown that agents can autonomously decide which tasks to undertake based solely on local information, without centralized control or global system knowledge [18,19]. Similarly, research on collaborative team formation in open Multi-Agent Systems has proposed self-evaluation mechanisms through which agents assess their own capabilities, resource constraints, and degree of self-interest before deciding whether to participate in collaborative activities [115]. Such approaches emphasize decentralized decision making and autonomous participation, reinforcing the

view that agents can actively determine their suitability for a task rather than relying exclusively on external evaluation.

Related ideas have also emerged within the trust-management literature. For example, ConTrust, proposed by Latif [116], is a context-aware trust model for the Social Internet of Things (SIoT) in which service providers evaluate their capability and willingness to satisfy a requested task before responding to a service requester. Although the final provider selection remains the responsibility of the requester, the model incorporates a form of provider-side self-assessment that shares important similarities with the trustee-centric philosophy adopted in the CA model. Trustworthiness in ConTrust is evaluated using a combination of provider capability, commitment, and job-satisfaction feedback, while task requirements and contextual information are explicitly considered during service provision.

A related direction was explored by Alam et al. [99], who proposed a blockchain-based two-way reputation framework for the SIoT that evaluates both service providers and service requesters. The framework combines local trust, global trust, reputation, social trust, and Quality of Service (QoS) attributes, while introducing incentive and penalty mechanisms to discourage dishonest behavior. Unlike conventional trust models, service providers are allowed to accept or reject service requests, thereby assuming a more active role in the interaction process. The model also incorporates mechanisms for detecting dishonest requesters through comparisons between local and global trust assessments and employs reputation-based classifications and penalties to encourage honest participation. Experimental results demonstrated resilience against several trust-related attacks, including on-off, discriminatory, opportunistic-service, selective-behavior, bad-mouthing, and ballot-stuffing attacks, although the framework does not address whitewashing attacks.

Taken together, these studies suggest that trust management need not rely exclusively on trustor-side evaluation and that trustees can play a more active role in trust formation and service provision. The trustee-centric paradigm therefore offers several potential advantages, which are explored in the following chapters. Nevertheless, introducing a trustee-centric perspective also raises important research questions. Can trustees accurately assess their own capabilities? Under what environmental conditions does trustee-centric trust outperform conventional trustor-centric approaches? How can

trustee self-assessment adapt when agent behavior changes over time? And can trustee-centric and trustor-centric paradigms be combined within a unified adaptive framework?

The remainder of this dissertation seeks to answer these questions through the development, evaluation, and extension of the *Create Assemblies* trust model. In particular, the following chapters investigate the strengths and limitations of trustee-centric trust management, explore mechanisms for adaptive self-assessment, and examine how reinforcement learning can be used to dynamically select between trustee-centric and conventional trust-management approaches according to environmental conditions.

1.4. Adaptive Trust Management

A fundamental assumption underlying most trust-management approaches is that a single trust model is selected and remains fixed throughout system operation. In practice, however, no trust mechanism can be expected to perform optimally under all environmental conditions. The effectiveness of a trust model depends on numerous factors, including agent mobility, population dynamics, behavioral volatility, interaction frequency, information availability, and communication constraints.

Taken together, these factors suggest that trust management should not be viewed as a search for a universally superior trust model. Instead, different trust mechanisms should be regarded as complementary tools whose effectiveness depends on the circumstances under which they are applied. This perspective is consistent with the broader principle that decision-making strategies should adapt to environmental conditions rather than remain fixed regardless of context.

The need for adaptability becomes particularly pronounced in open Multi-Agent Systems. Agents typically operate with only partial knowledge of their environment and cannot observe all factors influencing system dynamics. Population changes, behavioral shifts, resource availability, and communication patterns may evolve continuously and unpredictably. Consequently, determining which trust mechanism is most appropriate becomes a dynamic decision problem rather than a one-time design choice.

The experimental evaluation presented in this dissertation supports this view. As will be shown in subsequent chapters, different trust mechanisms exhibit distinct strengths and weaknesses depending on the characteristics of the environment. For example, the original CA model demonstrates superior performance in situations involving significant

changes in consumer populations, whereas conventional trustor-centric approaches such as FIRE may perform better when provider populations change frequently. Similarly, mechanisms designed to rapidly detect behavioral changes may outperform more stable approaches in volatile environments while introducing unnecessary complexity in environments where agent behavior remains relatively constant.

Recent advances in machine learning provide opportunities for addressing this challenge [117,118]. Reinforcement Learning (RL), in particular, enables agents to learn adaptive decision-making policies through interaction with their environment. Rather than relying on predefined rules, RL agents learn from experience by observing the consequences of their actions and gradually improving their behavior in order to maximize long-term rewards [118]. This ability makes RL particularly suitable for environments characterized by uncertainty, partial observability, and continuous change.

Building on these ideas, this dissertation investigates whether trust-model selection itself can be formulated as a learning problem. Instead of assuming that a single trust mechanism should be employed permanently, we explore the possibility that agents can dynamically select among alternative trust models according to current environmental conditions. More specifically, we examine whether a trustor can learn when it is advantageous to rely on a conventional trustor-centric approach and when a trustee-centric approach such as CA is expected to yield better outcomes.

The adaptive perspective adopted in this dissertation reflects a broader view of trust management. Rather than treating alternative trust mechanisms as competing solutions, they are viewed as complementary approaches that can be combined to exploit their respective strengths. This perspective motivates one of the central research directions of this work: the development of self-aware and adaptive trust-management frameworks capable of selecting the most appropriate trust strategy according to the evolving characteristics of their environment.

1.5. Beyond Trust Management: Decision Support and Healthcare Applications

Although computational trust originated as a mechanism for supporting partner selection and cooperation in Multi-Agent Systems, its underlying principles extend beyond traditional trust-management problems. The broader significance of computational trust

lies in its ability to support decision making under uncertainty, particularly in situations where complete information about future outcomes is unavailable. Consequently, trust-related concepts can be applied to a wide range of domains involving risk assessment, resource allocation, coordination, and adaptive decision making.

In recent years, the increasing complexity of socio-technical systems has created a growing demand for intelligent decision-support mechanisms capable of operating in dynamic environments. Such systems are encountered in diverse domains, including autonomous vehicles, the Internet of Things, collaborative robotics, smart cities, healthcare management, and human-machine teaming. In these settings, decision makers must often operate with incomplete information while balancing multiple and sometimes conflicting objectives.

Healthcare represents one of the most challenging examples of such environments. Modern healthcare systems must cope with fluctuating patient demand, limited resources, heterogeneous staff capabilities, and time-critical operational requirements. In particular, hospital Emergency Departments (EDs) constitute highly dynamic environments where delays, resource shortages, and inappropriate task allocation can directly affect patient outcomes and quality of care.

To address these challenges, researchers have increasingly employed simulation models, decision-support systems, optimization techniques, and artificial intelligence methods to improve resource allocation and operational efficiency. Simulation-based approaches, including Discrete Event Simulation, Agent-Based Simulation, and System Dynamics, have been widely used to analyze patient flow, staffing policies, and operational bottlenecks within Emergency Departments. Similarly, AI-driven decision-support systems have been proposed for staff scheduling, workload balancing, and patient-flow optimization.

Despite these advances, most existing approaches focus primarily on operational performance metrics such as waiting times, throughput, and resource utilization. Comparatively less attention has been devoted to the role of staff competence, self-assessment, and adaptive task allocation based on individual capabilities. Furthermore, computational trust concepts have only rarely been explored in healthcare environments, and existing studies typically focus on trust relationships between organizations or

healthcare stakeholders rather than on trust-informed coordination within healthcare teams.

The trustee-centric perspective introduced in this dissertation provides a natural way to address this limitation. The fundamental principle underlying the *Create Assemblies* (CA) approach—that agents evaluate their own suitability before undertaking tasks—can be interpreted as a form of competence-aware decision support. Rather than relying exclusively on external assignment mechanisms, individual agents can contribute to task-allocation decisions through self-assessment of their capabilities, availability, and expected performance. From this perspective, the healthcare application presented in this dissertation should not be viewed as a departure from the trust-management problem, but rather as an extension of the same trustee-centric principles to a different decision-making domain.

This observation motivates the extension of the CA paradigm beyond traditional trust management and into the domain of healthcare decision support. In Chapter 7, a simulation framework representing a hospital Emergency Department is developed to investigate how self-assessment and trust-informed task allocation can influence operational outcomes. The proposed framework serves as a decision-support tool that enables hospital managers and researchers to evaluate alternative operational strategies before their deployment in real-world environments.

1.5.1. Simulation of Emergency Departments

Simulation is a widely used tool for analyzing Emergency Department (ED) operations, including patient flow, resource allocation, and bottleneck identification. The three most common simulation paradigms are Discrete-Event Simulation (DES), Agent-Based Simulation (ABS), and System Dynamics (SD), each offering different levels of abstraction and analytical capabilities.

DES models patient arrivals, service times, resource utilization, and operational constraints, making it particularly effective for evaluating interventions aimed at reducing waiting times and overcrowding. For example, Hamza et al. [119] proposed SIM-PFED, a hybrid DES-ABS framework integrated with the TOPSIS multi-criteria decision-making method to optimize patient flow and reduce throughput time in EDs. Kim [120] combined DES with machine-learning techniques for dynamic physician allocation, while Castanheira-Pinto et al. [121] integrated DES with lean healthcare principles to identify

operational bottlenecks. Other studies have focused on DES calibration [122] and on combining DES with design-thinking methodologies to support organizational change [123]. Collectively, these studies demonstrate that DES provides a robust and flexible framework for modeling ED operations. However, its process-oriented nature often limits its ability to represent individual staff behaviors, autonomous decision making, and complex interpersonal interactions.

ABS extends simulation capabilities by representing patients, physicians, nurses, and other stakeholders as autonomous agents with individual attributes, goals, and decision-making rules. This approach enables the study of heterogeneous behaviors and interactions that are difficult to capture using purely process-based models. Recent applications include Digital Twin frameworks [124], modular and reusable simulation environments [125], and stakeholder-oriented modeling approaches [126]. ABS has also been employed to investigate the effects of spatial layouts on nursing efficiency [127,128] and to model team cognition and collaboration processes [129]. These studies demonstrate the suitability of ABS for capturing human-centered and context-dependent behaviors. Nevertheless, the potential of agent-based approaches to model competence assessment, self-evaluation, trust-related decision making, and adaptive task allocation remains largely unexplored in Emergency Department simulation research.

SD approaches provide a higher-level perspective by focusing on feedback loops, accumulations, and long-term system dynamics. SD models have been applied to study patient inflow–outflow dynamics [130], interventions targeting older patient populations [131], workplace violence and staff burnout [132], and resource-management strategies in resource-constrained healthcare settings [133]. Although SD is valuable for identifying structural factors that influence overall system performance, it generally lacks the granularity required to represent individual-level behavior, heterogeneous capabilities, and personalized decision-making processes.

Overall, the literature demonstrates that simulation is a well-established and valuable tool for improving Emergency Department performance. However, most existing approaches focus primarily on operational efficiency, patient flow, and resource utilization. Comparatively less attention has been devoted to modeling how individual staff members assess their own capabilities, make task-related decisions, and adapt their behavior to changing operational conditions. The simulation framework developed in this

dissertation addresses this gap by incorporating agent-specific self-assessment mechanisms inspired by computational trust research. By integrating competence-aware decision making into an Emergency Department simulation environment, the framework supports the evaluation of adaptive staffing and task-allocation strategies while providing a bridge between computational trust concepts and healthcare decision-support systems.

1.5.2. Decision Support Systems in Healthcare

Hospital managers increasingly rely on Decision Support Systems (DSS) to support resource planning, staff scheduling, patient-flow management, and operational decision making. Existing approaches range from analytical methods based on queuing theory and mathematical optimization to simulation-based and artificial-intelligence-driven frameworks capable of supporting complex decisions in dynamic healthcare environments.

A substantial body of research has focused on improving staffing and scheduling decisions in Emergency Departments. For example, Yousefi and Yousefi [134] proposed a metamodel-based simulation optimization framework that combines machine-learning techniques—including ANFIS, Feed Forward Neural Networks (FNN), Recurrent Neural Networks (RNN), and AdaBoost—with a discrete Imperialist Competitive Algorithm (ICA) to optimize staffing allocation, achieving significant reductions in door-to-doctor time. Similarly, Kim [120] integrated Discrete Event Simulation (DES) with machine learning to dynamically allocate physicians, while Ortiz-Barrios et al. [135] combined Extreme Gradient Boosting (XGBoost) with DES to predict treatment probabilities and adapt nurse staffing levels according to anticipated demand.

Other studies have addressed broader resource-allocation and capacity-planning problems. Wang et al. [136] proposed a simulation-based optimization framework incorporating part-time shifts and nonlinear stochastic programming, demonstrating that adaptive scheduling during peak demand periods can reduce waiting times and resource strain. Redondo et al. [137] investigated long-term staffing decisions under demand uncertainty using Sample Average Approximation techniques to support robust workforce planning. In addition, Castanheira-Pinto et al. [121], De Santis et al. [122], and Dosi et al. [123] demonstrated how simulation, optimization, and organizational-design methodologies can be combined to improve operational efficiency, process design, and decision-making quality in Emergency Departments.

Taken together, these studies demonstrate the considerable potential of simulation- and AI-based DSS frameworks to improve operational performance, optimize resource utilization, and support data-driven healthcare management. However, most existing approaches focus primarily on system-level performance metrics such as waiting times, throughput, staffing levels, and resource utilization. Comparatively less attention has been devoted to modeling variability in staff competence, individual decision-making behavior, and task suitability when allocating responsibilities within healthcare teams.

Recent developments in artificial intelligence have also encouraged the emergence of more personalized decision-support approaches, where predictive models adapt decisions according to patient-specific risks, acuity levels, and anticipated resource demands. Nevertheless, these approaches typically remain reactive, adjusting decisions in response to observed outcomes or changing conditions. Far less attention has been given to proactive decision-support mechanisms capable of anticipating performance variability before tasks are assigned.

The framework proposed in this dissertation addresses this limitation by using continuously updated trust and competence profiles to predict task success likelihoods in advance, enabling anticipatory and personalized allocation of staff resources.

1.5.3. Computational Trust Models

Computational trust models originated in computer science and Multi-Agent Systems research and have been widely applied in domains such as e-commerce, collaborative robotics, distributed networks, and autonomous decision-making systems. Their primary objective is to support coordination and cooperation under uncertainty by enabling agents to assess the reliability, competence, and expected behavior of potential interaction partners. As discussed in previous sections, computational trust mechanisms provide a structured framework for making decisions when complete information is unavailable and future outcomes cannot be predicted with certainty.

Many of the challenges addressed by computational trust research are also present in healthcare environments. Healthcare systems involve multiple autonomous actors, heterogeneous capabilities, limited resources, and decisions that must often be made under conditions of uncertainty. Consequently, trust-related concepts have the potential to support coordination, collaboration, and task-allocation decisions in complex healthcare

settings. Despite this conceptual alignment, applications of computational trust within healthcare remain relatively limited.

One of the few examples is the work of Anjana and Singhal [138], who investigated trust dynamics in healthcare networks using a multi-agent, game-theoretic simulation framework. Their study modeled interactions among hospitals, physicians, researchers, insurance providers, and patients within an Electronic Medical Record (EMR)-sharing environment. By simulating repeated cooperation and defection strategies, the authors demonstrated how trust relationships evolve over time and eventually converge toward equilibrium states. Their results illustrate the potential value of computational trust for supporting policy design and coordination in healthcare ecosystems.

However, existing studies have primarily focused on trust relationships between organizations and stakeholders operating at the healthcare-system level. Comparatively little attention has been devoted to the use of computational trust for supporting coordination within healthcare teams, particularly in highly dynamic environments such as Emergency Departments. As a result, the potential role of trust-inspired mechanisms in staff collaboration, competence-aware task allocation, and adaptive operational decision making remains largely unexplored.

This gap provides an important motivation for the research presented in this dissertation. By integrating computational trust concepts with simulation-based decision support, the proposed framework explores how trust-inspired self-assessment and competence-aware task allocation can contribute to more adaptive and personalized healthcare operations. In doing so, it establishes a bridge between computational trust research and emerging AI-driven approaches to healthcare management.

1.5.4. Personalized Medicine and Adaptive Staffing

Personalized medicine has traditionally focused on tailoring diagnosis and treatment to the characteristics of individual patients. With the emergence of artificial intelligence and data-driven healthcare, this concept has gradually expanded beyond clinical decision making toward operational personalization, where resource allocation and workflow management are adapted to the specific needs and circumstances of each patient. Savchenko and Bunimovich-Mendrazitsky [139] investigate the economic feasibility of personalized medicine for healthcare service providers using bladder cancer as a case study and propose a framework for identifying patient cohorts that balances

clinical effectiveness with operational efficiency. Building on this idea, Gummadi [140] presents a multimodal AI framework that integrates Electronic Health Records, patient-reported outcomes, genomic information, and real-time physiological data from wearable sensors to construct comprehensive patient profiles. Beyond supporting personalized treatment decisions, this approach can also inform workflow optimization and staffing policies, leading to improvements in clinical outcomes, patient satisfaction, and provider efficiency.

The growing emphasis on personalization has also influenced research on emergency care operations. Hodgson et al. [141] applied AI and machine-learning techniques to optimize Emergency Department (ED) patient flow through a personalized vertical processing pathway, demonstrating how individualized risk scores can support flexible patient-specific care protocols that improve throughput without compromising safety. At the operational level, Chan et al. [142] investigated dynamic server assignment in multiclass queuing systems with shift-based reassignment constraints, motivated by nurse staffing in EDs. Their results show that partial flexibility—where reassignment is possible only at shift boundaries—can substantially reduce waiting costs compared with static staffing policies while remaining operationally feasible. Collectively, these studies demonstrate that personalized approaches can influence not only clinical interventions but also operational planning, resource allocation, and workforce management in complex healthcare environments.

In practice, however, staffing adaptation is not limited to strategic scheduling decisions. Effective ED operation often requires continuous coordination within shifts, where tasks must be redistributed in response to changing workloads, evolving patient needs, or variations in staff performance. The framework proposed in this dissertation focuses on this form of intra-shift adaptability by investigating how trust-informed task allocation and self-assessment mechanisms can support dynamic task redistribution while maintaining both efficiency and safety.

Despite recent advances, most existing approaches remain primarily predictive or reactive, adapting decisions after changes in patient conditions or system performance have already occurred. Comparatively little attention has been devoted to simulation-based frameworks that combine personalized patient information, computational trust, and adaptive staffing mechanisms to proactively support decision making in real time. To

address this gap, this dissertation introduces a Unity-based simulation framework that incorporates computational trust models into Emergency Department operations. The proposed framework is used to investigate how trust-informed and competence-aware task allocation can affect operational efficiency, staff development, and patient-centered outcomes, thereby contributing to the broader vision of AI-supported personalized healthcare operations.

1.6. Research Objectives and Research Questions

The central objective of this dissertation is to investigate whether trust management in open and dynamic Multi-Agent Systems can be improved by shifting part of the trust formation process from the trustor to the trustee and by introducing adaptive mechanisms capable of selecting the most appropriate trust strategy according to environmental conditions.

Existing trust-management approaches predominantly follow a trustor-centric paradigm, where service requesters estimate the trustworthiness of potential service providers and select interaction partners accordingly. While this approach has produced numerous successful trust models, it continues to face important challenges related to agent mobility, behavioral volatility, information scarcity, communication overhead, and resistance to malicious behavior.

To address these limitations, this dissertation explores an alternative trustee-centric paradigm, implemented through the *Create Assemblies* (CA) trust model. Rather than relying exclusively on trustor evaluations, the proposed approach allows trustees to actively participate in the trust-management process through self-assessment of their own capabilities and suitability for task execution.

These objectives give rise to the following primary research question:

RQ1: *Can a trustee-centric trust-management approach provide an effective alternative to conventional trustor-centric trust models in open and dynamic Multi-Agent Systems?*

To answer this question, several more specific research questions are investigated.

The first group of questions examines the behavior and performance of the trustee-centric approach:

RQ2: *What advantages and limitations emerge when trust formation is partially delegated to trustees through self-assessment mechanisms?*

RQ3: *Under which environmental conditions does the trustee-centric CA approach outperform conventional trustor-centric trust models, and under which conditions does it perform less effectively?*

The second group of questions focuses on adaptation and behavioral change:

RQ4: *Can trustee self-assessment mechanisms improve the ability of the system to detect and respond to changes in agent performance profiles?*

RQ5: *How can trust-management mechanisms remain effective when agent capabilities evolve dynamically during system operation?*

The third group of questions addresses adaptive trust management and learning-based trust-model selection:

RQ6: *Can trust model selection itself be formulated as a learning problem?*

RQ7: *Can reinforcement learning enable agents to dynamically select between alternative trust-management mechanisms according to environmental conditions and thereby improve decision-making performance?*

Finally, this dissertation investigates whether the principles underlying the trustee-centric perspective can be applied beyond traditional trust-management problems:

RQ8: *Can trustee self-assessment and trust-informed coordination be utilized as decision-support mechanisms in complex real-world domains such as healthcare operations?*

The answers to these research questions are developed progressively throughout the dissertation. Chapters 2 and 3 introduce and evaluate the trustee-centric paradigm and its extensions. Chapter 4 investigates adaptive trust-model selection through Deep Reinforcement Learning. Chapters 5 and 6 provide a broader evaluation of the proposed approach against trust-management requirements and security-related challenges. Chapter 7 explores the application of trustee-centric principles to healthcare decision support through an emergency department simulation framework.

Collectively, these investigations seek to advance both the theoretical understanding of computational trust and the practical development of trustee-centric and adaptive trust-management mechanisms. More broadly, they contribute to the design of intelligent decision-support frameworks capable of operating effectively in complex, uncertain, and continuously evolving environments.

1.7. Dissertation Structure

This dissertation is organized into eight chapters that progressively introduce, develop, evaluate, and extend the proposed trustee-centric trust-management paradigm.

Chapter 1 introduces the research area and examines the role of computational trust in open and dynamic Multi-Agent Systems. It discusses the limitations of existing trust-management approaches, motivates the trustee-centric perspective adopted in this work, reviews the relevant literature, and formulates the research objectives and questions that guide the dissertation.

Chapter 2 introduces the *Create Assemblies* (CA) trust model. Drawing inspiration from biological mechanisms of synaptic plasticity, the chapter explains how these principles are translated into a trustee-centric trust framework. It formally defines the model and its components and discusses the main advantages of the proposed approach.

Building on this foundation, Chapter 3 provides a comparative evaluation of trustee-centric and conventional trust approaches. The performance of CA is compared with that of the FIRE trust and reputation model under a variety of dynamic environmental conditions, including changes in consumer populations, provider populations, and provider behavior profiles. The results highlight the strengths and limitations of both approaches and motivate the need for adaptive trust-management mechanisms.

Chapter 4 addresses this challenge by investigating adaptive trust-model selection through reinforcement learning. The trust-model selection problem is formulated as a Deep Reinforcement Learning task, and a Deep Q-Network is employed to dynamically select between alternative trust mechanisms according to environmental conditions. The effectiveness of the resulting adaptive trustor is evaluated across a wide range of dynamic scenarios.

Chapter 5 further extends the CA paradigm by introducing a self-assessment mechanism that enables trustees to detect and respond to changes in their own performance. An enhanced version of the CA algorithm is proposed and experimentally evaluated in environments characterized by dynamic trustee behavior. The chapter identifies the conditions under which the original and extended versions of CA are most effective.

Chapter 6 presents a broader assessment of the *Create Assemblies* paradigm. The model is evaluated against a comprehensive set of trust-management requirements,

including decentralization, scalability, privacy preservation, explainability, and robustness. The chapter also examines the resilience of CA against common trust-related attacks and discusses potential mechanisms for strengthening its resistance to malicious behavior.

Moving beyond traditional trust-management applications, Chapter 7 explores the applicability of trustee-centric principles to healthcare decision support. A simulation framework representing a hospital Emergency Department is developed to investigate how trust-informed task allocation and self-assessment mechanisms can support operational decision making. The chapter evaluates alternative staffing and training policies and demonstrates the potential of CA as a decision-support mechanism in complex socio-technical environments.

Finally, Chapter 8 summarizes the main research contributions of the dissertation, discusses the limitations of the proposed approaches, and outlines several directions for future work.

The appendices provide supplementary material on the biological foundations of the proposed model, background information on reinforcement learning, and additional technical details concerning the implementation and computational complexity of the proposed algorithms.

2. The Create Assemblies (CA) Trust Model

Most computational trust and reputation models adopt a trustor-centric perspective, according to which trustors are responsible for evaluating available trustees and selecting those considered most trustworthy for a given task. Although this approach has been extensively studied and successfully applied in a wide range of distributed systems, it also introduces several challenges, particularly in highly dynamic and open Multi-Agent Systems (MAS), where agents continuously enter and leave the environment and where the capabilities and behavior of service providers may change over time.

To address these challenges, this dissertation investigates an alternative perspective on computational trust, namely the trustee-centric approach. Instead of requiring trustors to estimate and compare the trustworthiness of potential service providers, the proposed model allows trustees to evaluate their own capability to successfully perform a task and to decide whether they should undertake it. This shift in perspective is inspired by biological mechanisms of adaptation observed in synaptic plasticity and leads to a fundamentally different interpretation of trust formation and trust management in MAS.

This chapter presents the *Create Assemblies* (CA) trust model, a biologically inspired trustee-centric computational trust framework. The chapter first introduces the biological principles that motivated the model and establishes the correspondence between synaptic plasticity and trust relationships among agents. It then formalizes the trust representation and the trust adaptation mechanism, describes the incorporation of trustor feedback through a reward-punishment process, and presents the *Create Assemblies* algorithm governing agent interactions. Finally, the main advantages of the proposed trustee-centric approach are discussed, highlighting its suitability for open and highly dynamic environments.

2.1. Biological Inspiration

The inspiration for the proposed trust model originates from synaptic plasticity, a fundamental biological mechanism of the human brain through which the connections between neurons are strengthened or weakened over time according to their interaction patterns.

One of the first researchers to formulate the idea of the plastic nature of the brain was the Canadian psychologist Donald Hebb [143]. Hebb proposed that when two neurons repeatedly fire together in a coordinated manner, the connection between them becomes stronger, increasing the probability that they will activate together again in the future. This principle later became widely known through the phrase "cell that fire together wire together".

2.2. From Synaptic Plasticity to Trustee-Centric Trust Modeling

Applying Hebb's principle to a Multi-Agent System (MAS) leads to a natural analogy: neurons are replaced by agents, while the coordinated activation of neurons is replaced by successful cooperation among agents during task execution.

To better understand this analogy, it is important to briefly examine the biological process of synaptic plasticity and how it is translated into the proposed computational trust framework. In the human brain, neurons communicate through electrical and chemical signals. When an electrical signal reaches the terminal of a presynaptic neuron, neurotransmitters, primarily glutamate, are released into the synaptic cleft. These neurotransmitters bind to specialized proteins, known as glutamate receptors, located on the membrane of the postsynaptic neuron. This interaction initiates a cascade of intracellular processes, increasing the flow of ions through the membrane and causing a change in the membrane potential of the postsynaptic neuron, which may eventually lead to its activation (Figure 1).

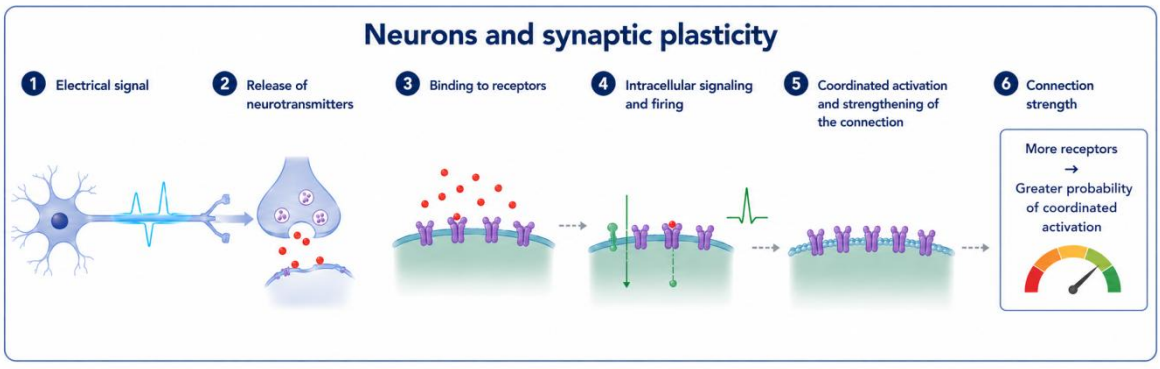


Figure 1. Synaptic plasticity. An action potential in a presynaptic neuron leads to neurotransmitter release and receptor activation in a postsynaptic neuron. Coordinated firing of the two neurons strengthens the

synapse by increasing the number of postsynaptic receptors. As a result, stronger synaptic connections are more likely to produce coordinated activation in the future.

If the postsynaptic neuron fires within a critical temporal window of approximately 20 milliseconds after the presynaptic neuron, the two neurons are considered to have activated in a coordinated manner. This successful coordination leads to the strengthening of the synaptic connection, a process carried out by the postsynaptic neuron itself through an increase in the number of available glutamate receptors on its membrane. The greater the number of receptors available, the more sensitive the postsynaptic neuron becomes to future activations by the same presynaptic neuron. Conversely, if the presynaptic neuron fires after the post synaptic neuron, again within a similar temporal window, the synaptic connection is weakened through a reduction in the number of available receptors on the postsynaptic membrane. Consequently, the strength of a synaptic connection depends largely on the adaptive behavior of the postsynaptic neuron.

Since a neuron may form synaptic connections with many other neurons simultaneously, signals arriving from neurons associated with stronger synaptic connections are more likely to trigger future activations. A more detailed description of synaptic plasticity as a biological process is provided in Appendix A.

This biological mechanism naturally incorporates the idea that when two units repeatedly cooperate successfully, the connection between them becomes stronger, increasing the likelihood of future collaboration. Inspired by this principle, the proposed computational trust model establishes a correspondence between biological neurons and interacting agents in a MAS.

More specifically, the presynaptic neuron is associated with the Trustor, while the postsynaptic neuron corresponds to the Trustee. An important observation is that the strengthening or weakening of the biological connection is performed by the postsynaptic neuron itself. In other words, all adaptive changes that increase or decrease the probability of future interactions are controlled by the receiving neuron. This observation naturally leads to a trustee-centric trust model, in which the Trustee evaluates its own trustworthiness based on the success or failure of previous interactions and adaptively updates the strength of its trust relationships over time.

2.3. Mathematical Formulation

Building on the biological analogy presented in the previous section, we now formalize the proposed computational trust model. Trust is represented from the perspective of the trustee, with each trustee continuously assessing its own ability to successfully execute tasks and adaptively updating its trust relationships over time.

Consider a Multi-Agent System (MAS) composed of autonomous agents that interact to accomplish tasks within a shared environment. The environment is modeled as a weighted directed graph:

$$G = (V, E)$$

where $V = \{u_1, u_2, \cdots, u_n\}$ is the set of agents and E is the set of directed trust relationships between them. Each directed edge represents a connection from a trustor to a trustee and is associated with a trust value:

$$w_{ij} \in [0,1]$$

where w_{ij} denotes the strength of the trust relationship between trustor u_i and trustee u_j, with $w_{ij} = 0$ representing complete lack of trust and $w_{ij} = 1$ representing maximum trust.

2.3.1. Task and Task Categories

The environment contains different categories of tasks. Let $C = \{c_1, c_2, \cdots, c_m\}$ be the set of all task categories.

A task is defined as the pair $task = (c, r)$

where:

- $c \in C$ denotes the task category,
- $r = \{r_1, r_2, \cdots, r_k\}$ is a set of requirements specified by the trustor.

The requirements describe the conditions under which the task is considered successfully completed. Examples may include execution deadlines, quality constraints, resource limitations, or other domain-specific criteria.

The set of all possible tasks is denoted by T.

When an agent identifies a task that must be performed it assumes the role of trustor. The trustor specifies the task requirements and broadcasts a request for task execution to the environment. Any agent receiving the request may act as a trustee. Unlike conventional trust models, the trustee is not selected by the trustor. Instead, each trustee autonomously evaluates whether it possesses the necessary capabilities to perform the

requested task. Following task execution, the trustee updates its own estimate of trustworthiness based on the outcome.

2.3.2. Trust Connections

Each trustee maintains locally a collection of such connections (Figure 2). Conceptually, this structure resembles the set of synaptic connections maintained by a biological neuron. Each connection stores three essential pieces of information: the identity of the trustor, the trust value, and the task specification.

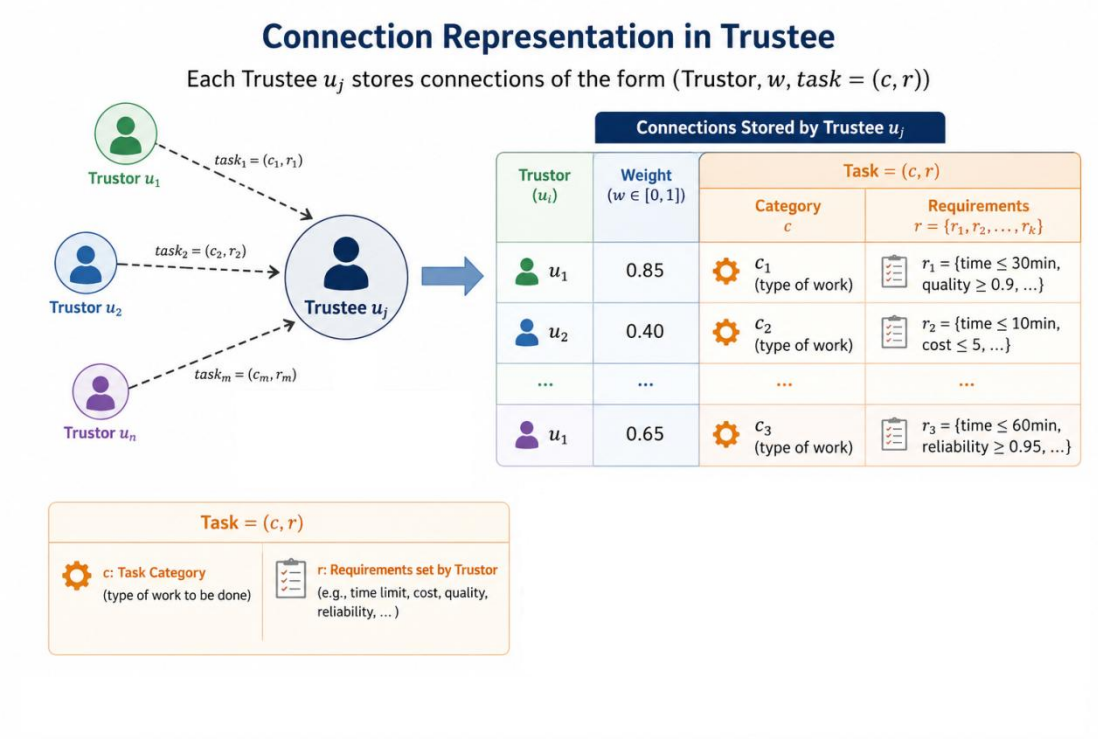


Figure 2. The Trustee stores, for each Trustor and for each requested task, a connection with a weight representing the trust value

Formally, a connection is defined as:

$$Co = (u_i, w, task)$$

where:

- u_i is the trustor,
- $w \in [0,1]$ is the trust value associated with the interaction,
- $task \in T$ is the specific task to which the connection refers.

This formulation reflects two key properties of trust: **context-dependence** and **subjectivity**. Trust is context-dependent because it is associated with a specific task.

Therefore, a trustee that is highly trustworthy for one type of task may be considerably less trustworthy for another. Trust is also subjective because trust values are maintained separately for different trustors. Since each trustee stores an independent connection for every trustor-task pair, the trustworthiness associated with the same task may vary across trustors depending on the outcomes of their past interactions. Consequently, trust emerges as a localized and personalized assessment rather than as a universal property of an agent.

Through repeated interactions, these connections become progressively strengthened or weakened, allowing the trustee to accumulate experience regarding which tasks it can successfully perform and for which trustors.

2.3.3. Decision Threshold

To avoid repeatedly attempting tasks for which it has demonstrated poor performance, each trustee uses a trust threshold

$$\theta \in [0,1]$$

A trustee considers executing a task only if the trust value of the corresponding connection exceeds this threshold:

$$w \geq \theta$$

This mechanism is analogous to biological neurons requiring sufficient synaptic strength before reliably participating in coordinated activation patterns. In the proposed model, the threshold serves as a self-regulation mechanism that prevents the unnecessary expenditure of computational resources on tasks for which the trustee has accumulated evidence of low competence.

Overall, the model provides a decentralized and self-organizing representation of trust in which trust values emerge from the trustee's own experience. Repeated successful executions strengthen the corresponding trust connections, while repeated failures weaken them, closely mirroring the strengthening and weakening of biological synapses observed in synaptic plasticity.

2.3.4. Trust Update Mechanism

The strength of each trust connection is updated by the trustee after every task execution (event-driven update). Consistent with the biological inspiration of synaptic plasticity, successful interactions strengthen the corresponding connection, whereas unsuccessful interactions weaken it.

Following a successful task execution, the trustee updates the trust value according to:

$$w = Min\big(1, w + \alpha(1 - w)\big) \quad (1)$$

where $\alpha > 0$ is a parameter controlling the rate of the increase.

Conversely, if the task execution fails, the trustee decreases the trust value according to:

$$w = Max\big(0, w - \beta(1 - w)\big) \quad (2)$$

where $\beta > 0$ is a parameter controlling the rate of the decrease.

As a result, repeated successful interactions progressively strengthen a connection, while repeated failures gradually weaken it.

2.4. Reward-Punishment Mechanism

A fundamental property of the proposed trust model is its preservation of subjectivity, one of the most important characteristics of trust. Trust is inherently dependent on the perspective, expectations, and objectives of the evaluating entity. In the proposed framework, subjectivity is maintained in two complementary ways. First, the trustor explicitly defines the requirements associated with a task, thereby determining the conditions under which the task will be considered successfully completed. Second, the evaluation of the provided service is performed according to the trustor's own needs, expectations, and contextual constraints through a reward-punishment mechanism that expresses the trustor's satisfaction.

After a task has been executed, the trustor assesses the quality of the received service with respect to the specified task requirements. If the delivered service satisfies the trustor's expectations and fulfills the agreed requirements, the trustor provides a reward to the trustee (service provider). Such a reward may take different forms depending on the application domain and may, for example, correspond to a monetary payment, access to resources, additional privileges, or any other benefit agreed upon by the participating entities prior to task execution.

Conversely, if the trustor is not satisfied with the quality of the provided service or determines that the task requirements have not been adequately fulfilled, the trustee does not receive the agreed reward. In some environments, dissatisfaction may also lead to an explicit punishment. Punishments may include utility loss, financial penalties, temporary

restrictions on participation, or even temporary or permanent exclusion from the system. Such mechanisms provide incentives for trustees to maintain high-quality service provision while discouraging selfish, unreliable, or malicious behavior.

The trustee subsequently uses this feedback to adapt the strength of the corresponding trust connection. A received reward is interpreted as evidence that the trustee successfully satisfied the trustor's requirements, leading to an increase in the connection weight. In contrast, the absence of a reward or the application of a punishment is interpreted as evidence of an unsuccessful interaction, resulting in a decrease in the connection weight. The magnitude of these adjustments is determined by the trust update equations presented in the previous section.

By incorporating the trustor's subjective evaluation into the trust update process, the model preserves the personalized nature of trust relationships. Different trustors may define different requirements for the same task category and may evaluate identical service outcomes differently according to their individual objectives and contextual constraints. Consequently, the same trustee may maintain different trust values for the same type of task when interacting with different trustors, naturally capturing the subjective nature of trust in multi-agent environments.

2.5. The Create Assemblies Algorithm

```
Algorithm 1: Create Assemblies (CA)

1:  while True do

    # --- Task perception and broadcast (Trustor role) ---
2:      when a new task task = (c, r) is perceived do
3:          broadcast message m = (request, i, task)
4:      end when

    # --- Request reception and connection initialization (Trustee role) ---
5:      when message m = (request, j, task) is received do
6:          add m to message list M

7:          if no connection Co = (j, w, task) exists then
8:              create Co = (j, 0.5, task)
9:          end if
10:     end when

    # --- Task selection ---
11:     select m = (request, j, task) from M such that
        Co = (j, w, task) has the highest trust value w among all stored
connections

    # --- Execution decision ---
12:     if task not completed and w ≥ θ then

13:         outcome ← performTask(task)
14:         feedback ← rewardPunishmentEvaluation(outcome, task)

15:         if feedback = reward then
16:             w ← min(1, w + α(1 − w))
17:         else
18:             w ← max(0, w − β(1 − w))
19:         end if

20:     end if

    # --- Remove processed request ---
21:     delete m from M

22: end while
```

After defining the trust representation and the trust update mechanism, we now describe the operational procedure through which agents interact and adapt their trust relationships over time. The complete pseudocode of the proposed algorithm is presented in Algorithm 1.

In the *Create Assemblies* (CA) algorithm, every agent can act both as a trustor and a trustee. When an agent perceives a new task (line 2), it acts as a trustor and broadcasts a request message containing the task specification to all agents in the environment, including itself (line 3). The message contains the identity of the requesting agent (the trustor), together with the task specification, including its category and the associated requirements that define the conditions for successful task completion.

When a candidate trustee receives a request message (line 5), it stores the message in its local list of pending requests M (line 6). If no trust connection already exists between the trustee and the trustor for the specific task (line 7), a new connection is created and initialized with a neutral trust value of 0.5 (line 8). This initial value reflects the absence of a previously established trust relationship between the trustee and the specific trustor for the requested task.

Upon processing a request, the trustee selects from its pending requests the task associated with the strongest trust connection (line 11). This mechanism is inspired by synaptic plasticity, where stronger biological connections are more likely to participate in future activations. Because task requests are broadcast to multiple agents, several trustees may independently consider undertaking the same task. Therefore, before attempting execution, the trustee first verifies whether the task has already been completed by another agent (line 12). This check prevents redundant task execution and unnecessary resource consumption in the decentralized environment. The trustee then determines whether the corresponding trust value exceeds the predefined decision threshold θ (line 12). The threshold acts as self-regulation mechanism, preventing trustees from repeatedly allocating resources to tasks for which they have accumulated evidence of poor performance. If both conditions hold, the trustee attempts to perform the task by invoking the *performTask* procedure (line 13).

Following task execution, the trustor evaluates the delivered service according to the reward-punishment mechanism described in Section 2.4. The resulting feedback reflects whether the task requirements and the trustor's expectations have been satisfactorily fulfilled. The trustee then uses this feedback to update the corresponding trust connection. If the trustee receives a reward (line 14), the interaction is interpreted as successful and the connection is strengthened according to the reinforcement rule presented in Section 2.3.4 (lines 15-16). Otherwise, the interaction is treated as unsuccessful and the connection is weakened using the corresponding penalty rule (lines 17-19). In this way, trust values gradually adapt to the trustee's observed performance over time as evaluated by the trustor. Finally, the processed request is removed from the pending-request list (line 21). Through repeated interactions, trust connections become progressively strengthened or weakened, allowing trustees to accumulate experience

regarding the tasks they can successfully perform and thereby self-organize their future participation decisions.

2.6. Key Advantages of Trustee-Centric Trust Management

The *Create Assemblies* (CA) model departs from conventional trust management approaches in two fundamental ways. First, trust is represented and maintained locally by the trustee rather than by the trustor. Second, the trustor does not select a trustee; instead, candidate trustees autonomously decide whether they are capable of performing the requested task. These design choices provide several important advantages, particularly in open and highly dynamic multi-agent environments.

2.6.1. Mobility and Reusability of Trust Information

Since trust information is stored locally by the trustee, each agent carries with it the knowledge accumulated through its past interactions. Consequently, trust values are not tied to a particular platform, application, or organizational context. A trustee can reuse the experience acquired in one environment when participating in another, making the model particularly suitable for highly dynamic systems characterized by frequent agent mobility. This property addresses one of the longstanding challenges in distributed and open multi-agent systems, namely the preservation of trust knowledge when agents move across different environments.

2.6.2. Reduced Communication Overhead

Traditional trust models often require trustors to collect trust information from multiple sources before selecting a trustee. Such approaches rely on the exchange of recommendations, reputation scores, or historical evaluations, resulting in additional communication costs and increased decision-making latency. In contrast, the CA model does not require agents to exchange trust information. Trustees make decisions based solely on their locally maintained trust connections and accumulated experience. As a result, the communication overhead is significantly reduced, improving the scalability of the system.

2.6.3. Resistance to Disinformation Attacks

Beyond reducing communication costs, the absence of recommendation exchange also provides an important security advantage. In many reputation-based systems,

malicious agents can manipulate trust assessments by disseminating false evaluations about service providers. Examples include bad-mouthing attacks, in which attackers deliberately submit negative evaluations to damage the reputation of an honest provider, and ballot-stuffing attacks, in which colluding agents submit artificially positive evaluations to promote a malicious or low-quality provider.

Since the CA model does not rely on recommendations or reputation dissemination, such attacks cannot directly influence trust formation. Trust relationships are established exclusively through direct interactions and personal experience, making the model inherently resilient to a broad class of disinformation attacks.

2.6.4. Enhanced Privacy Preservation

Finally, the lack of trust information exchange contributes to stronger privacy protection. In conventional trust and reputation systems, agents are often required to reveal their evaluations, preferences, or interaction histories to other participants. Such information may expose sensitive details regarding an agent's behavior, decision-making criteria, or service consumption patterns. In the CA model, trust assessments remain local to the trustee and are never disseminated throughout the network. Consequently, agents are not required to disclose private judgments about other agents, reducing the risk of privacy leakage while preserving the autonomy of individual participants.

Overall, the CA approach combines decentralized trust management, low communication requirements, improved privacy preservation, and inherent resistance to disinformation attacks, making it particularly suitable for open, dynamic, and large-scale Multi-Agent Systems.

3. Comparative Evaluation of Trustor-Centric and Trustee-Centric Trust Models

The conventional trustor-centric paradigm has dominated computational trust research for several decades, with most trust and reputation models assuming that trustors are responsible for evaluating and selecting trustees. In contrast, the *Create Assemblies* (CA) model introduced in the previous chapter proposes a fundamentally different perspective: a trustee-centric approach in which trustees assess their own trustworthiness and decide whether they are capable of successfully undertaking a requested task. This shift in perspective raises important questions regarding the relative advantages, limitations, and applicability of the two approaches, particularly in dynamic and open Multi-Agent Systems (MAS).

This chapter presents a comparative study between the trustee-centric CA approach and the conventional trustor-centric paradigms. To this end, the proposed CA model is experimentally compared with FIRE, a well-established trust and reputation model for open MAS that follows the traditional trustor-centric approach. Through a series of simulation experiments, we investigate how the two models respond to key challenges associated with open and dynamic environments, including agent population turnover and changes in trustees' performance profiles. The objective is not only to evaluate the effectiveness of the proposed trustee-centric approach but also to identify the conditions under which each paradigm offers distinct advantages.

3.1. The FIRE Trust and Reputation model

To evaluate the proposed trustee-centric approach, we compare the CA model with FIRE [8], a well-established trust and reputation model for open Multi-Agent Systems (MAS). FIRE was selected as a reference model because it follows a decentralized and distributed architecture, similarly to CA, while representing the conventional trustor-centric paradigm, in which trustors evaluate and select trustees for task delegation.

FIRE estimates the trustworthiness of a target agent by combining information from four distinct trust sources:

- Interaction Trust (IT), which is derived from the evaluator's direct experiences and previous interactions with the target agent.
- Witness Reputation (WR), which is based on the opinions and evaluations provided by other agents who have previously interacted with the target agent.
- Role-Based Trust (RT), which assesses trustworthiness using role relationships and domain knowledge, such as norms, rules, and organizational structures.
- Certified Reputation (CR), which relies on third-party references and certificates maintained by the target agent and made available upon request.

The FIRE model combines these four sources of trust information to produce an overall trust estimate, which is then used by the trustor to select an appropriate trustee for task execution. Direct interaction experience is generally considered the most reliable source of trust information, while the remaining components provide complementary evidence when limited or no direct interaction history is available.

The incorporation of multiple trust sources allows FIRE to operate effectively in open and decentralized environments and has contributed to its widespread adoption as a benchmark model in computational trust research. For this reason, FIRE constitutes an appropriate reference point for evaluating the properties and performance of the proposed CA model. Moreover, FIRE provides a useful comparison baseline because it embodies the traditional trustor-centric view of trust management, thereby allowing the fundamental differences between trustor-centric and trustee-centric approaches to be systematically evaluated.

3.2. Extending CA for Dynamic Trustee Profiles

The CA algorithm presented in the previous chapter was designed under the assumption that a trustee's capabilities remain relatively stable over time and that trust values gradually converge through repeated task execution. However, one of the objectives of the present study is to evaluate trustee-centric trust mechanisms in dynamic environments, where trustees may improve, deteriorate, or even change their performance profiles over time. To support this investigation, an extended version of the CA algorithm was developed, capable of adapting to dynamic trustee profiles.

The first extension introduces a mechanism that allows trustees to revise their previous assessments of their own capabilities. In the original CA algorithm, when the

trust value associated with a task falls below a predefined threshold, the trustee interprets this as evidence that it is unable to perform the task successfully and consequently stops attempting it. While this behavior is appropriate in static environments, it becomes problematic when a trustee's capabilities can evolve over time.

To address this limitation, the extended algorithm exploits information obtained from the execution of simpler tasks belonging to the same task category. If a trustee demonstrates satisfactory performance on an easier task, this is interpreted as evidence that its capabilities may have improved. Consequently, the trust values associated with more demanding tasks of the same category, which had previously fallen below the threshold, are restored to the threshold level. This mechanism allows the trustee to reconsider tasks it had previously abandoned and enables learning and capability improvement to be reflected in its future behavior. This dynamic profile adaptation mechanism is implemented in lines 25-31 of Algorithm 2.

A second extension addresses the cold-start problem that arises when a trustee receives a task request from a trustor with whom it has no previous interaction history. In the original CA algorithm, a newly created trust connection is initialized with a predefined value. In the updated version, the trustee first examines whether it has previously performed the same task for other trustors. If such experience exists, the new connection is initialized using the average trust value of the corresponding existing connections. Otherwise, the predefined initial value is used. This modification allows trustees to exploit previously acquired knowledge when interacting with new trustors. The corresponding connection-initialization procedure is implemented in lines 6-13 of Algorithm 2.

Algorithm 2 presents the pseudocode of the extended CA algorithm employed throughout the experimental study reported in this chapter.

```
Algorithm 2: Create Assemblies (CA) for Dynamic Trustee Profiles

1:  while True do

    # --- Task perception and request broadcasting ---
2:      when a new task task = (c, r) is perceived do
3:          broadcast message m = (request, i, task)

    # --- Request reception and connection initialization ---
4:      when message m = (request, j, task) is received do
5:          add m to message list M

6:          if no existing connection Co = (j, w, task) exists then
7:              if ∃ similar connections Co' = (k, w', task) for other trustors
k then
8:                  avg_w ← average weight of all Co' for task
9:                  create Co = (j, avg_w, task)
10:             else
11:                 create Co = (j, 0.5, task)
12:             end if
13:         end if


    # --- Task selection ---
14:     select m = (request, j, task) from M such that
        Co = (j, w, task) has the highest trust value w among all stored
connections

    # --- Execution decision ---
15:     if task not completed and w ≥ θ then

16:         outcome ← performTask(task)
17:         feedback ← rewardPunishmentEvaluation(outcome,task)

18:         if feedback = reward then
19:             w ← min(1, w + α(1 − w))
20:         else
21:             w ← max(0, w − β(1 − w))
22:         end if

23:     end if

    # --- Remove processed request ---
24:     delete m from M

    # --- Dynamic trustee profile adaptation ---
25:     for all connections Co' = (j, w', task') where w' < θ do
26:         if task' = (c, r') and r' > r then
27:             if performance ≥ minSuccessfulPerformance(task') then
28:                 w' ← θ   # re-enable previously rejected tasks
29:             end if
30:         end if
31:     end for

32: end while
```

3.3. Experimental Environment

To evaluate and compare the behavior of the proposed *Create Assemblies* (CA) model against the FIRE trust and reputation model, a simulation testbed was implemented based on the experimental environment originally proposed for FIRE. The testbed was

designed to emulate an open and highly dynamic Multi-Agent System (MAS), allowing the systematic investigation of how different trust paradigms respond to population changes, agent mobility, and evolving trustee behavior.

In this study, openness is considered only with respect to agents. Specifically, agent addition and removal are represented through continuous population turnover, while agent evolution is represented through changes in providers' performance over time, allowing agents to improve, deteriorate, or alter their performance profiles. These forms of environmental dynamics provide the basis for the experimental scenarios examined in this chapter. Openness related to tasks and resources is not considered in the present study and is left for future investigation.

The simulated environment consists of two categories of agents: providers, which offer services and act as trustees, and consumers, which request and use services acting as trustors. For simplicity, all providers offer the same type of service. Agents are randomly distributed within a spherical world of radius 1.0. Each agent is characterized by a radius of operation r_0, which represents its ability to interact with other agents. Agents located within this radius are considered acquaintances and can potentially participate in service interactions.

The quality of the service delivered by a provider determines the utility gained by the consumer from an interaction. This utility gain (UG) constitutes the outcome used by the trust models to assess trustee performance. Providers are assigned one of four performance profiles: good, ordinary, bad, and intermittent. For good, ordinary, and bad providers, service performance follows a normal distribution around a profile-specific mean performance level μ_p, with standard deviation σ_p. Intermittent providers exhibit unpredictable behavior, producing performance values randomly within the interval [PL_BAD, PL_GOOD]. The performance characteristics of the provider profiles are presented in Table 1, while the corresponding performance-level constants are summarized in Table 2.

Table 1. Profiles of provider agents (performance constants defined in Table 2)

Profile	Performance range	$\boldsymbol{\sigma_p}$
Good	[PL_GOOD, PL_PERFECT]	1.0
Ordinary	[PL_OK, PL_GOOD]	2.0

Bad	[PL_WORST, PL_OK]	2.0

Table 2. Performance level constants

Performance level	Utility gained
PL_PERFECT	10
PL_GOOD	5
PL_OK	0
PL_BAD	-5
PL_WORST	-10

In addition to defining communication capabilities, the operational radius of a provider also represents its normal service range. Within this range, the provider can deliver its service at full quality. When a consumer is located outside the provider's operational range, service quality degrades linearly as a function of the distance between the two agents. The resulting performance value is bounded within the interval [-10, +10] and corresponds directly to the utility gain obtained by the consumer.

Simulations are executed in discrete rounds. As in real world environments, consumers do not require services continuously. When a consumer agent is created, it is assigned an activity level a, representing the probability that it will require a service during a given round. If a service is required, the consumer always initiates a service request during that round. The round number serves as a temporal reference for all events occurring within the simulation.

Consumers belong to one of three categories:

- Consumers without a trust model
- Consumers using the FIRE trust model
- Consumers using the CA trust model

When a consumer requires a service, it first identifies all nearby providers within its interaction range. Consumers without a trust model select a provider randomly. Consumers using FIRE evaluate available providers according to FIRE's trust assessment mechanism and select a provider based on the resulting trust estimates. After service execution, the consumer receives a utility gain and records an evaluation equal to the

obtained utility. This evaluation is subsequently used for future trust assessments. The provider is also informed of the evaluation and may retain it as evidence of its past performance for future interactions.

The interaction process differs substantially for consumers employing the CA model. Rather than selecting a provider, a CA consumer broadcasts a request message to all nearby providers. The message specifies the required service quality, which constitutes the task requirement. The possible quality levels correspond to the performance levels shown in Table 2. Initially, the consumer requests service at the highest quality level (PL_PERFECT). If no provider successfully fulfills the request within a predefined waiting period WT, the consumer broadcastas a new request with a lower quality requirement (PL_GOOD). This process continues progressively through lower performance levels until either the service is delivered or the minimum acceptable quality level is reached. Upon receiving a request message, providers store it locally and execute the CA algorithm described in the previous chapter. The parameter WT represents the maximum time allowed for all service requests generated during a simulation round to be processed.

To reproduce the characteristics of open MAS, the testbed incorporates continuous population changes. After each round, randomly selected agents may leave the system and be replaced by new agents. The number of additions and removals is bounded by predefined population-change limits. The parameters p_{CPC} and p_{PPC} denote the maximum rates of consumer and provider population replacement, respectively. Newly introduced agents receive randomly assigned characteristics, while preserving the overall proportions of provider profiles and consumer categories.

Agent mobility is also explicitly modeled. Changing an agent's location changes its relationships with others, as well as its individual situation. Specifically, polar coordinates (r, φ, θ) are used for agents' location in the spherical world. In order to change an agent's location, amounts of angular changes $\Delta\varphi$ and $\Delta\theta$ are added to φ and θ, respectively. In this case, $\Delta\varphi$ and $\Delta\theta$ are selected randomly in $[-\Delta\phi, +\Delta\phi]$. Thus, maximum angular distance $\Delta\phi$ limits the variability of agents' locations. Not every agent changes its location every round and in particular, p_{CLC} and p_{PLC} are used to denote the probabilities that a given consumer or provider, respectively, changes its location in a

round. Mobility continuously alters agents' neighborhoods and interaction opportunities, contributing to the dynamic nature of the environment.

Finally, the testbed models changes in trustee behavior over time. The average performance of a provider may vary by an amount Δ_{μ}, randomly selected from the interval $[-M, +M]$, with probability $p_{\mu C}$ at each round. Furthermore, providers may completely switch to a different performance profile with probability $p_{ProfileSwitch}$. These mechanisms allow the simulation of both gradual performance fluctuations and abrupt behavioral changes, creating challenging conditions for trust assessment and adaptation.

Collectively, population replacement, agent mobility, and evolving provider performance create a highly dynamic experimental environment, enabling a systematic comparison of the robustness and adaptability of trustor-centric and trustee-centric trust models under realistic open-system conditions.

3.4. Experimental Methodology

The objective of the experimental study is to compare the effectiveness and robustness of the proposed trustee-centric CA model against a representative trustor-centric trust model under a variety of dynamic environmental conditions. To provide a baseline for comparison, a third group of agents that does not employ any trust mechanism is also included.

Specifically, the experiments compare the performance of the following three consumer groups:

- **NoTrustModel**: consumer agents do not use any trust mechanism and select a nearby provider randomly.
- **FIRE**: consumer agents use the FIRE trust and reputation model to evaluate and select providers.
- **CA:** consumer agents use the proposed *Create Assemblies* trust model.

The inclusion of the NoTrustModel group allows the assessment of the overall benefits provided by trust-based decision-making, while the comparison between FIRE and CA enables the evaluation of the trustee-centric and trustor-centric paradigms under identical environmental conditions.

3.4.1. Performance Measure

The primary performance metric used throughout the experiments is the Utility Gain (UG) obtained by consumers after each service interaction. As discussed in the previous section, the utility gain reflects the quality of the service received and takes values in the interval $[-10, +10]$.

A trust model that successfully identifies trustworthy providers is expected to guide consumers towards interactions that yield higher utility gains. Consequently, higher average utility gain indicates better trust assessment performance.

During each simulation run, the testbed records:

- The utility gain produced by every interaction,
- The trust model used by the consumer, and
- The corresponding simulation round.

After the completion of all simulation runs, the average utility gain is calculated for each consumer group at every interaction.

3.4.2. Statistical Analysis

To reduce the influence of stochastic effects and random fluctuations, each experiment is repeated multiple times through a **Number of Independent Simulation Runs (NISR)**. The number of simulation runs varies depending on the experiment, as shown in Table 3, and is selected so as to achieve statistically reliable results.

Table 3. Number of Independent Simulation Runs (NISR)

Experiment	NISR
1, 2, 6, 8, 10, 11	30
3, 4, 5, 7	10
9	12

Following the methodology adopted in the original FIRE experimental framework, the performance of the consumer groups is compared using a two-sample t-test for means comparison [144] with a confidence level of 95%.

The comparison produces performance rankings among the evaluated groups. When two groups exhibit statistically indistinguishable performance, they are assigned the

same rank. Otherwise, the group with the higher average utility gain receives the higher ranking.

3.4.3. Presentation of Results

The results of each experiment are presented using two-axis graphs. The left vertical axis depicts the average utility gain achieved by each consumer group over time, while the right vertical axis shows the corresponding performance rankings obtained from the statistical comparison. We use the group's name prefixed by 'R' to denote the plot for the performance ranking of a group. Higher ranking values indicate superior performance, whereas equal ranking values indicate that the observed differences are not statistically significant at the selected confidence level.

3.4.4. Experimental Parameters

Unless otherwise stated, all experiments use the same baseline configuration as the original FIRE testbed. The provider population corresponds to a typical provider population, consisting of approximately equal proportions of beneficial and harmful providers. Beneficial providers generate positive utility gains, whereas harmful providers generate negative utility gains, including intermittent providers whose behavior fluctuates over time.

The main simulation parameters are summarized in Table 4.

Table 4. General Simulation Parameters

Simulation variable	Symbol	Value
Number of simulation rounds (Default)	N	500
Number of simulation rounds (Experiments 7, 9)	N	1000
Total providers	N_P	100
Good providers	N_{GP}	10
Ordinary providers	N_{PO}	40
Intermittent providers	N_{PI}	5
Bad providers	N_{PB}	45
Total consumers	N_C	500
Consumer activity level	a	$[0.25, 1.00]$

Waiting time	WT	1000 ms

The default parameter values used for FIRE are shown in Table 5, while the default parameters of the proposed CA model are presented in Table 6.

Table 5. Default FIRE parameters

Parameters	Symbol	Value
Local rating history size	H	10
IT recency scaling factor	λ	$-(5/\ln(0.5))$
Branching factor	n_{BF}	2
Referral length threshold	n_{RL}	5
Component coefficients		
• Interaction trust	W_I	2.0
• Role-based trust	W_R	2.0
• Witness reputation	W_W	1.0
• Certified reputation	W_C	0.5
Reliability function parameters		
• Interaction trust	γ_I	$-\ln(0.5)$
• Role-based trust	γ_R	$-\ln(0.5)$
• Witness reputation	γ_W	$-\ln(0.5)$
• Certified reputation	γ_C	$-\ln(0.5)$

Table 6. Default CA parameters

Parameters	Symbol	Value
Trust threshold	Threshold	0.5
Strengthening factor	α	0.1
Weakening factor	β	0.1

The corresponding FIRE parameters are reported for completeness and to ensure the reproducibility of the experimental results. The values used are identical to those employed in the original FIRE evaluation study.

3.5. Experimental Scenarios and Results

3.5.1. Baseline Comparison in a Static Environment

The first experiment serves as a baseline comparison between the proposed CA model, the FIRE trust model, and the NoTrustModel consumer group under static environmental conditions. No dynamic factors are active during this experiment; therefore, agent populations, locations, and provider performance profiles remain unchanged throughout the simulation.

The objective of this experiment is to investigate whether CA enables consumers to achieve higher utility gains than both FIRE and random provider selection. The results are presented in Figure 3. As expected, the NoTrustModel group performs the worst in all interactions. CA and FIRE, on the other hand, both help consumers in gaining far greater utility gain (UG), demonstrating that both models are capable of learning how to pick profitable provider agents.

The chart, as well as the t-test ranking, show that FIRE can learn about the providers faster than CA in the first 38 interactions, since it achieves a greater UG than CA. Yet, after interaction 38, CA consistently outperforms FIRE with a rather stable performance. On the contrary, FIRE's performance is steadily improving, tending to surpass CA's performance after interaction 480. Nevertheless, additional experiments with a larger number of simulation rounds would be required to determine whether FIRE eventually surpasses CA.

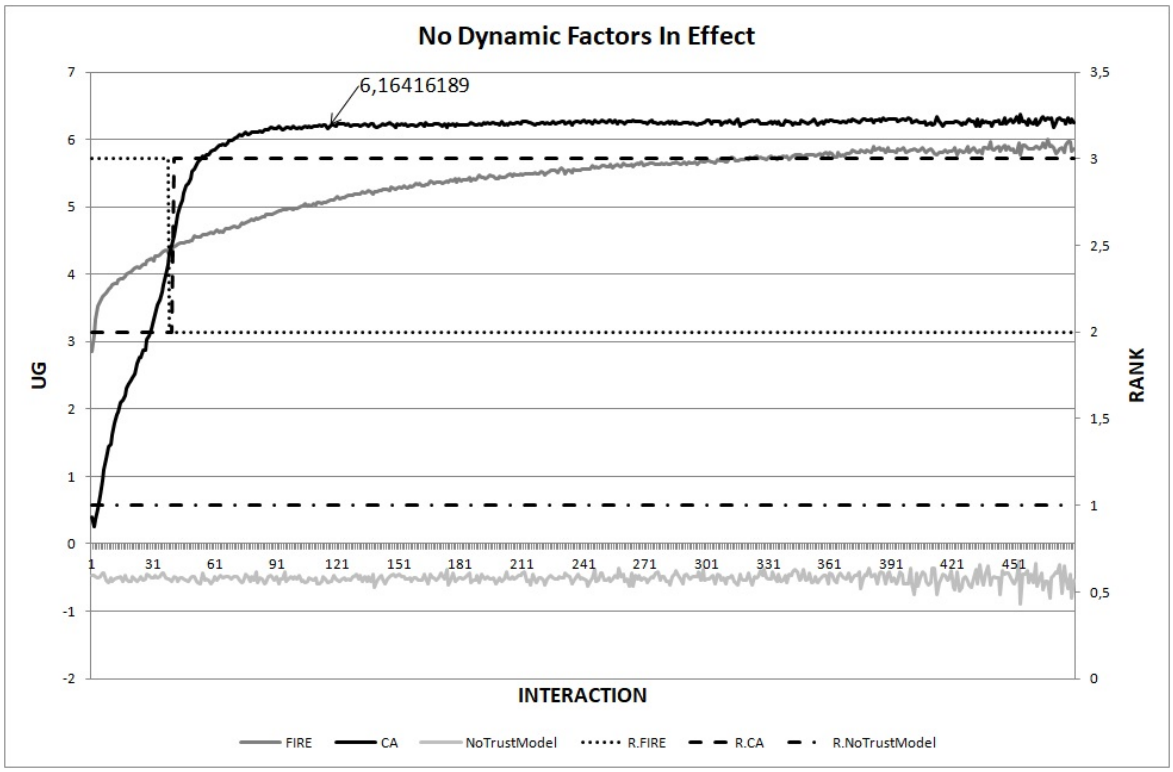


Figure 3. Experiment 1: CA versus FIRE performance in a static environment.

3.5.2. CA's vs. FIRE's Performance under Provider Population Changes

The second set of experiments examines the impact of provider population change on the performance of the CA and FIRE trust models. In an open Multi-Agent System, provider agents may continuously enter and leave the environment, requiring trust models to adapt to the presence of previously unknown service providers. To investigate this effect, three experiments were conducted:

- **Experiment 2**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$).
- **Experiment 3**: The provider population changes at maximum 5% in every round ($p_{PPC} = 0.05$).
- **Experiment 4**: The provider population changes at maximum 10% in every round ($p_{PPC} = 0.10$).

The results of Experiments 2-4 are presented in figures 4-6. In all cases, both models continue to help the consumers in obtaining positive utility gain (UG), whereas the NoTrustModel group consistently exhibits the lowest performance. However, provider

population changes negatively affect both trust models compared to the static setting presented in Experiment 1 (Figure 3).

More specifically, in Experiment 2 (Figure 4), both the chart and the t-test ranking show that FIRE is superior in the first 41 interactions, while CA outperforms FIRE with a stable performance after interaction 41. FIRE, on the other hand, is constantly improving to reach CA's performance.

Experiment 3 (Figure 5) and Experiment 4 (Figure 6) show that when the provider population changes more dramatically, CA performs worse than FIRE. Figures 7 and 8 show more clearly that FIRE is more resilient to this type of environmental change. As the provider population change rate increases from 2% to 10%, FIRE's performance drops by only one UG unit, while CA's performance drops by four.

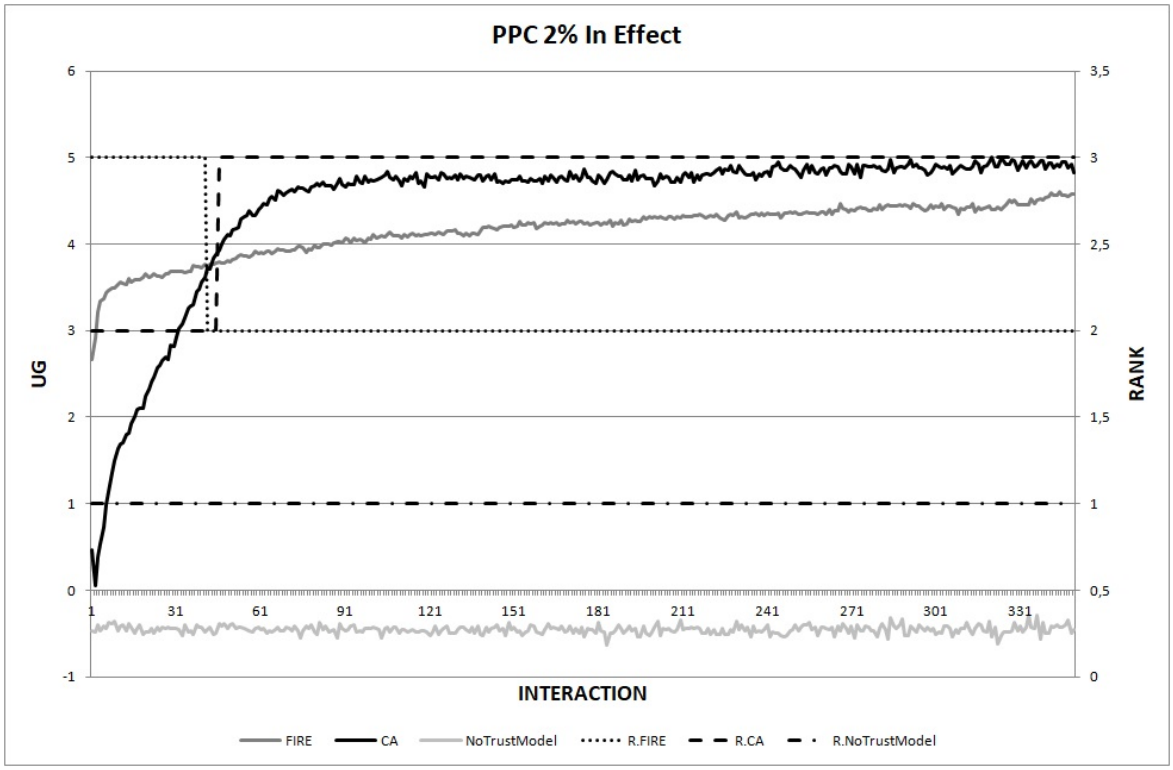


Figure 4, Experiment 2. Provider population change $(\boldsymbol{p_{ppc}} = \mathbf{2\%})$

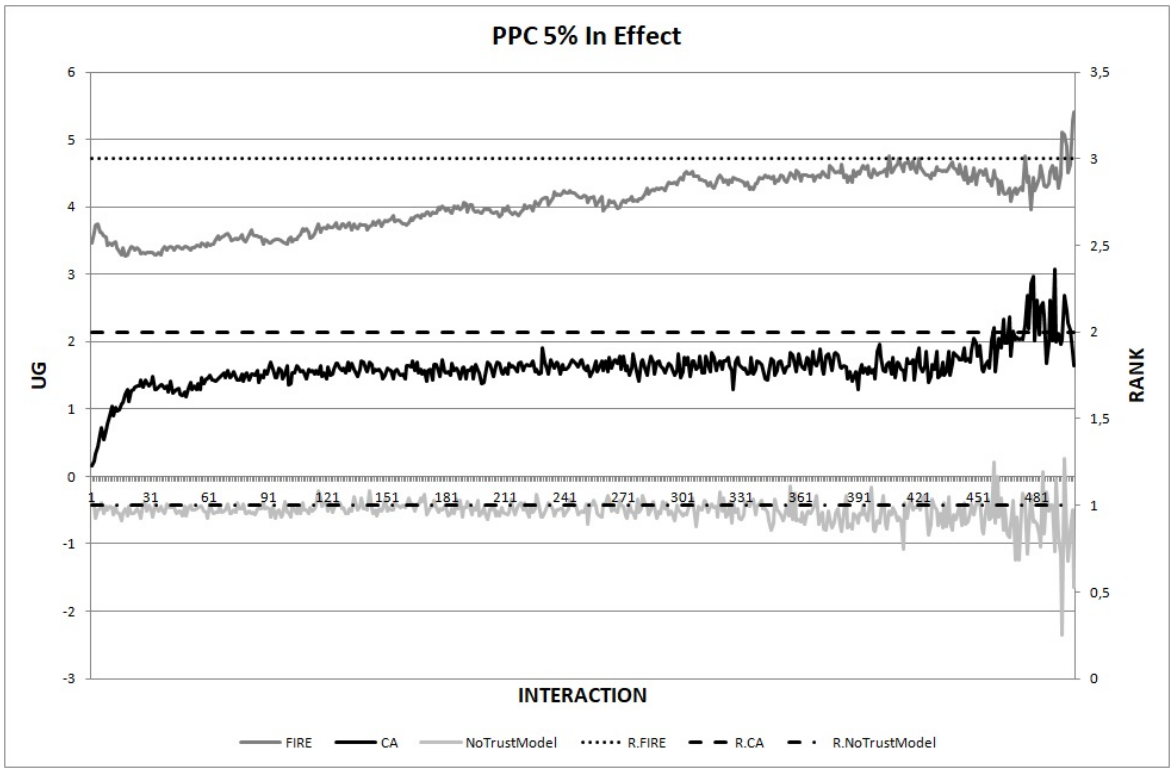


Figure 5. Experiment 3. Provider population change $(\boldsymbol{p_{ppc}} = \mathbf{5}\%)$

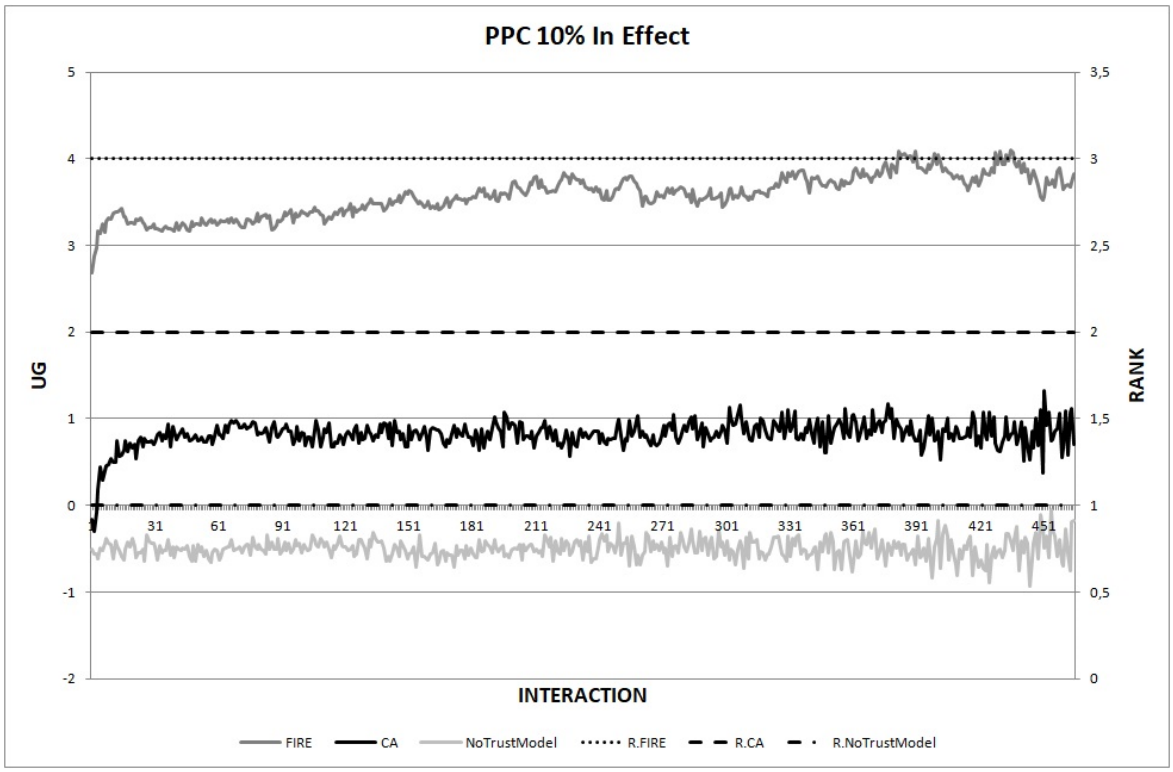


Figure 6. Experiment 4. Provider population change $(\boldsymbol{p_{ppc}} = \mathbf{10}\%)$

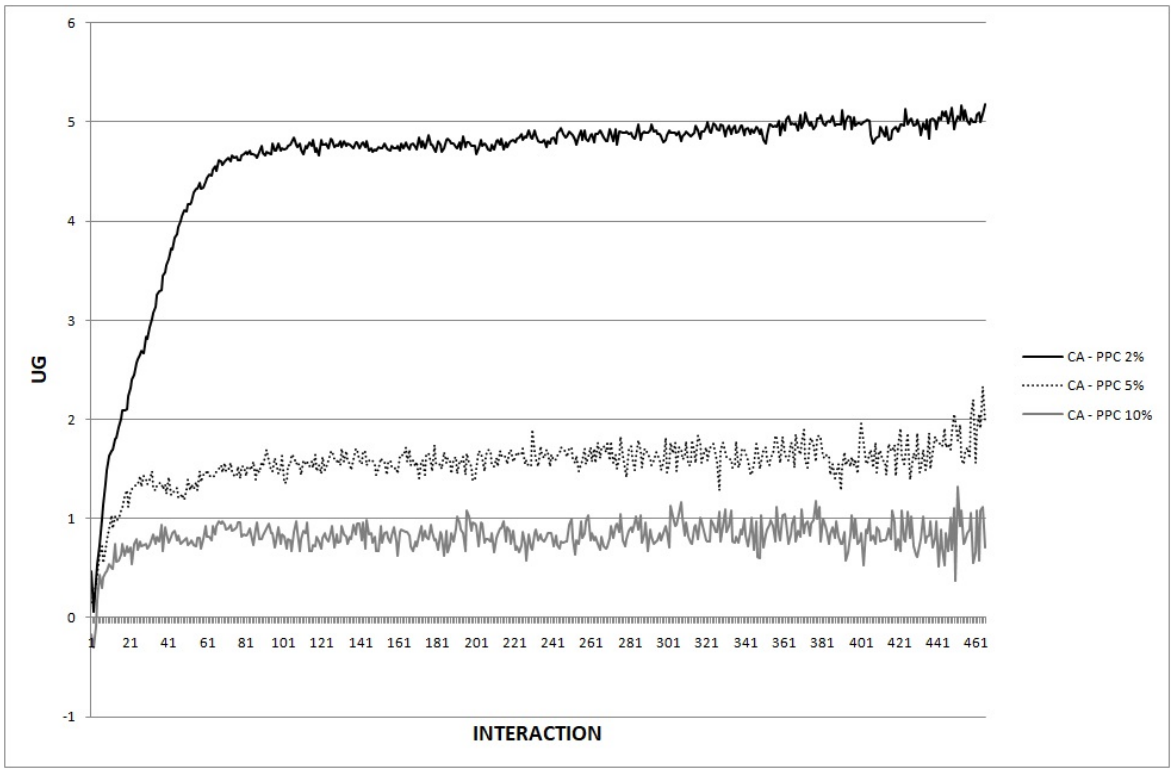


Figure 7. Impact of increasing provider population change on CA performance

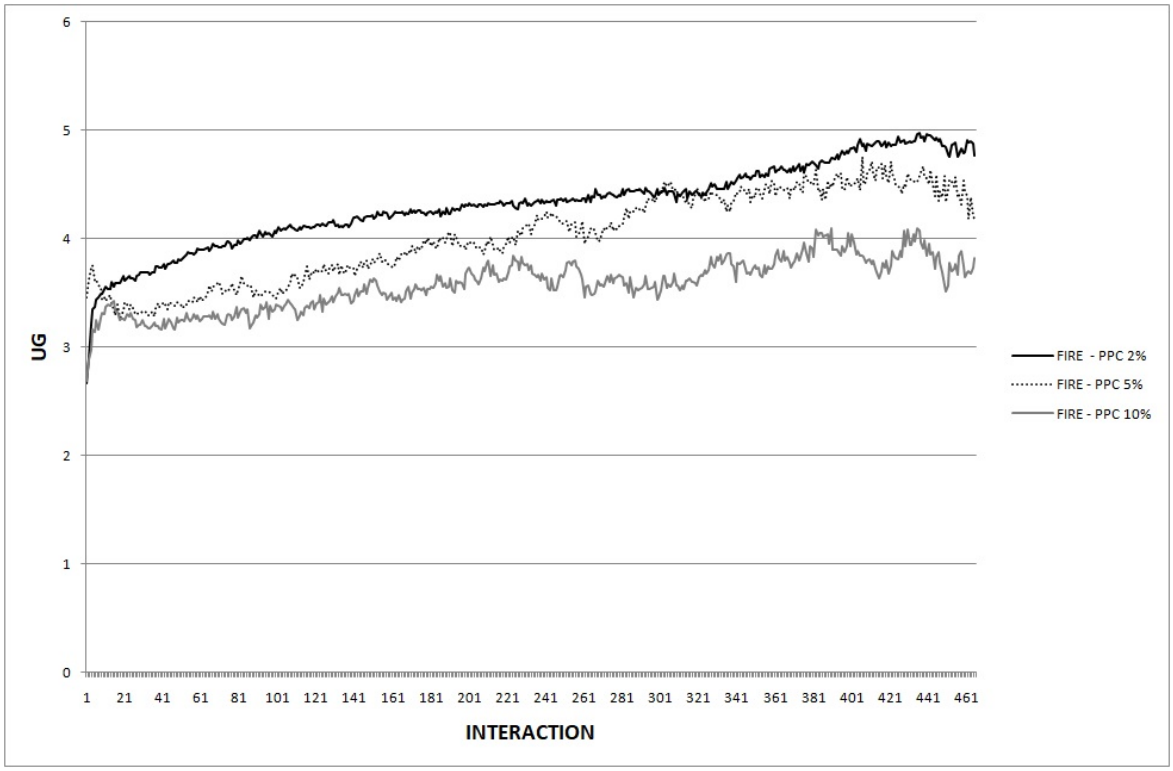


Figure 8. Impact of increasing provider population change on FIRE performance

3.5.3. CA's vs. FIRE's Performance under Consumer Population Changes

The third set of experiments investigates the effect of consumer population change on the performance of the CA and FIRE trust models. In open Multi-Agent Systems, consumer agents may continuously enter and leave the environment, affecting the availability of trust-related information and the ability of trust models to identify trustworthy providers. To evaluate the impact of this type of environmental change, three experiments were conducted:

- **Experiment 5**: The consumer population changes at maximum 2% in every round ($p_{CPC} = 0.02$).
- **Experiment 6**: The consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$).
- **Experiment 7**: The consumer population changes at maximum 10% in every round ($p_{CPC} = 0.10$).

The results of Experiments 5-7 are presented in Figures 9-11. In all experiments, both CA and FIRE continue to achieve positive utility gain (UG), while the NoTrustModel group consistently exhibits the lowest performance.

More specifically, in Experiment 5 (Figure 9), CA outperforms FIRE from the first interaction with a rather steady performance at the levels of its performance in the static environment. FIRE on the other hand performs worse than in the static setting. As the consumer population change rate increases gradually from 2% to 10%, CA consistently outperforms FIRE. Figure 12 shows that CA is not only resilient to this kind of change, but also that its performance improves with the increase of consumer population change rate.

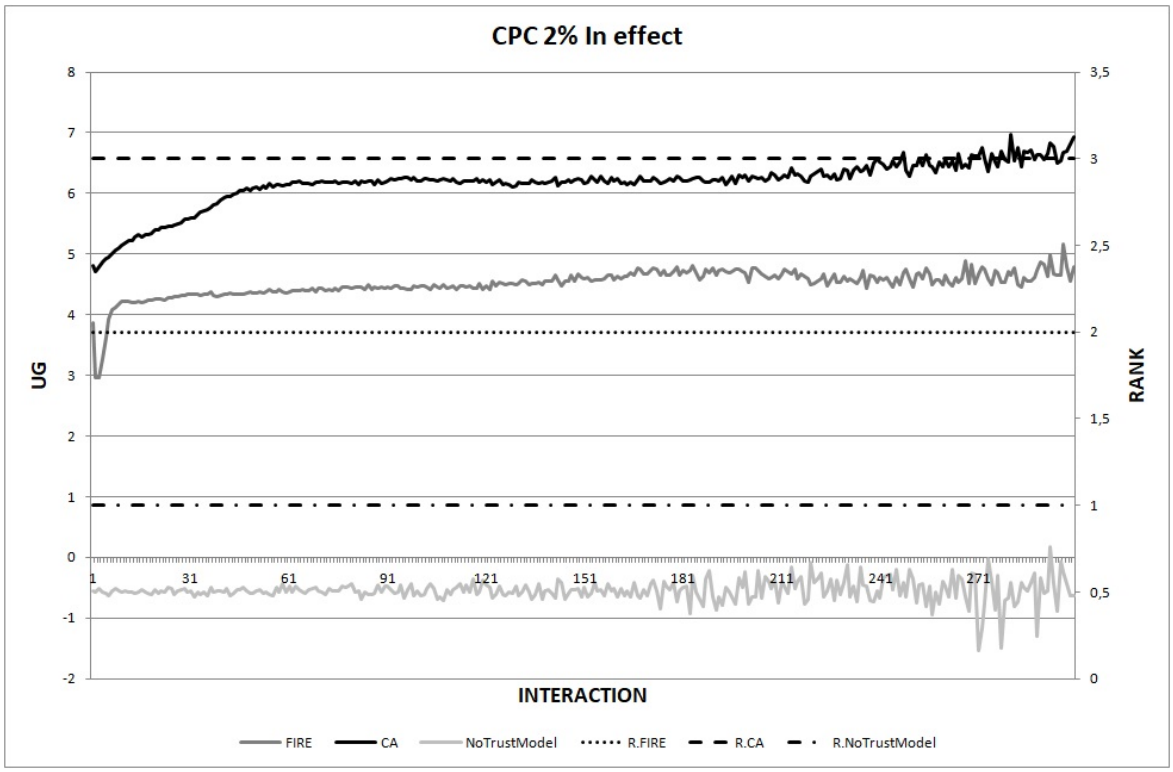


Figure 9. Experiment 5. Consumer population change ($\boldsymbol{p_{CPC} = 2\%}$)

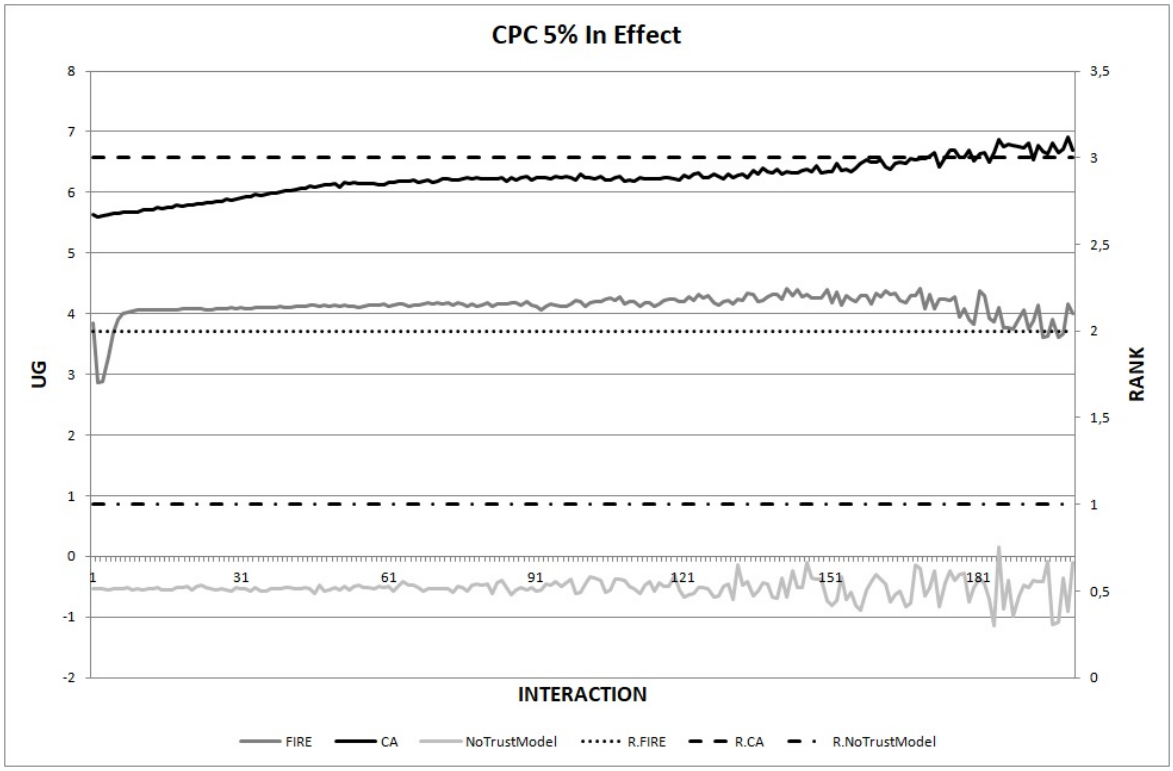


Figure 10. Experiment 6. Consumer population change ($\boldsymbol{p_{CPC} = 5\%}$)

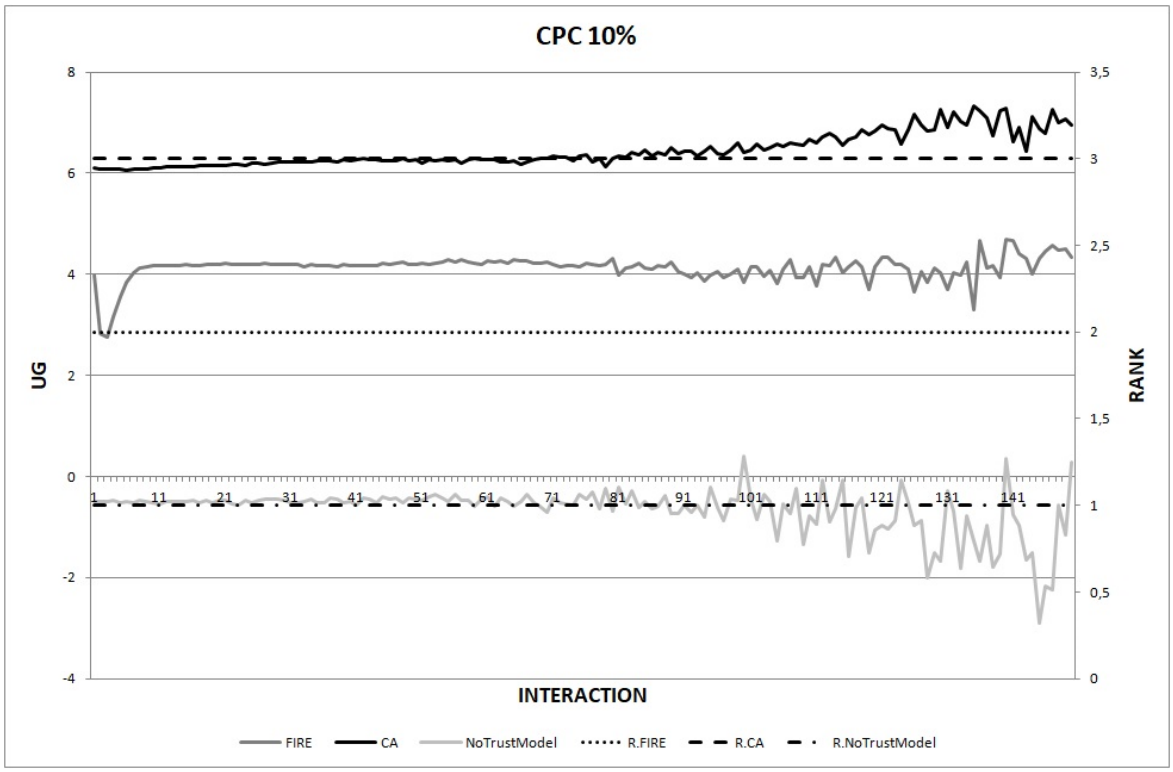


Figure 11. Experiment 7. Consumer population change ($\boldsymbol{p_{CPC} = 10\%}$)

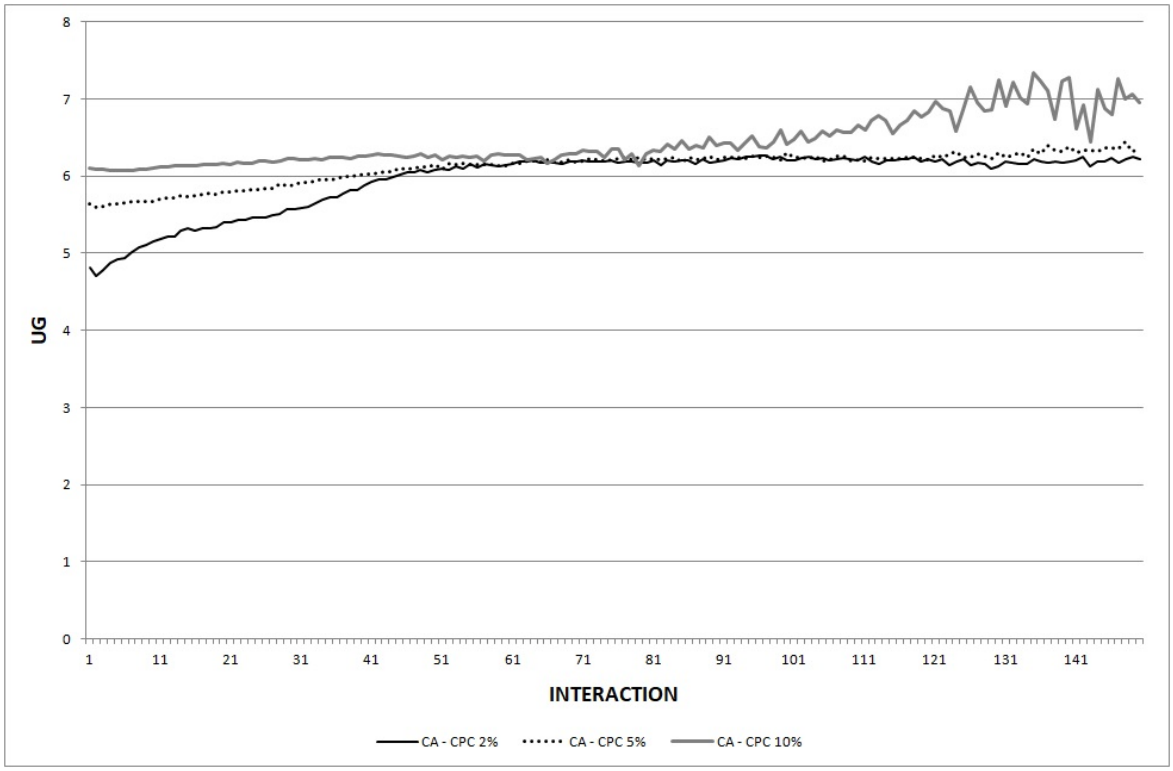


Figure 12. CA performance under increasing consumer population change rates

3.5.4. The Effect of Concurrent Consumer and Provider Population Changes

The fourth set of experiments examines the performance of the CA and FIRE trust models when both consumer and provider populations change simultaneously. This scenario combines the two forms of population dynamics examined in the previous experiments and represents a more challenging environment. Two experiments were conducted:

- **Experiment 8**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$) and the consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$).
- **Experiment 9**: The provider population changes at maximum 10% in every round ($p_{PPC} = 0.10$) and the consumer population changes at maximum 10% in every round ($p_{CPC} = 0.10$).

Experiment 9 (Figure 14) shows that while CA is better in the first few interactions, FIRE eventually prevails. This is in contrast to the results of Experiment 8 (Figure 13), in which CA outperforms FIRE throughout all interactions. As the population change becomes more severe, FIRE's performance is slightly affected, whereas CA's performance drops significantly, by three UG units.

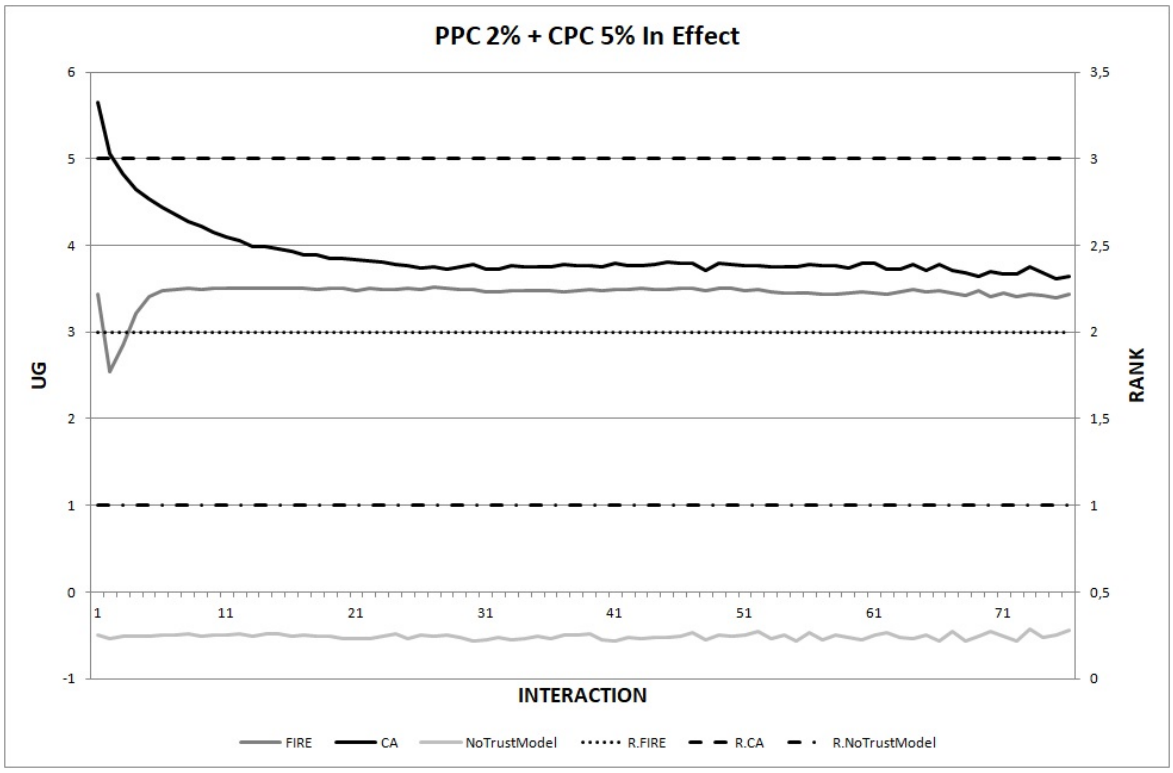


Figure 13. Experiment 8. Concurrent population changes ($p_{PPC} = 2\%, p_{CPC} = 5\%$)

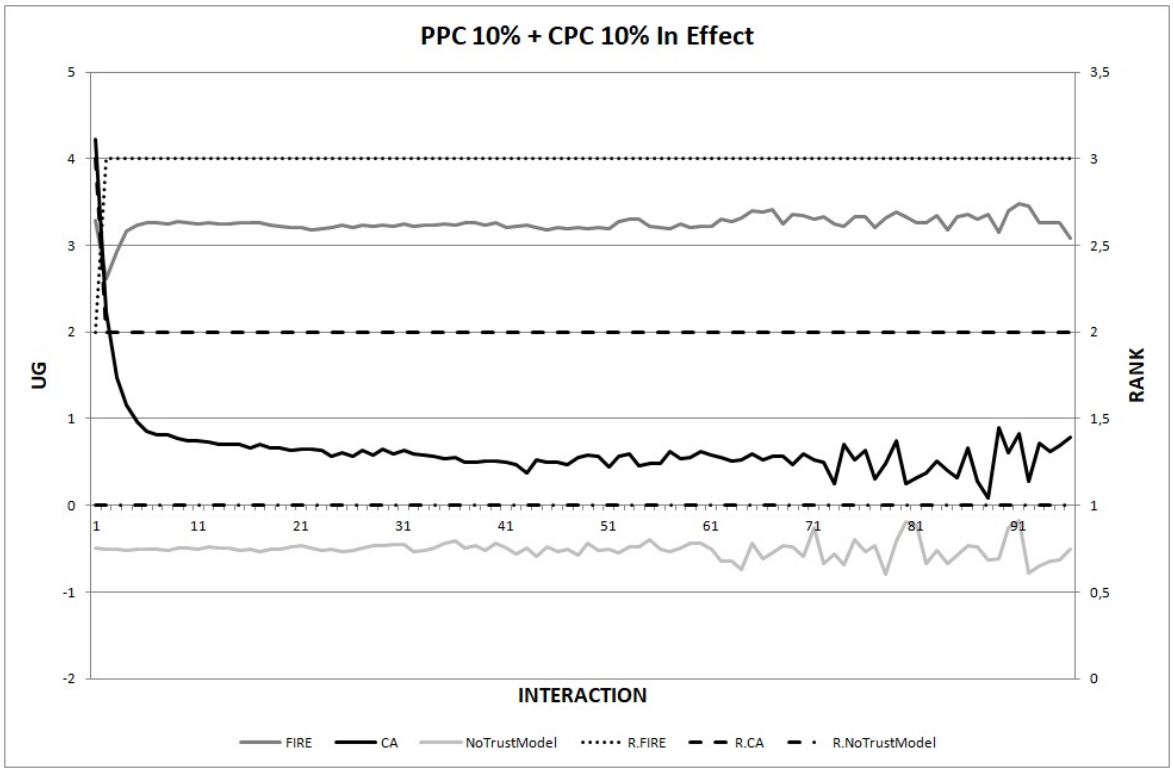


Figure 14. Experiment 9. Concurrent population changes ($\boldsymbol{p_{PPC} = 10\%, p_{CPC} = 10\%}$)

3.5.5. CA's vs. FIRE's performance in dynamic trustee profiles

The final set of experiments investigates the performance of the CA and FIRE trust models when providers continuously modify their service capabilities over time. Two experiments were conducted:

- **Experiment 10**: A provider may alter its average level of performance at maximum 1.0 UG unit with a probability of 10% in every round $(p_{\mu C} = 0.10, M = 1.0)$.
- **Experiment 11**: A provider may switch into a different (performance) profile with a probability of 2% in every round $(p_{ProfileSwitch} = 0.02)$.

In Experiment 10 (Figure 15), providers continuously modify their performance levels. CA starts with an average utility gain close to zero and rapidly improves until approximately interaction 60, after which its performance stabilizes at an average utility gain of about 6.6. In contrast, FIRE begins with a higher utility gain and gradually improves over time, eventually surpassing CA after approximately interaction 285.

Experiment 11 (Figure 16) has the strongest negative impact on the performance of both trust models. In this setting, providers may switch to entirely different performance

profiles during the simulation. FIRE outperforms CA through all interactions, achieving an average utility gain of 3.15, whereas CA achieves an average utility gain of 2.64.

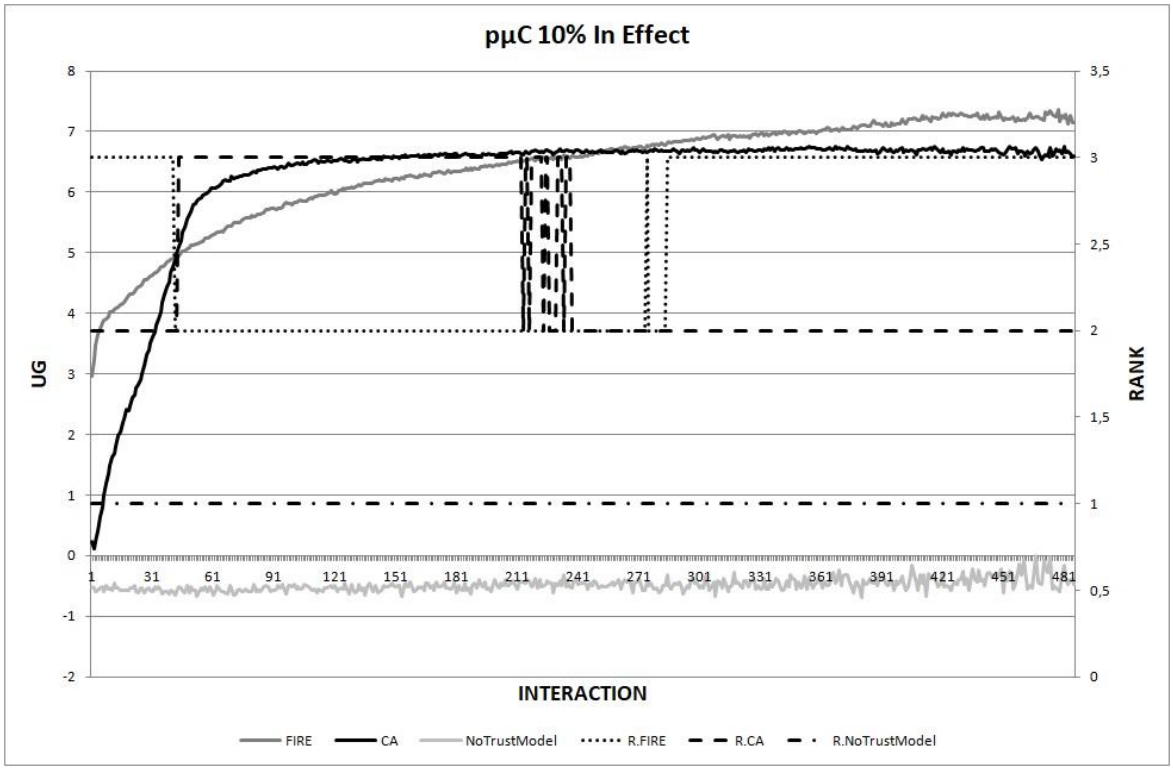


Figure 15. Experiment 10. Providers change their performance $\left(\boldsymbol{p_{\mu C}} = \mathbf{10\%}, \boldsymbol{M} = \mathbf{1.0}\right)$

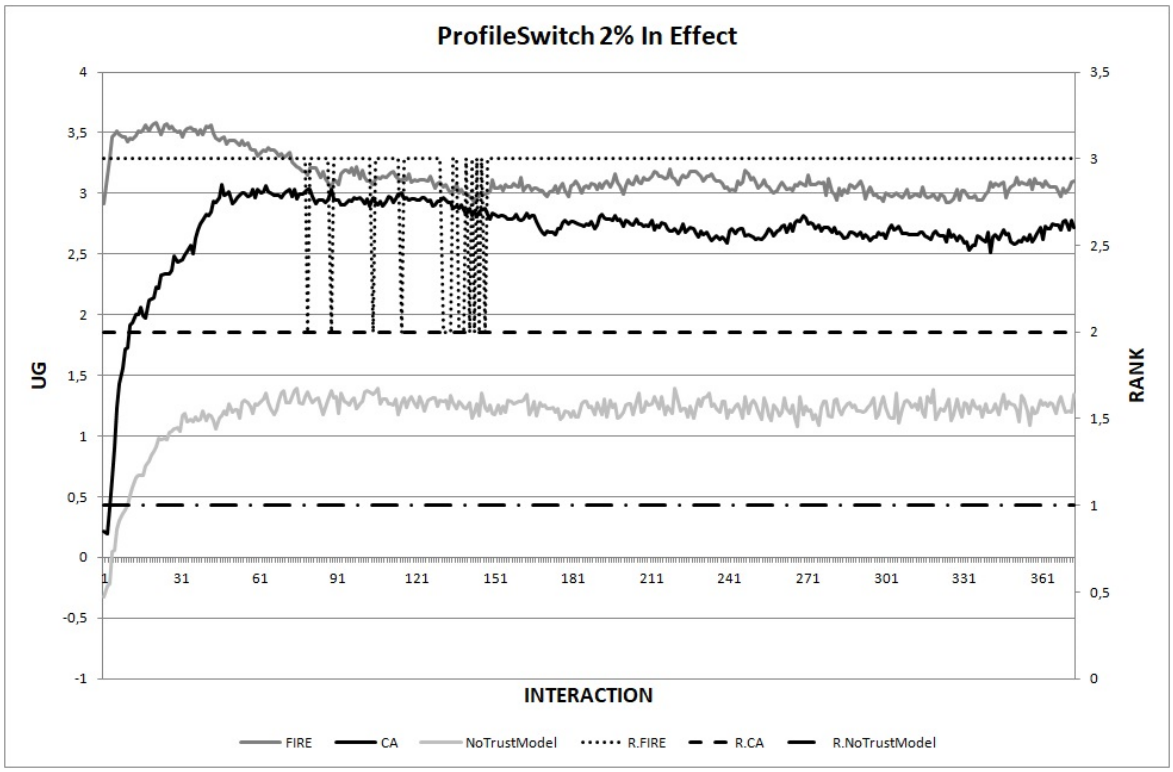


Figure 16. Experiment 11. Providers switch performance profiles $(\boldsymbol{p_{ProfileSwitch}} = \mathbf{2\%})$

3.6. Explaining the Performance Differences Between CA and FIRE

The experimental results presented in the previous section reveal that the relative performance of CA and FIRE depends on the type of environmental dynamics affecting the Multi-Agent System. While both models consistently outperform the NoTrustModel group and are capable of identifying profitable providers, they exhibit different strengths under different conditions.

The results indicate that CA has a clear advantage in environments characterized by changes in the consumer population. FIRE relies on multiple sources of trust information, including Witness Reputation, which estimates a provider's trustworthiness based on the opinions of other consumers that have previously interacted with it. When consumers leave the system, the information they possess becomes unavailable, reducing the effectiveness of this trust source. In contrast, CA does not depend on recommendations or evaluations provided by other consumers. Consequently, its performance remains largely unaffected by consumer population change, even when the rate of change is high.

On the other hand, FIRE demonstrates greater resilience to changes in the provider population. The CA model is based on trustees' knowledge of their own capabilities,

represented though the trust connections they maintain and update over time. As providers accumulate interaction experience, these trust values become increasingly representative of their actual ability to perform tasks successfully. However, when providers are frequently replaced, newly introduced trustees possess no prior knowledge regarding their capabilities and must begin the learning process from scratch. As a result, the performance of CA deteriorates as provider population change increases.

The experiments also show that changes in trustees' performance characteristics constitute a significant challenge for both trust models. In particular, the most demanding scenario is when providers switch entirely to different performance profiles. Under these conditions, both models experience a substantial reduction in performance, although FIRE adapts more effectively than CA.

Overall, the findings suggest that the trustee-centric approach introduced by CA offers important advantages in environments where the trustee population remains relatively stable and consumer population change is high. Conversely, when the provider population changes rapidly, the trustor-centric approach represented by FIRE appears to be more robust. These observations indicate that neither approach dominates across all environmental conditions, highlighting the importance of considering the characteristics of the application domain when selecting an appropriate trust model.

3.7. Conclusions

This chapter presented a comparative evaluation of the proposed *Create Assemblies* (CA) trust model and the FIRE trust and reputation model under a variety of static and dynamic environmental conditions. The objective was to investigate the strengths and limitations of the trustee-centric and trustor-centric trust paradigms in open Multi-Agent Systems.

The experimental results showed that CA is capable of maintaining a stable performance both in static environments and in environments affected by several forms of dynamism. In all examined scenarios, CA successfully identified profitable providers and maintained positive utility gains for the consumer population. The results further demonstrated that the trustee-centric approach can constitute a viable alternative to conventional trustor-centric trust models.

The comparison with FIRE revealed that each model exhibits different strengths under different environmental conditions. FIRE proved to be more resilient to provider population changes, whereas CA showed remarkable resilience to consumer population change. This behavior can be attributed to the fact that CA does not rely on witness opinions and reputation information exchanged among consumers. Consequently, changes in the consumer population have a limited impact on its operation.

The experiments also showed that dynamic changes in trustees' capabilities constitute a challenging condition for both trust models. In particular, provider profile switching had the most significant negative effect on performance. Although both models were affected, FIRE demonstrated greater resilience under this scenario.

Overall, the findings indicate that neither the trustee-centric nor the trustor-centric approach dominates across all dynamic conditions. Instead, the effectiveness of each approach depends on the specific characteristics of the environment in which it operates.

3.8. Research Directions

The findings of this study highlighted two important research directions. The first concerns the enhancement of the CA algorithm to better cope with dynamic changes in trustees' capabilities. The experimental results showed that abrupt profile switching has a significant impact on the model's performance, indicating the need for mechanisms that enable trustees to adapt more rapidly when their actual capabilities change.

The second direction concerns the adaptive selection of trust models according to the characteristics of the environment. The experimental comparison demonstrated that neither CA nor FIRE consistently outperforms the other under all dynamic conditions. Instead, each model exhibits advantages in different situations. This observation raises an important question regarding whether agents can identify the dynamic conditions present in their environment and exploit this knowledge to select the trust model that is expected to maximize utility gain.

Both research directions were subsequently investigated as part of this doctoral research and are presented in the following chapters. The next chapter focuses on the second direction, namely the development of mechanisms that enable agents to identify environmental conditions and adaptively employ the trust model that is most appropriate for the current context. The first research direction, concerning improvements to the CA

algorithm for handling dynamic changes in trustees' capabilities, is examined in Chapter 5.

4. Adaptive Trust Model Selection in Dynamic Multi-Agent Systems

The experimental results presented in the previous chapter demonstrated that the effectiveness of a trust model depends strongly on the characteristics of the environment in which it operates. While the trustee-centric CA model exhibited superior performance under conditions of consumer population change, the trustor-centric FIRE model proved more resilient under conditions of provider population change. Consequently, neither model consistently outperformed the other across all dynamic conditions.

These findings raise an important question: rather than relying permanently on a single trust mechanism, can agents identify the characteristics of their environment and dynamically select the trust model that is most appropriate for the current conditions?

This chapter investigates this question by formulating trust model selection as a machine learning problem in a dynamic and partially observable Multi-Agent System. In such environments, agents do not possess global knowledge regarding the factors affecting the system and must instead infer environmental conditions from information available through their own interactions. The objective is therefore to enable agents to learn which trust model is expected to maximize utility gain under varying environmental conditions.

To achieve this, we introduce an adaptive trustor capable of dynamically selecting between the CA and FIRE trust models. The problem is formulated as a reinforcement learning task, where the agent learns through experience which trust mechanism yields the highest utility gain in each observed situation. More specifically, we employ Deep Q-Learning (DQL) to approximate the optimal trust model selection policy based on a set of measurable environmental features.

The chapter, first, presents the conceptualization of trust model selection as a reinforcement learning problem and describes the architecture of the proposed adaptive trustor. It then introduces the environmental features used to characterize the state of the system, the Deep Q-Network configuration, and the simulation environment employed for evaluation. Finally, a series of experiments compares the performance of the adaptive trustor against agents that rely exclusively on either CA or FIRE, assessing whether

dynamic trust model selection can improve performance in open and continuously changing environments.

4.1. Adaptive Trust Model Selection as a Reinforcement Learning Problem

The comparative evaluation presented in the previous chapter showed that neither FIRE nor CA constitutes a universally optimal trust mechanism. Each model performs better under specific environmental conditions, while its effectiveness may deteriorate when those conditions change. Consequently, selecting the most suitable trust model becomes a decision-making problem that depends on the current state of the environment.

In open MAS, however, agents typically operate under incomplete information. Important environmental factors, such as population turnover, changes in agent behavior, mobility, and performance fluctuations, are not directly observable. Instead, agents must infer the state of the environment from locally available information obtained through interactions with nearby agents and accumulated trust evidence. Therefore, trust model selection can be viewed as a sequential decision-making problem under uncertainty in partially observable environments.

To address this challenge, the adaptive trustor is modeled as a reinforcement learning agent. At each decision point, the agent observes a representation of its local environment and selects one of two possible actions:

- **Pull strategy**: use the FIRE trust model, where the trustor evaluates and selects the most trustworthy provider.
- **Push strategy**: use the CA trust model, where providers self-assess their capability and decide whether to undertake the requested task.

Following the interaction, the agent receives a utility gain reflecting the quality of the obtained service. This utility serves as a reward signal, allowing the agent to evaluate the effectiveness of its decision. Over time, the agent learns which trust mechanism tends to maximize utility under different environmental conditions.

The problem can therefore be formulated within the Reinforcement Learning (RL) framework. The adaptive trustor interacts with a dynamic environment, observes its current state, selects an action corresponding to a trust model, and receives a reward based

on the outcome of the interaction. The objective is to learn a policy that maximizes the expected cumulative utility over time.

Since the environment is only partially observable and its underlying dynamics are unknown, traditional rule-based approaches are difficult to design effectively. Reinforcement Learning offers a suitable alternative because it enables the agent to learn directly from experience without requiring an explicit model of environmental behavior. In this work, Deep Q-Learning is employed to approximate the optimal policy for trust model selection using a set of observable environmental features, which are described in the following sections. A concise overview of Reinforcement Learning, Q-Learning, and Deep Q-Learning is provided in Appendix B, while the remainder of this chapter focuses on the formulation, implementation, and evaluation of the proposed adaptive trust model.

4.2. Deep Q-Network Architecture

To learn an effective trust model selection policy, the adaptive trustor employs a Deep Q-Network (DQN), which approximates the action-value function of the reinforcement learning problem. The purpose of the network is to estimate the expected future utility associated with each available trust model and enable the agent to select the most beneficial action under the current environmental conditions.

The DQN receives as input a representation of the agent's local environment, expressed through a set of observable features. Based on these inputs, the network produces estimates of the expected cumulative reward for each available action. In the context of trust model selection, the action space consists of two possible choices:

- **Action 1**: Select the FIRE trust model (pull strategy).
- **Action 2**: Select the CA trust model (push strategy).

The architecture of the neural network is intentionally simple to reduce computational complexity while maintaining sufficient representational capability. The network consists of three fully connected layers.

The input layer contains nine neurons, corresponding to the nine environmental features used to characterize the current state of the system. These features provide indirect information regarding environmental dynamics, such as population changes, mobility, and variations in provider performance.

The hidden layer consists of six neurons and uses a sigmoid activation function. This layer enables the network to learn nonlinear relationships between environmental conditions and the expected effectiveness of each trust mechanism.

The output layer contains two neurons, one for each available action. A linear activation function is used at the output layer since the network is required to estimate Q-values rather than probabilities. Each output neuron therefore represents the estimated long-term utility associated with selecting one of the two trust models.

During operation, the adaptive trustor evaluates the current environmental state and feeds the corresponding feature vector into the network. The DQN then produces two Q-values, one for FIRE and one for CA. The agent selects the action associated with the highest estimated value, while occasional exploratory actions are allowed through an $\varepsilon - greedy$ exploration strategy. Following the interaction, the obtained utility gain is used as a reward signal to update the network and improve future decision making.

The use of a Deep Q-Network allows the adaptive trustor to generalize across a large number of possible environmental conditions. Instead of relying on predefined rules, the agent gradually learns the relationship between observable environmental characteristics and the trust mechanism that is most likely to maximize utility. This capability is particularly important in open and dynamic Multi-Agent Systems, where environmental conditions may continuously evolve over time.

4.3. State Representation and Environmental Features

A critical component of any reinforcement learning system is the representation of the environment's state. The quality of the information provided to the learning agent directly affects its ability to identify patterns, distinguish between different situations, and ultimately learn an effective decision-making policy.

In the trust model selection problem considered in this work, the adaptive trustor operates in a partially observable environment. The agent does not have direct access to the actual environmental factors that influence the performance of trust mechanisms, such as provider population turnover, consumer population changes, agent mobility, or fluctuations in provider behavior. Consequently, these factors cannot be used directly as state variables. Instead, the adaptive trustor must rely on information that is locally available through its interactions and observations. The objective is therefore to construct

a set of measurable features that can serve as indirect indicators of the underlying environmental conditions. Although these features do not reveal the true state of the environment, they provide sufficient information to allow the learning agent to infer important changes and adapt its trust model selection strategy accordingly.

4.3.1. Available Local Information in a Partially Observable Environment

In the proposed framework, trust model selection takes place in a partially observable environment. The adaptive trustor does not possess direct knowledge of the underlying environmental factors that influence trust model performance, such as population changes, provider behavior modifications, or agent mobility. Instead, it must infer the current state of the environment from information that is locally available through its own observations and interactions.

More specifically, the adaptive trustor has access to the following sources of information:

1. **Local ratings database**. After each interaction, the trustor evaluates the received service and stores the resulting rating in its local ratings database. These ratings constitute the agent's personal interaction history and provide the information required for calculating direct trust assessments.
2. **Acquaintance agents**. The trustor maintains knowledge of the consumer agents located within its operational radius. These neighboring consumers may provide ratings of referrals that contribute to reputation-based trust evaluation.
3. **Nearby providers**. The trustor is aware of the provider agents currently located within its operational range and can potentially interact with them.
4. **Trustworthiness assessments of nearby providers**. Using the FIRE trust model, the trustor attempts to calculate trust values for all nearby providers. Providers for which a trust value can be computed are placed in the set *HasTrustValue*, whereas providers for which insufficient evidence exists are placed in the set *NoTrustValue*.
5. **Location information**. The trustor has knowledge of its own position in the spherical simulation world, represented by the polar coordinates (r, φ, θ). Changes in these coordinates can be used to estimate agent mobility.

6. **Witness reputation information**. When evaluating a provider through the Witness Reputation (WR) mechanism, the trustor requests ratings from selected acquaintances that are likely to possess relevant interaction experience. Acquaintances that do not have matching ratings may instead provide referrals to other agents.
7. **Certified reputation information**. Through the Certified Reputation (CR) mechanism, providers maintain certified ratings obtained from previous interactions. When a provider is evaluated, it can supply these certified ratings to the trustor, allowing reputation estimates to be calculated even when direct interaction history is unavailable.

Based on these locally observable data sources, the adaptive trustor derives a set of environmental indicators that approximate the hidden conditions of the system. These indicators constitute the state representation supplied to the Deep Q-Network and enable the learning agent to adapt its trust model selection strategy to changing environmental conditions.

4.3.2. Environmental Features

The selected feature set was designed to capture the six main categories of environmental dynamics considered in the simulation environment:

1. Changes in the provider population.
2. Changes in the consumer population.
3. Variations in providers' average performance levels.
4. Changes in providers' performance profiles.
5. Provider mobility within the environment.
6. Consumer mobility within the environment.

Based on these environmental factors, a state representation consisting of nine observable features was constructed. These features are calculated using information available to the trustor, including its local ratings database, nearby providers, neighboring consumers, trust assessments produced by the FIRE model, certified reputation information, and the agent's own location within the environment.

The complete state vector used as input to the Deep Q-Network is composed of the following features:

- *ProvidersPopulationDirectChangeEstimate*

- *MeanProvidersPopulationDirectChangeEstimate*
- *ProvidersPopulationIndirectChangeEstimate*
- *MeanProvidersPopulationIndirectChangeEstimate*
- *NewcomerEstimate*
- *ConsumersPopulationIndirectChangeEstimate*
- *MeanConsumersPopulationIndirectChangeEstimate*
- *ProvidersPerformanceChangeEstimate*
- *ConsumersLocationChangeEstimate*

Collectively, these features provide an approximation of the current environmental conditions and enable the adaptive trustor to identify situations in which either FIRE or CA is expected to produce higher utility. The following subsections describe each feature in detail and explain how it is calculated from locally observable information.

4.3.3. Features Related to Provider Population Changes

One of the most influential factors affecting the effectiveness of trust mechanisms in open Multi-Agent Systems is provider population turnover. The continuous arrival and departure of service providers can significantly alter the availability and reliability of trust information, thereby affecting the performance of trust-based decision-making processes.

In highly dynamic environments, newly introduced providers may not possess sufficient interaction histories to support accurate trust assessments. As a result, trust models that rely heavily on historical information may experience reduced effectiveness. Since the adaptive trustor does not have direct knowledge of provider population changes, it must estimate such changes using locally observable information.

To capture provider population dynamics, four complementary features are employed: two direct estimates and two indirect estimates. Together, these features provide indications of the rate at which providers enter or leave the system.

4.3.3.1. ProvidersPopulationDirectChangeEstimate

The first feature measures the proportion of nearby providers that appear to be new within the trustor's operational range. Let $nearbyProviders_t$ denote the set of providers located within the trustor's radius of operation during round t, and $nearbyProviders_{t-1}$ the corresponding set from the previous round.

The set of newly observed providers, denoted by $newProviders_t$, consists of providers that are present in $nearbyProviders_t$ but were not observed in $nearbyProviders_{t-1}$. The direct population change estimate is then calculated as:

$$ProvidersPopulationDirectChangeEstimate_t = \frac{|newProviders_t|}{|nearbyProviders_t|}$$

This ratio expresses the fraction of nearby providers that have recently appeared in the trustor's neighborhood. Higher values suggest increased provider turnover and potentially greater uncertainty regarding provider trustworthiness.

However, this measure is inherently noisy because providers may appear as new not only due to entering the system but also as a result of mobility, either on the part of the providers themselves or of the trustor.

4.3.3.2. MeanProvidersPopulationDirectChangeEstimate

To reduce the effects of temporary fluctuations and mobility-related noise, the adaptive trustor computes an average direct population change estimate over a sliding window of consecutive simulation rounds.

Given a window $W = \{t_1, t_2, \cdots t_n\}$, the mean estimate is calculated as:

$$MeanProvidersPopulationDirectChangeEstimate = \frac{\sum_{i=1}^{n} ProvidersPopulationDirectChangeEstimate_{t_i}}{n}$$

By aggregating observations across multiple rounds, this feature provides a more stable indication of long-term provider population changes.

4.3.3.3. ProvidersPopulationIndirectChangeEstimate

In addition to directly observing newly appearing providers, the adaptive trustor can infer provider turnover from the availability of trust information.

Newly introduced providers often lack sufficient interaction histories. Consequently, their trustworthiness may not be computable through *Interaction Trust (IT)* or *Witness Reputation (WR)*. Some providers may possess no trust evidence at all, while others may only provide *Certified Reputation (CR)* information originating from previous interactions in different systems.

Let $NoTrustValue_t$ denote the set of nearby providers for which no trust value can be computed, and let A_t represent the set of providers whose trustworthiness can only be estimated through certified reputation. The indirect population change estimate is defined as:

$$ProvidersPopulationIndirectChangeEstimate_t = \frac{|NoTrustValue_t| + |A_t|}{|nearbyProviders_t|}$$

This feature captures the proportion of providers that appear to be newcomers based on the lack of locally available trust information.

4.3.3.4. MeanProvidersPopulationIndirectChangeEstimate

Similarly to the direct estimate, the indirect estimate may fluctuate from round to round due to local environmental variations. Therefore, a moving average is also computed:

$$MeanProvidersPopulationIndirectChangeEstimate = \frac{\sum_{i=1}^{n} ProvidersPopulationIndirectChangeEstimate_{t_i}}{n}$$

The resulting feature provides a smoother and more reliable estimate of provider population turnover over time.

Taken together, these four features enable the adaptive trustor to detect both explicit and implicit signs of provider population change. Since the experimental results presented in Chapter 3 demonstrated that provider turnover significantly influences the relative performance of FIRE and CA, accurately estimating this environmental factor is essential for effective trust model selection.

4.3.4. Features Related to Consumer Population Changes

Consumer population turnover constitutes another important source of environmental dynamics in open Multi-Agent Systems. Frequent arrivals and departures of consumers affect the availability of trust information and may reduce the effectiveness of trust mechanisms that rely on historical interactions and reputation exchange. In particular, newly introduced consumers possess little or no interaction history, limiting their ability to evaluate providers accurately through conventional trust assessment methods.

Since the adaptive trustor has no direct knowledge of the actual rate of consumer population change, it must infer this information from observable indicators available through its own interactions. To achieve this, three features are employed to estimate consumer population turnover: the *NewcomerEstimate*, the *ConsumersPopulationIndirectChangeEstimate*, and the *MeanConsumersPopulationIndirectChangeEstimate*.

4.3.4.1. NewcomerEstimate

The first feature attempts to determine whether the adaptive trustor itself is likely to be a newcomer in the system. A newly introduced consumer has not yet accumulated ratings regarding nearby providers and therefore lacks the interaction history required by trust mechanisms such as FIRE.

The feature is calculated by examining the trustor's local ratings database. If no ratings are available for any of the providers located within the trustor's operational range, the agent assumes that it is a newcomer and assigns a value of 1 to the feature. Otherwise, the feature is assigned a value of 0.

Formally:

$$NewcomerEstimate = \begin{cases} 1, if\ \forall provider \in nearbyProviders: \nexists rating = (consumer, provider, _, _, _, _) \in ratingDatabase_{consumer} \\ 0, otherwise \end{cases}$$

This binary indicator provides useful information regarding the trustor's familiarity with its local environment and the amount of trust evidence available for decision making.

4.3.4.2. ConsumersPopulationIndirectChangeEstimate

The second feature estimates consumer population turnover indirectly through the witness reputation process used by FIRE.

When a consumer evaluates a provider using witness reputation, it requests ratings from neighboring consumers that may possess relevant interaction histories. If a neighboring consumer has previously interacted with a provider, it can return corresponding ratings. However, newcomer consumers typically possess no such information and therefore return no ratings.

Let C_t denote the set of consumer acquaintances queried during round t, and let B_t represent the subset of those acquaintances that return no ratings for any query. The indirect estimate of consumer population change is then calculated as:

$$ConsumersPopulationIndirectChangeEstimate_t = \frac{|B_t|}{|C_t|}$$

Higher values indicate that a larger proportion of neighboring consumers appear unable to provide trust information, suggesting increased consumer turnover within the local environment.

4.3.4.3. MeanConsumersPopulationIndirectChangeEstimate

Because local observations may vary significantly between rounds, a moving average is computed over a window of consecutive simulation rounds to provide a more stable estimate of consumer population change.

Given a window containing n rounds, the mean estimate is defined as:

$$MeanConsumersPopulationIndirectChangeEstimate = \frac{\sum_{i=1}^{n} ConsumersPopulationIndirectChangeEstimate_{t_i}}{n}$$

This feature reduces short-term fluctuations and captures longer-term trends in consumer population dynamics.

Together, the three features described above provide indirect evidence regarding the presence and magnitude of consumer population turnover. Such information is particularly important because previous experimental findings demonstrated that changes in consumer populations can significantly affect the relative performance of FIRE and CA. By incorporating these indicators into the state representation, the adaptive trustor is better equipped to identify situations in which one trust mechanism may be expected to outperform the other.

4.3.5. Features Related to Performance and Mobility Changes

In addition to population turnover, the effectiveness of trust mechanisms may also be influenced by changes in providers' behavior and by the mobility of agents within the environment. Providers may alter the quality of the services they deliver, switch to different performance profiles, or change their location. Similarly, consumers may move to new areas of the environment, thereby changing the set of providers and consumers with which they can interact.

As in the case of population changes, these environmental dynamics are not directly observable by the adaptive trustor. Consequently, indirect indicators must be employed to estimate their occurrence. To capture such changes, the state representation includes two additional features: the *ProvidersPerformanceChangeEstimate* and the *ConsumersLocationChangeEstimate*.

4.3.5.1. ProvidersPerformanceChangeEstimate

One of the key challenges in dynamic environments is that a provider's performance may vary over time. A provider that was previously reliable may gradually become less effective, while a low-performing provider may improve its behavior.

Furthermore, providers may switch between predefined performance profiles, resulting in significant changes in the quality of the services they offer.

To estimate these behavioral changes, the adaptive trustor compares the performance predicted by the FIRE trust model with the actual utility gained after interacting with a provider. If the provider behaves consistently, the predicted and observed performance values should be relatively close. Large deviations, however, may indicate changes in the provider's behavior or performance profile.

The feature is calculated as the average absolute difference between the expected performance and the actual performance observed during the last N interactions with a provider:

$$ProvidersPerformanceChangeEstimate = \frac{\sum_i^N |trustValue_i * 10 - actualPerformance_i|}{N}$$

where $trustValue_i$ denotes the trust value assigned by FIRE during interaction i, and $actualPerformance_i$ represents the utility gain obtained from that interaction.

Higher values of this feature indicate that the provider's observed behavior deviates significantly from previous expectations, suggesting either performance fluctuations or transitions between different provider profiles. Consequently, this feature serves as an indicator of behavioral instability within the provider population.

4.3.5.2. ConsumersLocationChangeEstimate

Agent mobility constitutes another important source of environmental change. When a consumer moves within the environment, its set of nearby providers and neighboring consumers may change substantially. As a result, previously accumulated trust information may become less useful, while new trust relationships must be established.

Unlike other environmental factors, a consumer has direct knowledge of its own location. The simulation environment represents locations using spherical coordinates (r, φ, θ). Therefore, the adaptive trustor can determine whether it has changed position by comparing its current coordinates with those recorded during the previous simulation round.

The ConsumersLocationChangeEstimate feature is defined as:

$$ConsumersLocationChangeEstimate = \begin{cases} 1, if\ \Delta r = |r_t - r_{t-1}| \neq 0 \vee \Delta\varphi = |\varphi_t - \varphi_{t-1}| \neq 0 \vee \Delta\theta = |\theta_t - \theta_{t-1}| \neq 0 \\ 0, otherwise \end{cases}$$

The feature takes the value 1 whenever the consumer detects a change in at least one of its spatial coordinates and 0 otherwise.

Although providers' movements cannot be directly observed, provider mobility may indirectly affect other features, particularly those associated with provider population changes. For example, a provider that moves into the consumer's operational range may appear indistinguishable from a newly introduced provider. Consequently, the provider population change features partially capture the effects of provider mobility as well.

4.4. Hyperparameter Configuration and Training Setup

The effectiveness of a Deep Q-Network depends not only on the selected state representation and network architecture but also on the configuration of the learning process. The training procedure of the adaptive trustor therefore requires the specification of a number of hyperparameters that control exploration, learning, experience replay, and value estimation.

Due to the large combinatorial search space and the computational cost associated with extensive experimentation, a systematic hyperparameter optimization process was not performed. Instead, the selected values were obtained through an informal empirical search aimed at identifying a configuration that produced stable learning behavior and satisfactory performance across different environmental conditions.

Table 7 summarizes the hyperparameters used throughout all experiments.

Table 7: Hyperparameters and their values

Parameter	Value
Sliding window size (n)	10
Exploration constant (ε)	0.05
L2 regularization coefficient (λ)	0.01
DQN learning rate	0.30
Backpropagation learning rate	0.15
Discount factor (γ)	0.95
Replay memory size	50
Minibatch size	5
Target network update interval	5 steps

The parameter $n = 10$ determines the size of the observation window used when calculating moving-average environmental features. By aggregating observations over ten consecutive rounds, the adaptive trustor obtains more stable estimates of environmental changes while reducing the influence of temporary fluctuations.

Exploration during training is controlled through an $\varepsilon - greedy$ action-selection policy with $\varepsilon = 0.05$. Under this strategy, the adaptive trustor selects the action with the highest estimated Q-value with probability $1 - \varepsilon$, while a random action is chosen with probability ε. This mechanism balances exploitation of acquired knowledge with continued exploration of alternative trust model selections.

The discount factor is set to $\gamma = 0.95$, allowing the agent to consider the long-term consequences of its decisions while still assigning greater importance to immediate utility gains. Since trust model selection influences future interaction outcomes, incorporating future rewards is essential for learning an effective policy.

To improve learning stability, experience replay is employed. Interaction experiences are stored in a replay memory containing up to 50 transitions. During training, network updates are performed using randomly sampled minibatches of size 5. This approach reduces correlations between consecutive experiences and improves the efficiency of the learning process.

The DQN is trained using a learning rate of 0.30, while a backpropagation learning rate of 0.15 is used during weight updates. In addition, L2 regularization with coefficient $\lambda = 0.01$ is applied to reduce overfitting and encourage smoother parameter estimates.

Finally, a separate target network is maintained to stabilize Q-value estimation. The target network parameters are synchronized with those of the main network every five training steps. This mechanism mitigates oscillations during learning and is a standard technique for improving the convergence properties of Deep Q-Network algorithms.

Overall, these hyperparameter settings provide a balance between learning speed, exploration, and training stability, enabling the adaptive trustor to learn an effective trust model selection policy under a wide range of dynamic environmental conditions.

4.5. Experimental Methodology

The experimental evaluation presented in this chapter is evaluated using the same simulation environment, performance measure, statistical analysis procedure, and provider population configuration presented in the previous chapter. Consequently, all experimental settings remain unchanged.

The purpose of the present evaluation is to examine whether an agent capable of dynamically selecting between FIRE and CA can achieve superior performance compared to agents that rely exclusively on a single trust model.

The experimental methodology therefore differs from that of the previous chapter only in the introduction of a new consumer group, the specification of the Number of Independent Simulation Runs (NISR) per experiment, and the number of simulation rounds used in selected experiments.

4.5.1. Consumer Groups

The evaluation compares the performance of three distinct groups of consumer agents:

- **Adaptable**: consumer agents capable of dynamically selecting either the FIRE or the CA trust model based on their assessment of environmental conditions using Deep Q-Learning.
- **FIRE**: consumer agents that use only the FIRE trust model.
- **CA**: consumer agents that use only the CA trust model.

The FIRE and CA groups serve as baseline approaches, while the Adaptable group represents the proposed adaptive trust model selection mechanism.

4.5.2. Number of Independent Simulation Runs

To obtain statistically reliable results and reduce the effect of random fluctuations, each experiment is executed multiple times. The Number of Independent Simulation Runs (NISR) per experiment is shown in Table 8.

Table 8: Number of Independent Simulation Runs (NISR) per experiment

Experiment	NISR
1, 2, 6, 8-11, 17	30
3-5, 7, 12, 14-16, 18	10

13	12

4.5.3. Simulation Rounds

Unless otherwise stated, simulations are executed for 500 rounds. To investigate longer-term adaptation under highly dynamic conditions, Experiments 7 and 13 are executed for 1000 rounds (Table 9).

Table 9: Number of simulation rounds

Experiment	Rounds (N)
Default	500
Experiments 7 and 13	1000

All other experimental variables and trust model parameters are identical to those presented in the previous chapter.

4.6. Experimental Scenarios and Results

The experimental scenarios are designed to investigate the adaptive trustor's ability to learn an effective trust model selection policy under a variety of environmental conditions. More specifically, the experiments are organized into three categories:

- **Single Environmental Change**, where only one environmental factor is active throughout the simulation.
- **Combined Environmental Changes**, where multiple environmental factors occur simultaneously.
- **Time-Varying Environmental Changes**, where the environmental conditions themselves change during the course of the simulation.

The first category evaluates whether the adaptive trustor can identify which trust model performs better when a single source of environmental change is present. The second category examines the agent's behavior in more complex situations involving multiple concurrent changes. Finally, the third category investigates whether the adaptive trustor can continue adapting when the environmental conditions evolve over time.

The following sections present the results obtained for each category of experiments and analyze the effectiveness of adaptive trust model selection under increasingly dynamic operating conditions.

4.6.1. Adaptable's Performance under Single Environmental Changes

In this section, we compare the performance of three groups of consumers (Adaptable, CA, and FIRE) when there is no environmental change (a static setting) and when conducting single environmental factor changes while remaining constant throughout the simulation, through the following experiments:

- **Experiment 1**: The setting is static, with no changes.
- **Experiment 2**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$).
- **Experiment 3**: The provider population changes at maximum 5% in every round ($p_{PPC} = 0.05$).
- **Experiment 4**: The provider population changes at maximum 10% in every round ($p_{PPC} = 0.10$).
- **Experiment 5**: The consumer population changes at maximum 2% in every round ($p_{CPC} = 0.02$).
- **Experiment 6**: The consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$).
- **Experiment 7**: The consumer population changes at maximum 10% in every round ($p_{CPC} = 0.10$).
- **Experiment 8**: A provider may alter its average level of performance at maximum 1.0 UG unit with a probability of 0.10 each round ($\mathrm{p_{\mu C}} = 0.10, \mathrm{M} = 1.0$).
- **Experiment 9**: A provider may switch to a different (performance) profile with a probability of 2% in every round ($\mathrm{p_{ProfileSwitch}} = 0.02$).
- **Experiment 10**: A consumer may move to a new location within the spherical world by a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($\mathrm{p_{CLC}} = 0.10, \Delta\Phi = \pi/20$).

- **Experiment 11**: A provider may move to a new location within the spherical world by a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($p_{PLC} = 0.10, \Delta\Phi = \pi/20$).

The research question addressed by this set of experiments is whether the Adaptable consumer can maintain its performance higher than that of the worst choice (i.e., the trust model with the lowest performance), when only one environmental change occurs.

The experiments' results are presented in Figures 17–27.

Overall, the performance of the Adaptable consumer is satisfactory in all the experiments and follows the performance of the best-choice model in each situation. This conclusion is drawn more easily if we consider experiments 3–7 (Figures 19–23), where the performance of the adaptable consumer is always between the performance of the other two models. Specifically, when the provider population changes (Figures 19 and 20), the Adaptable's performance is always higher than CA, the worst choice in this case. On the contrary, when the consumer population is volatile (Figures 21–23), the adaptable maintains a performance higher than that of FIRE.

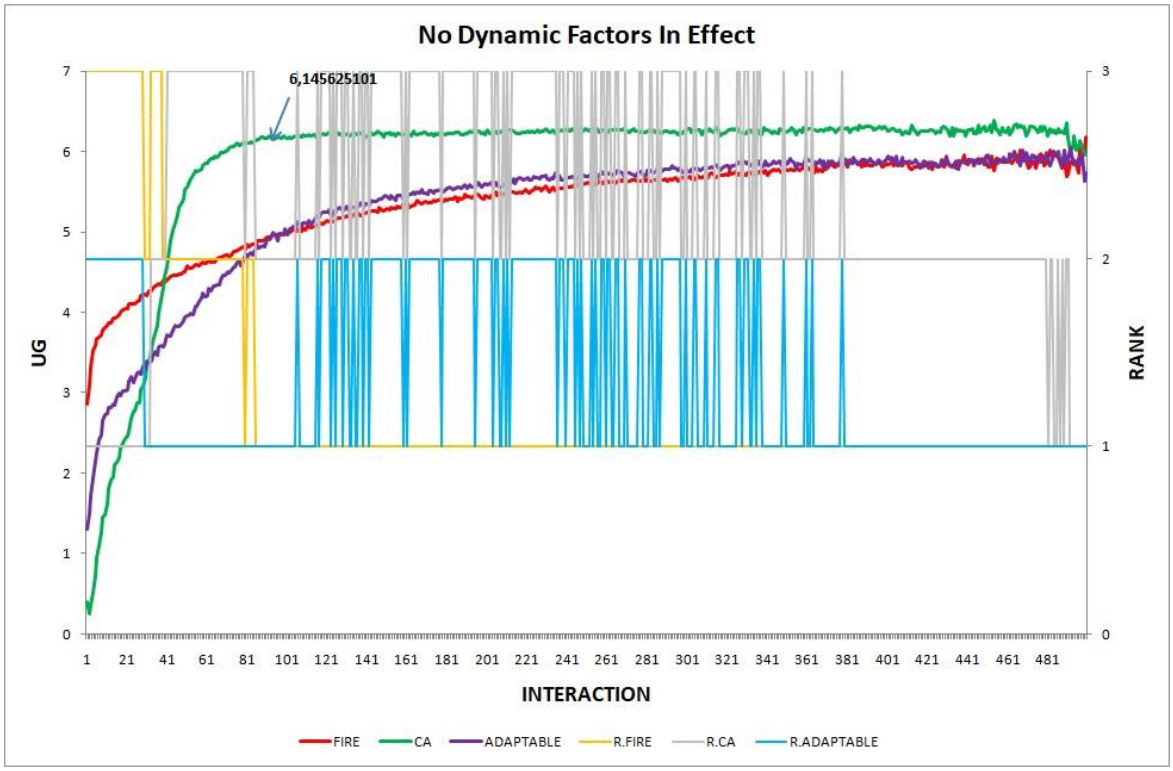


Figure 17. Experiment 1: performance of the adaptable consumer in the static setting

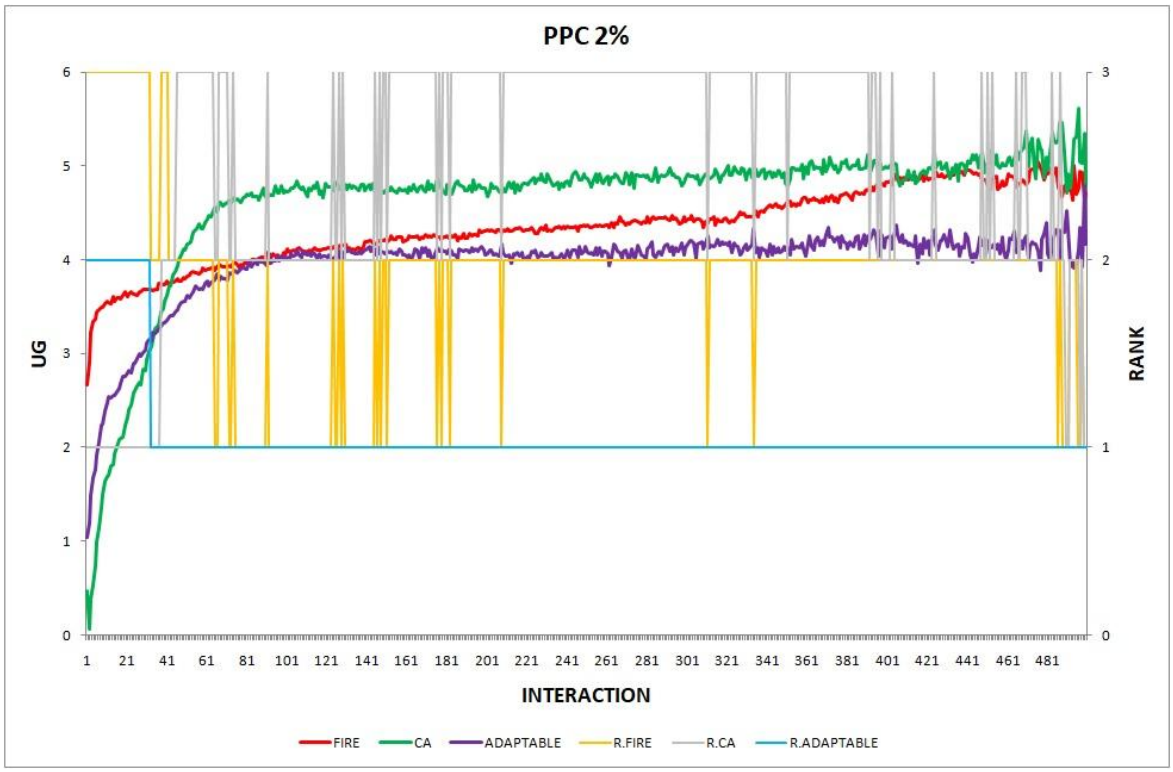


Figure 18. Experiment 2: provider population change; $\boldsymbol{p_{PPC} = 2\%}$

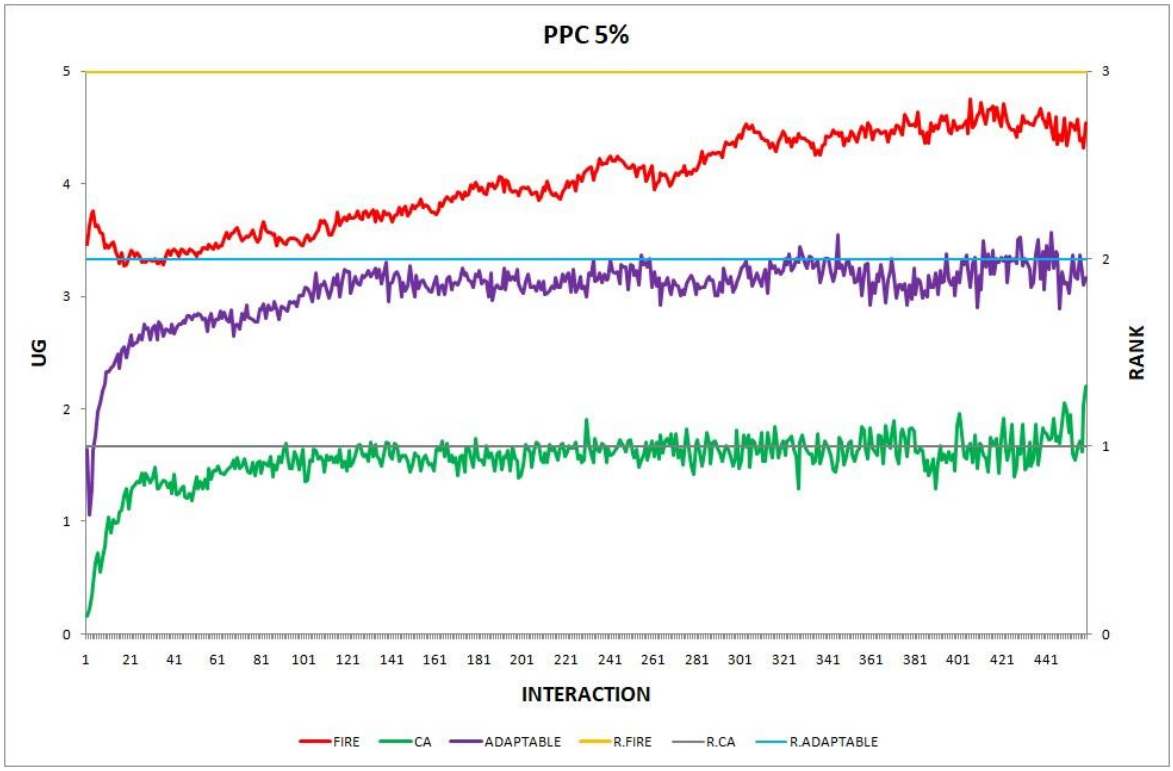


Figure 19. Experiment 3: provider population change; $\boldsymbol{p_{PPC} = 5\%}$

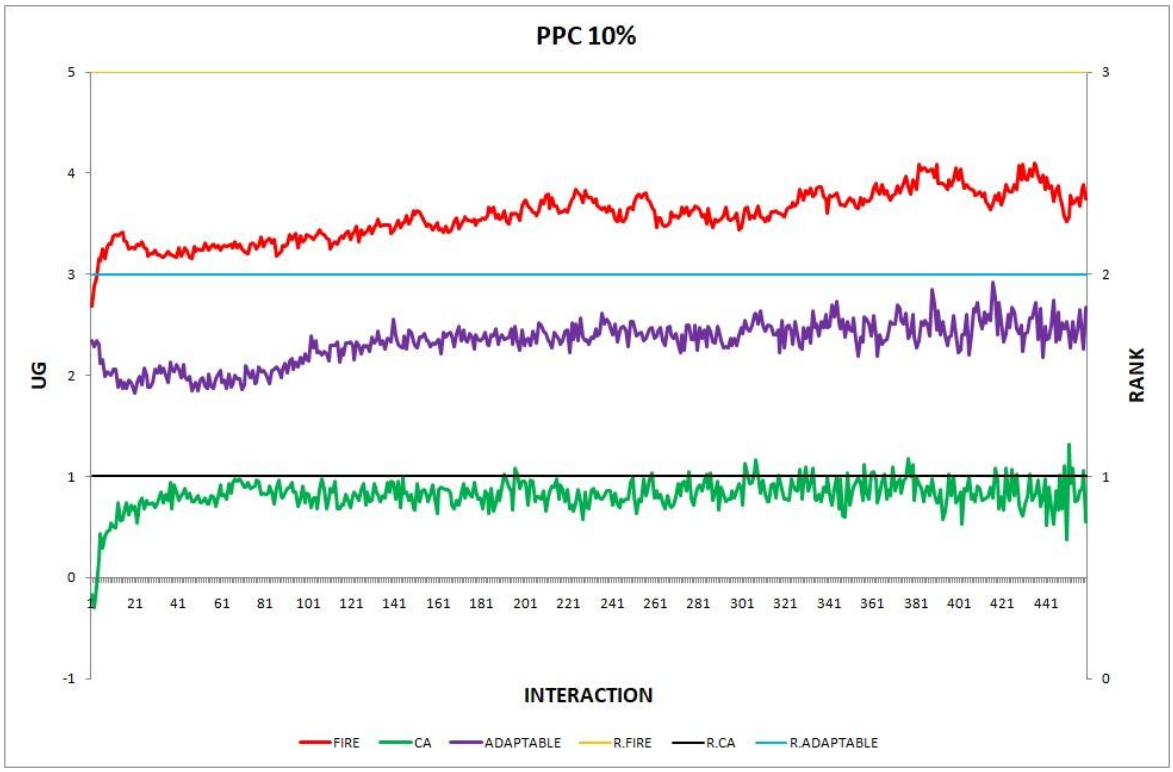


Figure 20. Experiment 4: provider population change; $\boldsymbol{p_{PPC} = 10\%}$

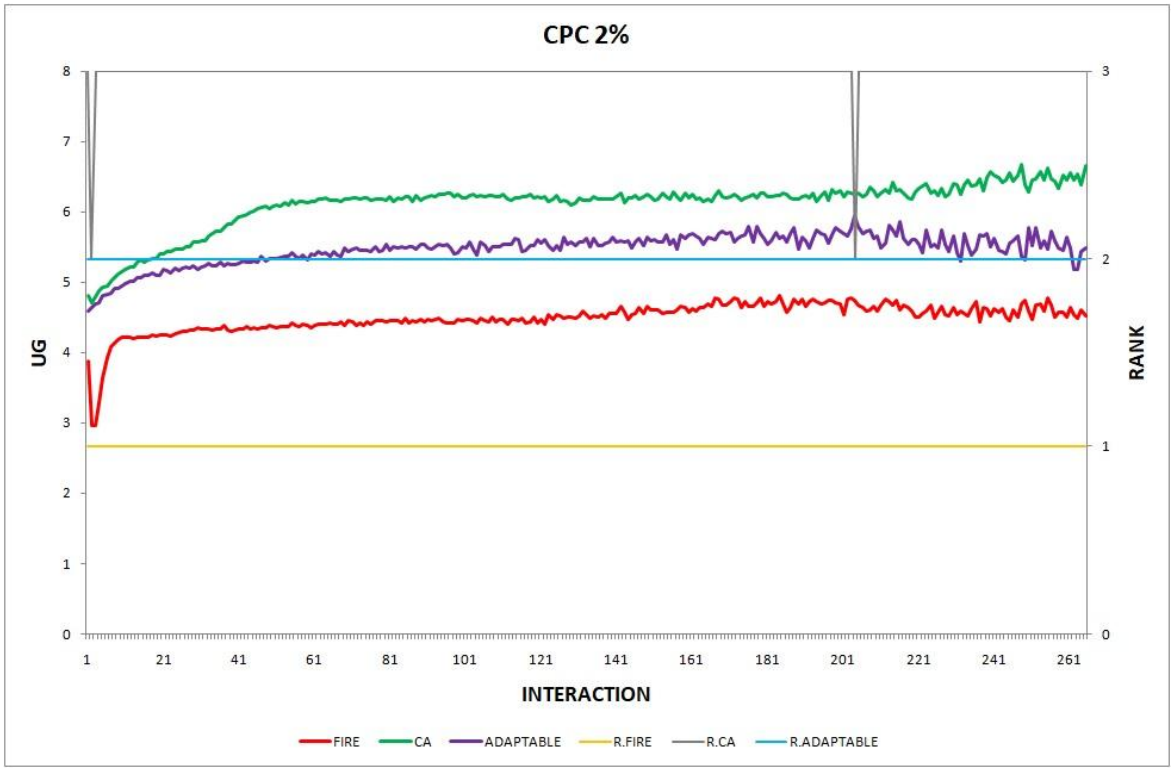


Figure 21. Experiment 5: consumer population change; $\boldsymbol{p_{CPC} = 2\%}$

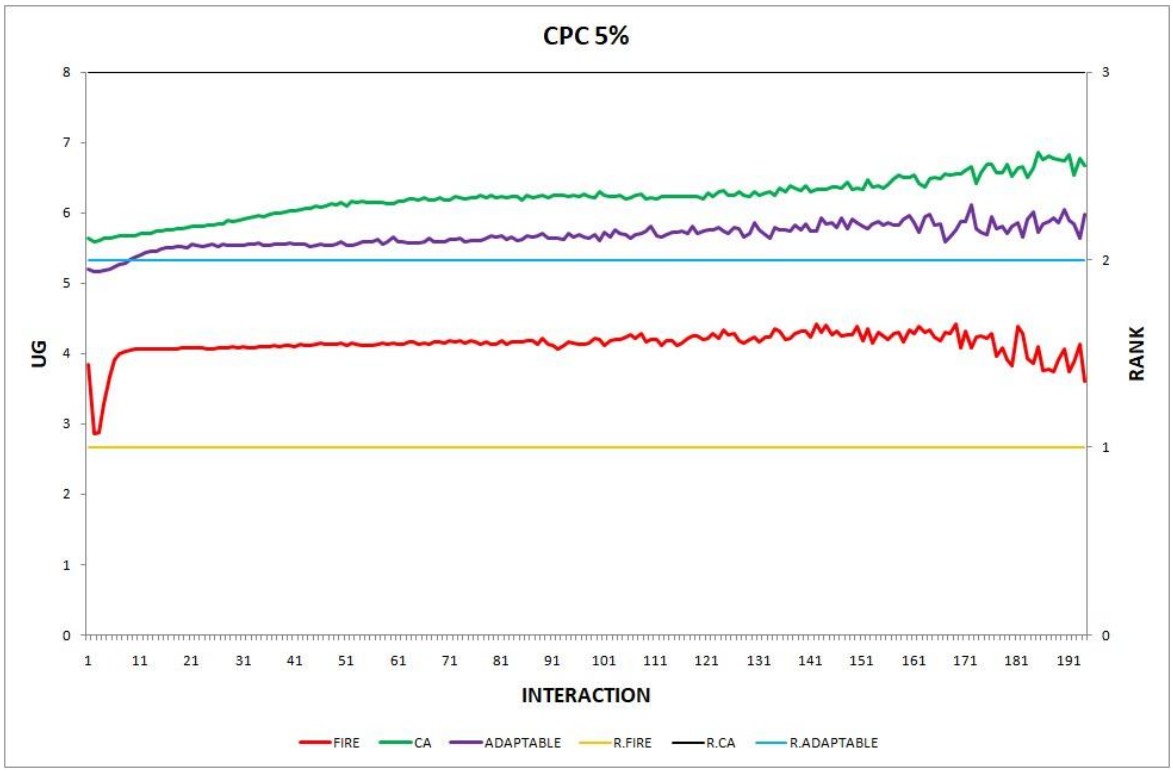


Figure 22. Experiment 6: consumer population change; $\boldsymbol{p_{CPC} = 5\%}$

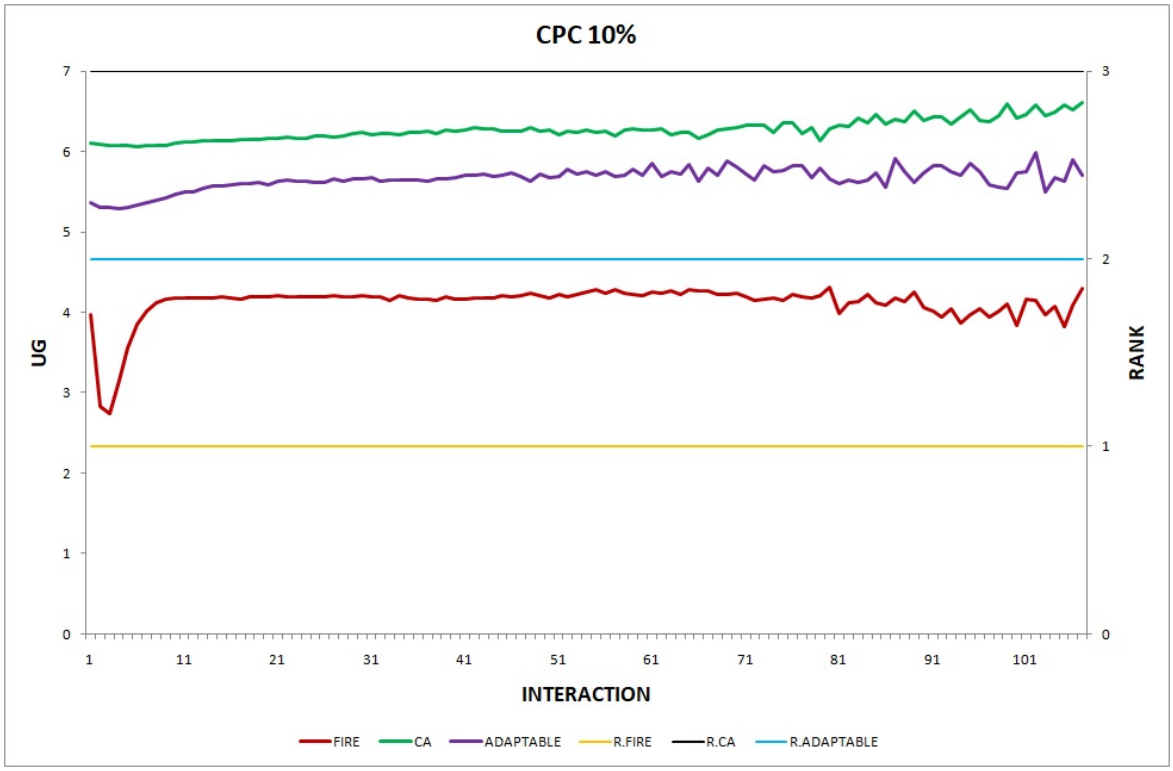


Figure 23. Experiment 7: consumer population change; $\boldsymbol{p_{CPC} = 10\%}$

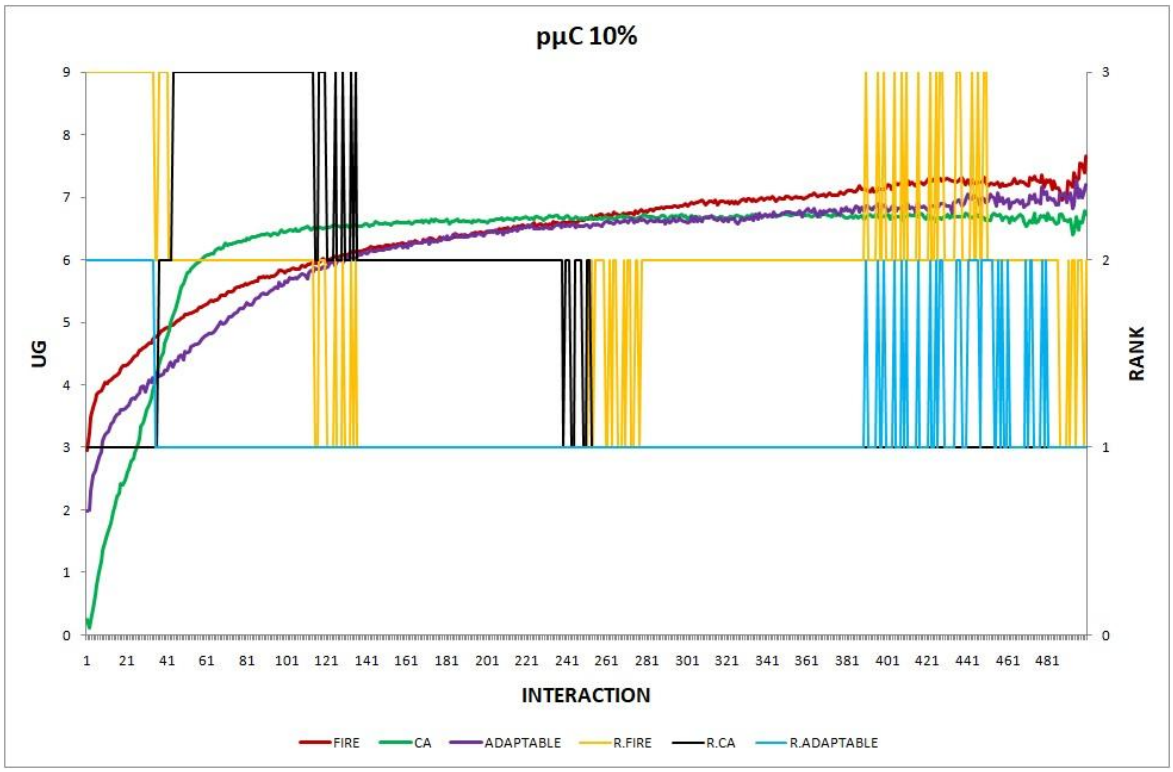


Figure 24. Experiment 8: providers change their performance; $\boldsymbol{p_{CPC} = 10\%, M = 1.0}$

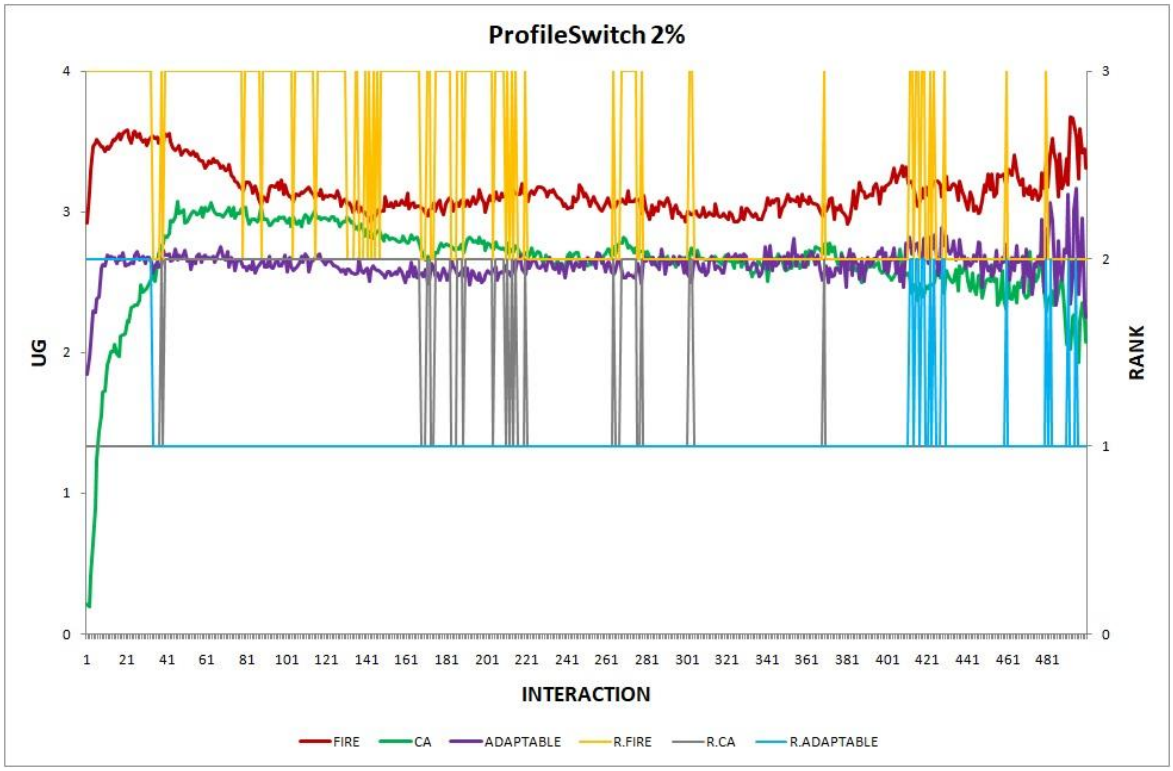


Figure 25. Experiment 9: providers switch their profiles; $\boldsymbol{p_{ProfileSwitch} = 2\%}$

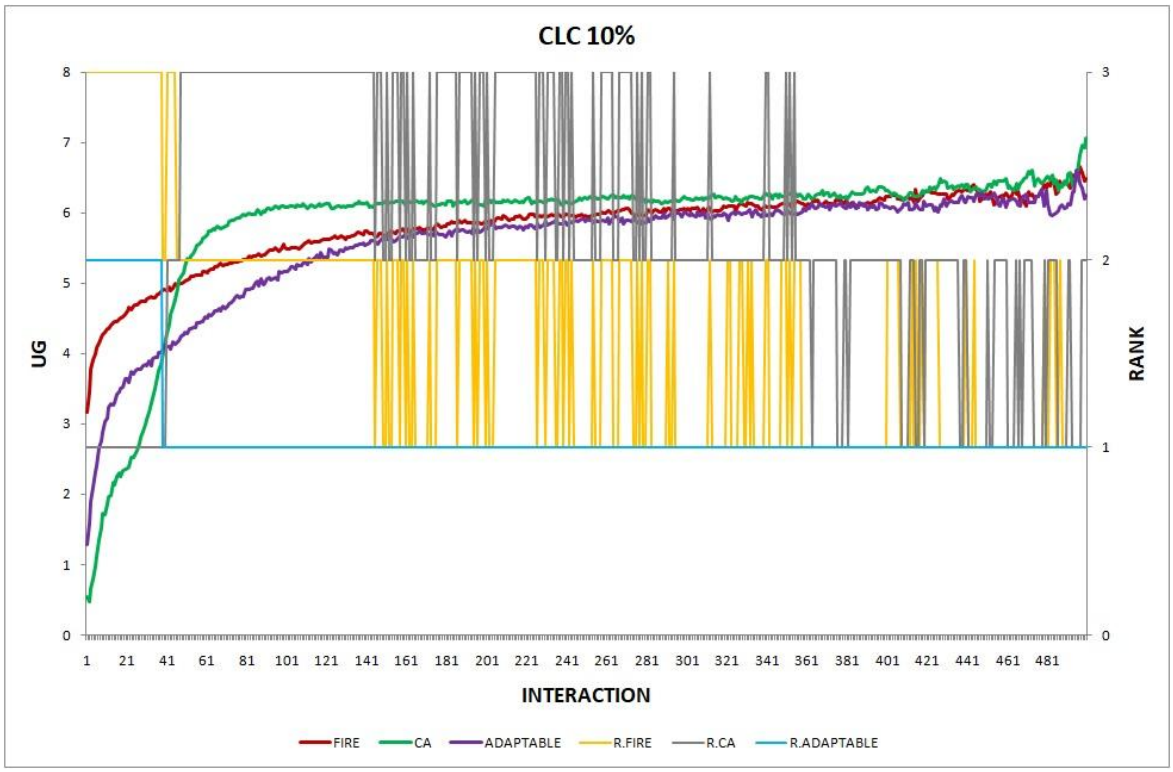


Figure 26. Experiment 10: consumers change their locations; $\boldsymbol{p_{CLC} = 10\%,\ \Delta\Phi = \pi/20}$

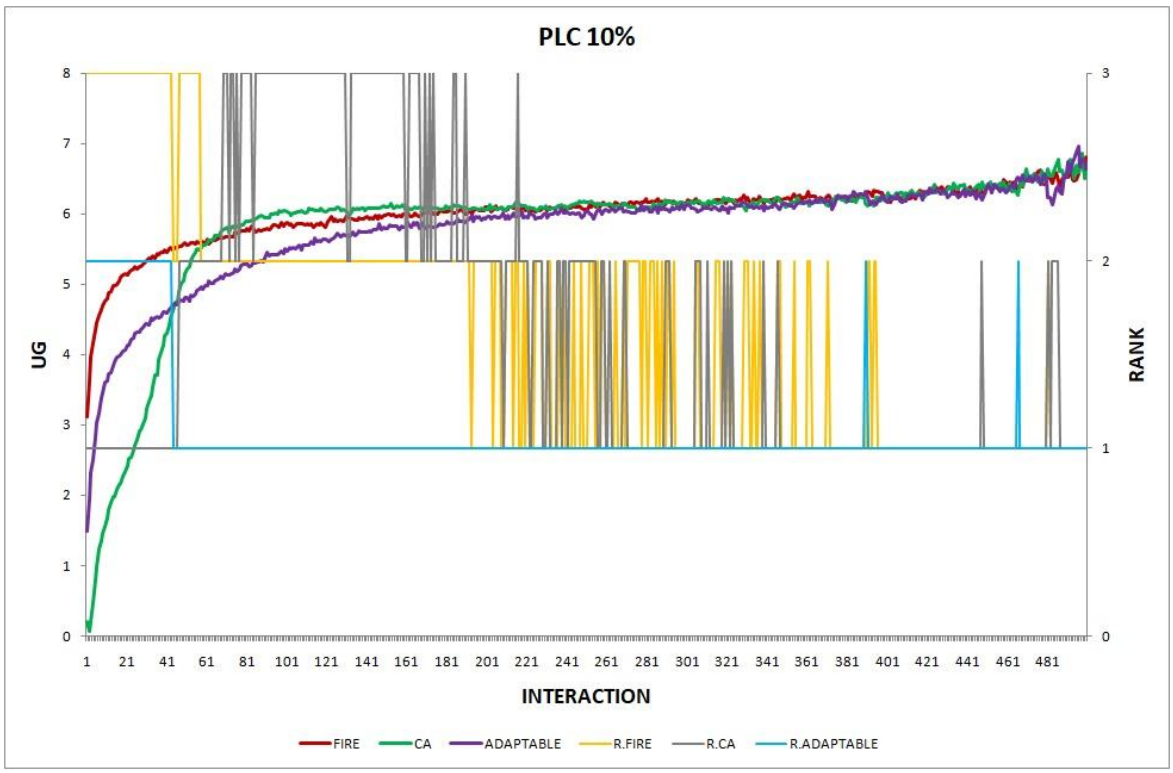


Figure 27. Experiment 11: providers change their locations; $\boldsymbol{p_{PLC} = 10\%,\ \Delta\Phi = \pi/20}$

4.6.2. Adaptable's Performance under Combined Environmental Changes

The experiments of this section aim to demonstrate that the adaptable consumer can find the optimal policy when there are multiple, concurrently acting environmental changes and it is difficult to predict (using the previous section's results) which of the two models, FIRE or CA, is the best option. Note that all changes are maintained constant throughout the simulation rounds of the experiment. The experiments conducted in this section are described as follows:

- **Experiment 12**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$) and the consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$).
- **Experiment 13**: The provider population changes at maximum 10% in every round ($p_{PPC} = 0.10$) and the consumer population changes at maximum 10% in every round ($p_{CPC} = 0.10$).
- **Experiment 14**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$), the consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$), and consumers may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($\mathrm{p_{CLC}} = 0.10, \Delta\Phi = \pi/20$).
- **Experiment 15**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$), the consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$), and both consumers and providers may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($\mathrm{p_{CLC}} = 0.10, \mathrm{p_{PLC}} = 0.10, \Delta\Phi = \pi/20$).
- **Experiment 16**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$), the consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$), both consumers and providers may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($\mathrm{p_{CLC}} = 0.10, \mathrm{p_{PLC}} = 0.10, \Delta\Phi = \pi/20$), and providers may alter their average level of performance at maximum 1.0 UG unit with a probability of 0.10 each round ($p_{\mu C} = 0.10, M = 1.0$).

- **Experiment 17**: The provider population changes at maximum 2% in every round ($p_{PPC} = 0.02$), the consumer population changes at maximum 5% in every round ($p_{CPC} = 0.05$), both consumers and providers may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($\mathrm{p_{CLC}} = 0.10, \mathrm{p_{PLC}} = 0.10, \Delta\Phi = \pi/20$), providers may alter their average level of performance at maximum 1.0 UG unit with a probability of 0.10 each round ($p_{\mu C} = 0.10, M = 1.0$), and providers may switch into a different (performance) profile with a probability of 2% in every round ($\mathrm{p_{ProfileSwitch}} = 0.02$).

The research question we aim to answer in this set of experiments is whether the adaptable consumer will be able to maintain its performance higher than that of the worst choice (the model with the lowest performance), when multiple, concurrently acting environmental changes occur.

The experiments' results are presented in Figures 28–33.

Overall, in no experiment does the performance of the adaptable agent fall below the performance of the model with the worst performance, which supports the conclusion that the adaptable agent indeed succeeds in finding the optimal policy in all the experiments.

An interesting result emerges, in Experiment 12 (Figure 28), where the performance of the adaptable trustor not only maintains performance above the weaker baseline model but actually outperforms both FIRE and CA. This finding suggests that, under certain combinations of environmental conditions, dynamically switching between trust mechanisms may provide advantages that neither static trust model can achieve individually.

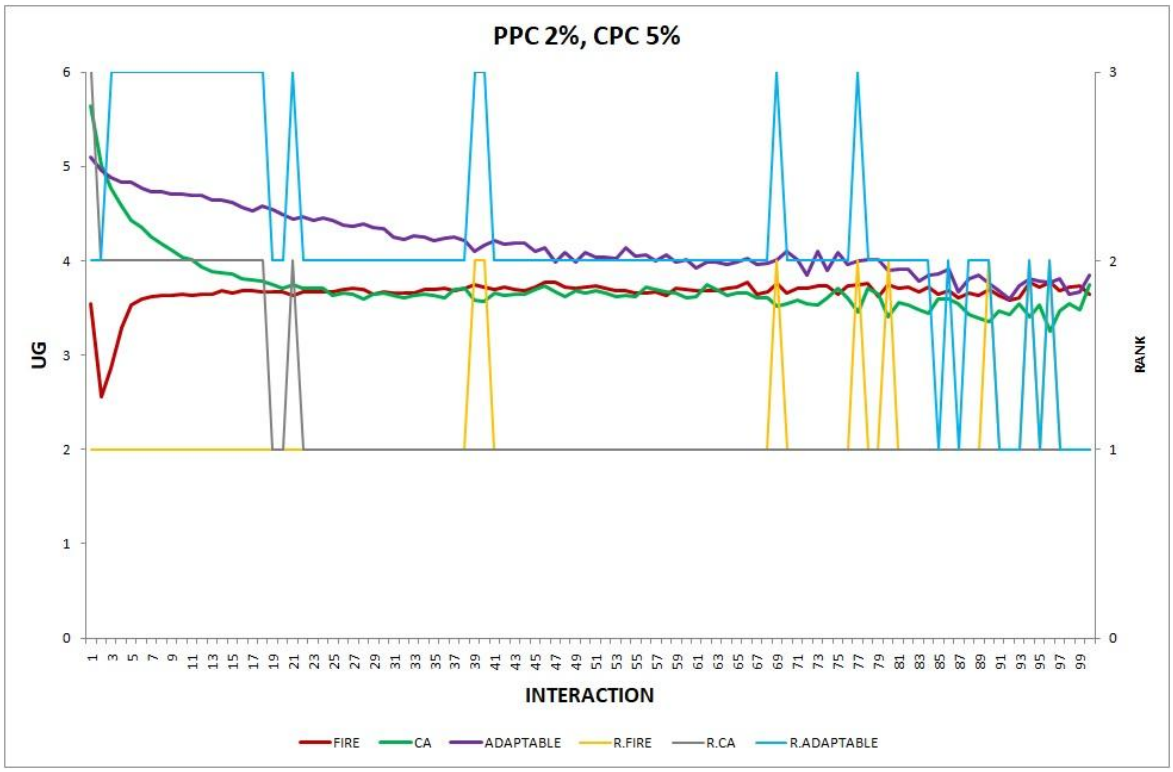


Figure 28. Experiment 12: $\boldsymbol{p_{PPC} = 2\%}$ and $\boldsymbol{p_{CPC} = 5\%}$.

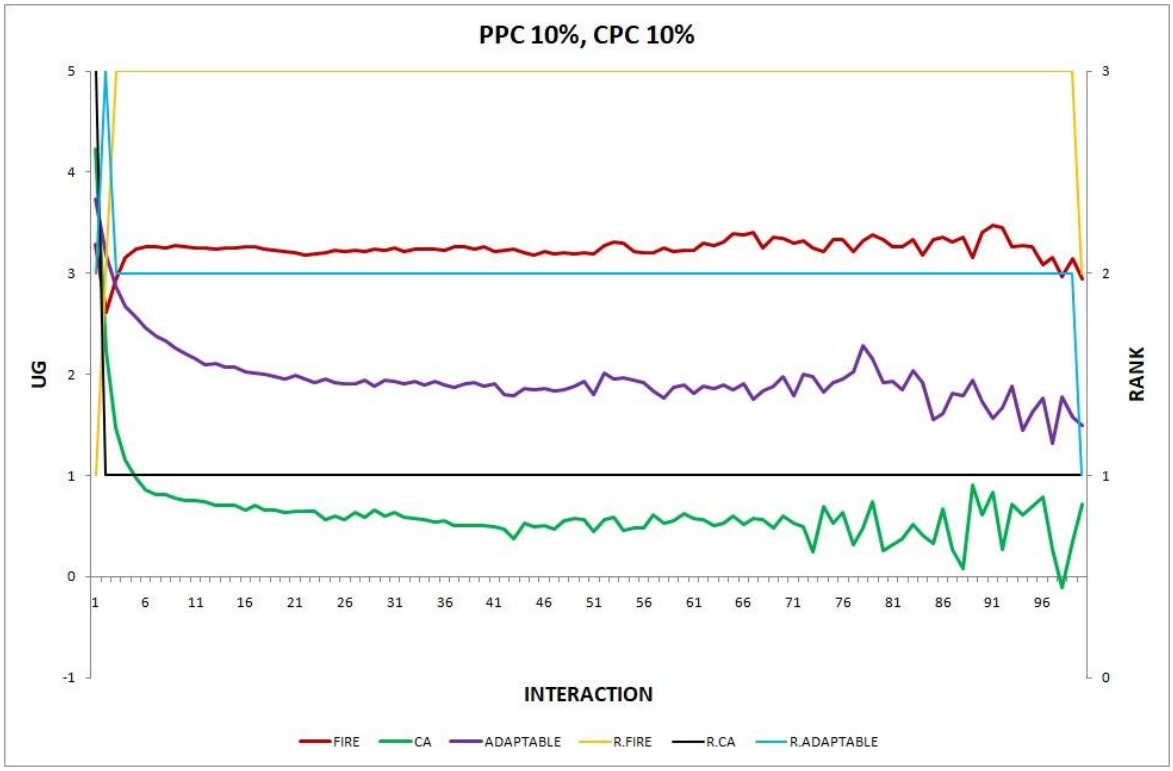


Figure 29. Experiment 13: $\boldsymbol{p_{PPC} = 10\%}$ and $\boldsymbol{p_{CPC} = 10\%}$.

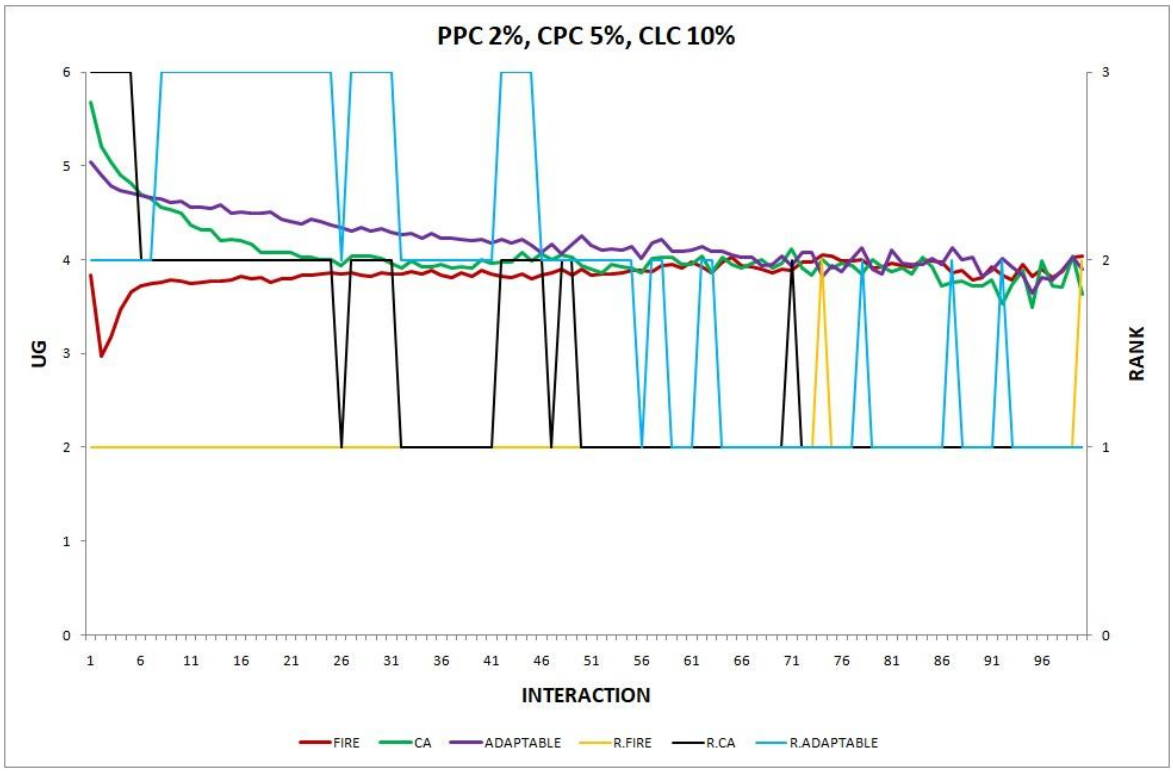


Figure 30. Experiment 14: $\boldsymbol{p_{PPC} = 2\%,\ p_{CPC} = 5\%,\ p_{CLC} = 10\%, \Delta\Phi = \pi/20}$.

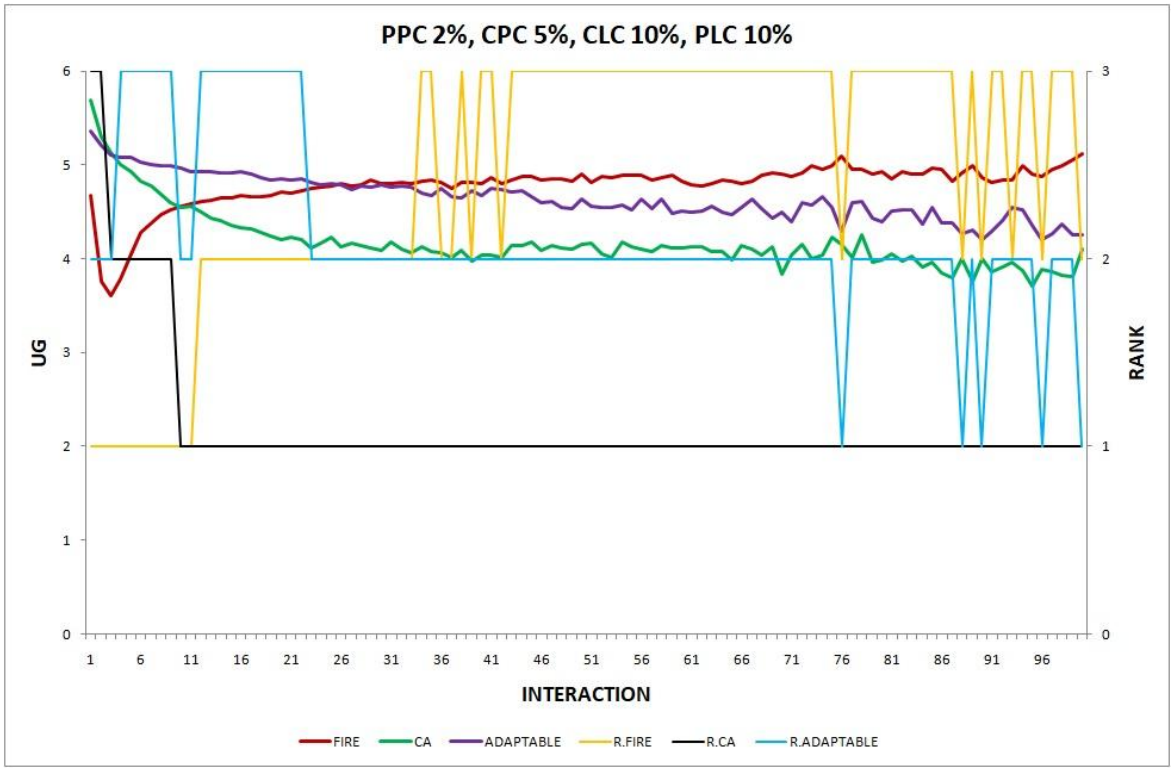


Figure 31. Experiment 15: $\boldsymbol{p_{PPC} = 2\%,\ p_{CPC} = 5\%,\ p_{CLC} = 10\%,\ p_{PLC} = 10\%, \Delta\Phi = \pi/20}$.

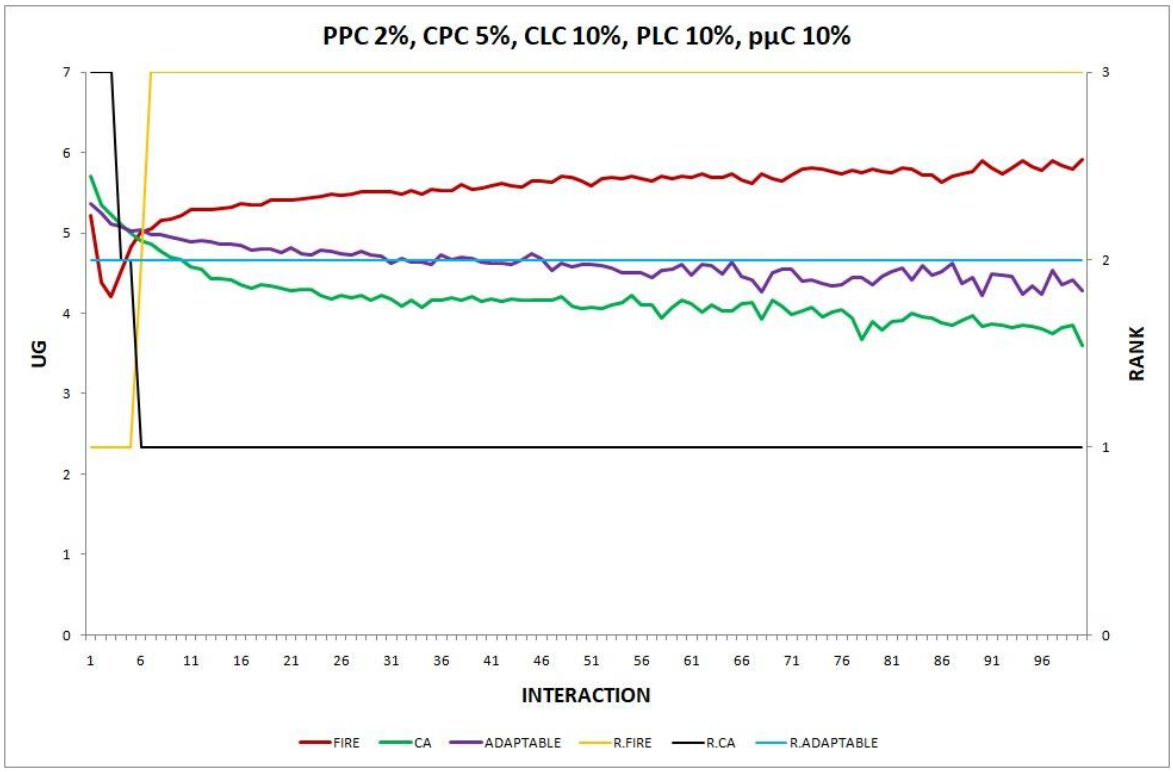


Figure 32. Experiment 16: $\boldsymbol{p_{PPC} = 2\%,\ p_{CPC} = 5\%,\ p_{CLC} = 10\%,\ p_{PLC} = 10\%, \Delta\Phi = \pi/20, p_{\mu C} = 10\%,\ M = 1.0}$.

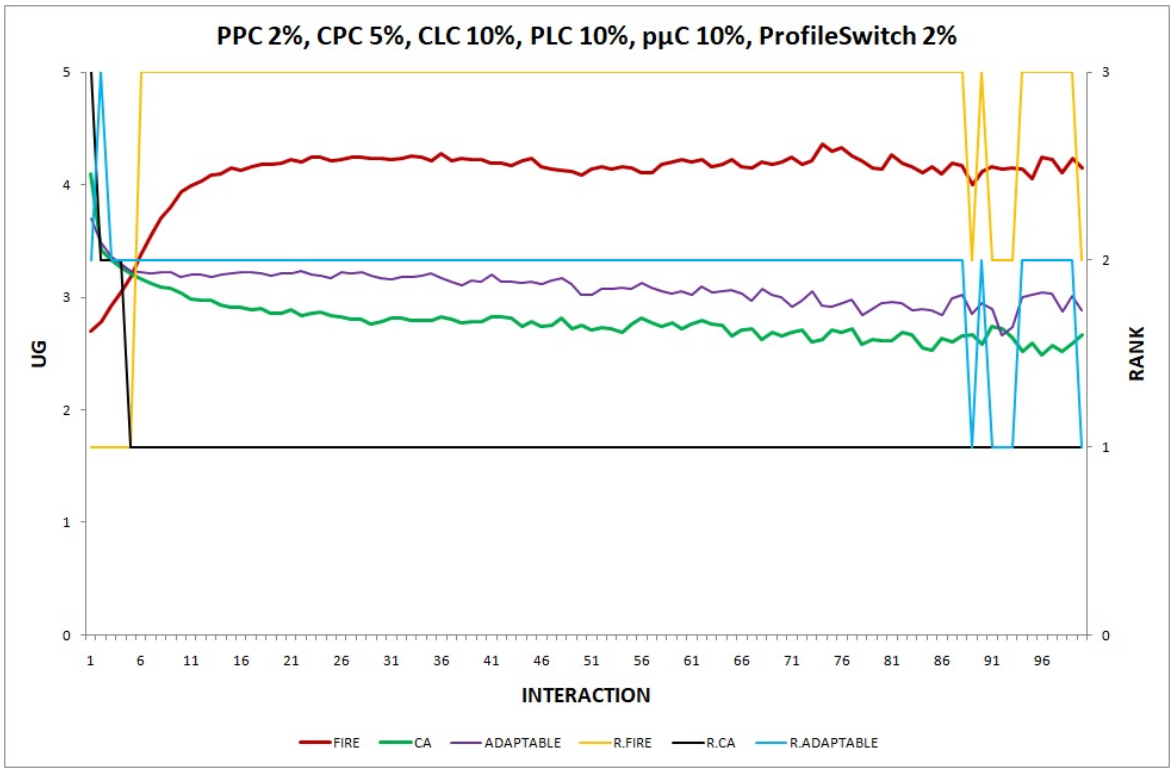


Figure 33. Experiment 17: $\boldsymbol{p_{PPC} = 2\%,\ p_{CPC} = 5\%,\ p_{CLC} = 10\%,\ p_{PLC} = 10\%, \Delta\Phi = \pi/20, p_{\mu C} = 10\%,\ M = 1.0, p_{ProfileSwitch} = 2\%}$.

4.6.3. Adaptable's Performance under Time-Varying Environmental Changes

The final experiment investigates the performance of the adaptive trustor in an environment where the environmental conditions themselves change during the course of the simulation. This scenario is considerably more challenging than those examined in the previous sections, since not only do multiple environmental changes occur, but the type and intensity of these changes vary over time.

As a result, the trust model that is most appropriate at one stage of the simulation may no longer be the optimal choice at a later stage. Consequently, the adaptable trustor must continuously reassess environmental conditions and adjust its trust model selection policy accordingly. The environmental changes applied during the simulation are shown in Table 10.

Table 10. Environmental changes applied in each simulation round. PPC stands for provider population change, and CPC stands for consumer population change.

Simulation Round	Changes Applied
1-200	PPC 2%, CPC 5%
201-250	PPC 2%
251-300	PPC 5%
301-350	PPC 10%
351-400	CPC 2%
401-450	CPC 5%
451-500	CPC 10%

The research question addressed by this experiment is whether the adaptable trustor can maintain a level of performance higher than that of the worst-performing trust model when multiple environmental changes occur and these changes vary throughout the simulation.

The results of the experiment are presented in Figure 34. Overall, the adaptable trustor successfully learns an effective trust model selection policy and adapts its behavior as the environmental conditions evolve over time.

During the initial phase of the simulation, where provider and consumer population changes occur simultaneously, the adaptive trustor exhibits behavior similar to that of the CA trust model, indicating that CA is identified as the most beneficial option under these

conditions. As the environmental conditions change and the relative performance of CA deteriorates, the adaptable trustor adjusts its strategy and maintains utility gains at levels higher than those achieved by the weaker trust model. Finally, during the later stages of the simulation, when consumer population changes again favor the CA approach, the adaptable trustor adapts its behavior accordingly and once more follows the performance of the CA model.

The results indicate that the adaptable agent is capable of continuously adapting its trust model selection policy in response to changing environmental conditions. The ability to adjust behavior as the environment evolves provides further evidence that the proposed Deep Q-Learning approach can successfully identify and exploit the trust mechanism that is expected to yield the highest utility under dynamic operating conditions.

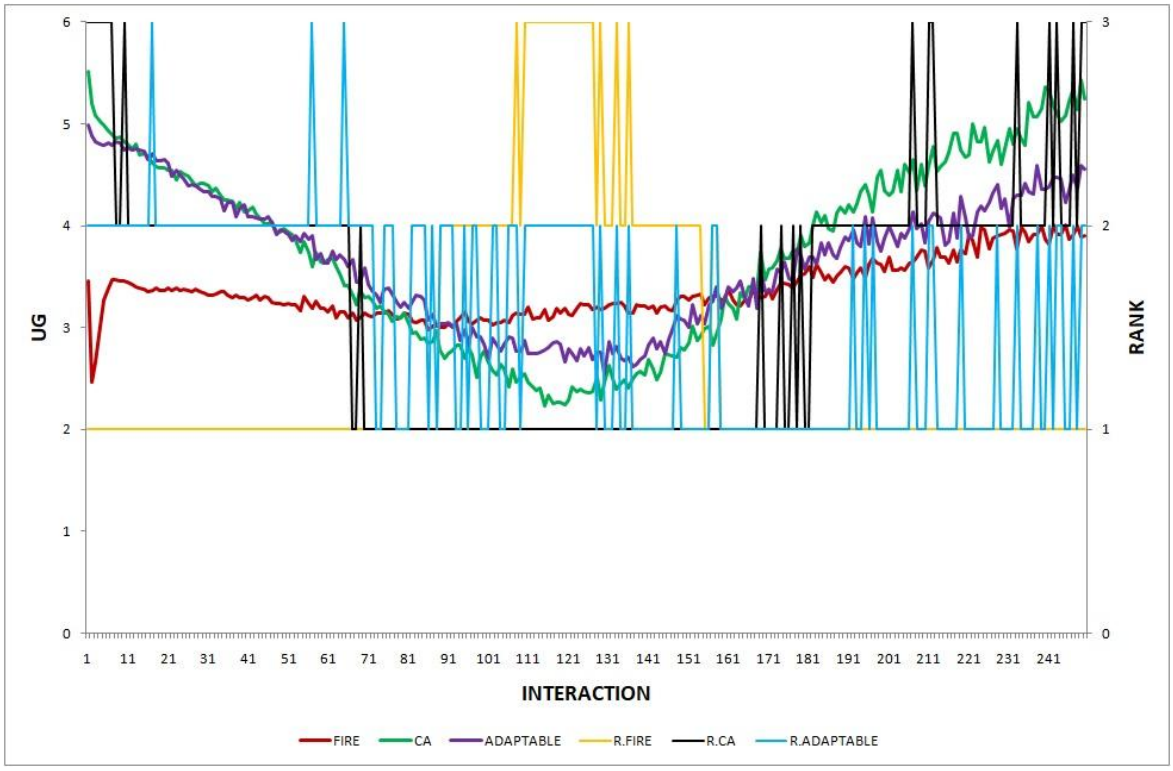


Figure 34. Experiment 18: Performance of Adaptable, FIRE, and CA consumer groups under time-varying environmental changes.

4.7. Explaining the Performance of the Adaptable Trustor

The experimental results presented in the previous sections demonstrate that the adaptable trustor is generally capable of selecting the most appropriate trust mechanism under a wide range of environmental conditions. Across all experimental scenarios, its performance remains consistently above that of the worst-performing trust model,

indicating that the learning process successfully captures important relationships between observable environmental characteristics and trust model effectiveness.

A particularly interesting observation is that, in most experiments, the performance of the adaptable trustor lies between the performances of the FIRE and CA consumer groups. This behavior is consistent with the nature of the learning problem. The adaptable trustor attempts to identify which trust mechanism is more suitable under the current environmental conditions, but it must simultaneously balance exploration and exploitation. Consequently, although it often approaches the performance of the best-performing model, it does not always fully match it.

However, the experimental results also reveal several cases in which the adaptable trustor achieves performance that is superior to both non-adaptive alternatives. These findings suggest that adaptive trust model selection can provide benefits beyond simply avoiding poor decisions and may enable agents to exploit local environmental conditions more effectively than agents permanently committed to a single trust mechanism.

The following subsections discuss these observations in greater detail.

4.7.1. Why the Adaptable Trustor Typically Performs Between FIRE and CA

In the majority of the experiments, the adaptable trustor maintains performance levels that fall between those of the FIRE and CA consumer groups. This outcome is expected, as the objective of the learning agent is to identify and apply the trust mechanism that is most appropriate under the prevailing environmental conditions.

When one trust model consistently outperforms the other, the adaptable trustor gradually learns to favor the superior model. Nevertheless, because learning occurs through interaction with the environment and the agent continues to perform occasional exploratory actions, some suboptimal decisions remain unavoidable. As a result, the adaptable trustor generally approaches the performance of the best-performing model without completely matching it.

This behavior can be observed particularly clearly in Experiments 3-7, where the adaptable consistently achieves utility gains that remain above those of the worst-performing trust model. When provider population changes dominate the environment, FIRE exhibits superior performance and the adaptive trustor maintains performance above that of CA. Conversely, when consumer population changes are present, CA becomes the more effective trust mechanism and the adaptable trustor consistently outperforms FIRE.

These results indicate that the learning process successfully captures the relationship between environmental conditions and trust model effectiveness, allowing the adaptable trustor to avoid persistently selecting the less appropriate trust mechanism.

4.7.2. Cases where the Adaptable Trustor Outperforms Both FIRE and CA

Although the adaptable trustor typically performs between the two baseline models, Experiments 12 and 14 present a particularly noteworthy outcome. In both cases, the adaptable trustor achieves higher utility gains than either FIRE or CA individually.

These experiments involve the simultaneous presence of provider population changes (PPC = 2%) and consumer population changes (CPC = 5%). Previous experimental results demonstrated that these two environmental factors favor different trust mechanisms. Provider population changes tend to favor FIRE, whereas consumer population changes favor CA.

From a global perspective, these conflicting environmental influences make it difficult to determine which trust model should be preferred throughout the entire system. However, environmental changes are not necessarily distributed uniformly across all regions of the environment. Some consumers may experience a higher concentration of newcomer providers, while others may operate in neighborhoods characterized by substantial consumer turnover. Consequently, the optimal trust mechanism may vary locally from one region of the environment to another.

Unlike the FIRE and CA consumer groups, which are permanently restricted to a single trust mechanism, the adaptable trustor can evaluate local environmental conditions and dynamically select the trust model that appears most appropriate at a particular time and location. Furthermore, the agent can revise previous decisions and switch trust mechanisms when environmental conditions change.

This ability to respond to local environmental variations provides additional flexibility and allows the adaptable trustor to exploit opportunities that remain inaccessible to non-adaptive agents. As a result, the adaptable trustor is able to achieve utility gains that exceed those of both FIRE and CA in Experiments 12 and 14.

These findings suggest that adaptive trust model selection may provide benefits not only through avoiding poor choices but also through exploiting heterogeneous environmental conditions more effectively than static trust strategies.

4.7.3. The Impact of Severe Environmental Dynamics

An interesting contrast can be observed between Experiments 12 and 13. While the Adaptable trustor outperforms both FIRE and CA in Experiment 12, this advantage disappears in Experiment 13 when the magnitude of environmental change becomes substantially greater.

The primary difference between the two experiments is the increase in consumer population change. In Experiment 13, the rate of consumer turnover rises to 10%, resulting in a significantly more volatile environment. Under these conditions, newly introduced consumer agents continuously enter the system and begin operating with no prior experience.

Since adaptability is itself a learned capability, newly introduced adaptable consumers must gradually acquire sufficient experience before they can accurately identify the most suitable trust mechanism. When consumer turnover becomes excessively high, many adaptive agents leave the system before they have accumulated enough experience to fully benefit from the learning process. Similarly, newly arriving agents repeatedly restart the learning process from scratch.

Consequently, the knowledge acquired by adaptable agents cannot be retained and exploited for sufficiently long periods to produce the same advantages observed under more moderate environmental dynamics. This effect reduces the relative benefit of adaptive trust model selection and explains why the superiority observed in Experiment 12 does not persist when environmental volatility becomes more severe.

Overall, these results highlight an important characteristic of adaptive trust model selection: its effectiveness depends not only on the quality of the learning algorithm but also on the stability of the environment in which learning takes place. Moderate environmental dynamics provide sufficient opportunities for learning and adaptation, whereas extremely rapid changes may limit the ability of agents to accumulate and exploit experience effectively.

4.8. Conclusions

This chapter investigated whether trust model selection can be learned dynamically rather than being predefined. Building upon the findings of the previous chapter, which demonstrated that neither FIRE nor CA consistently outperforms the other across all

environmental conditions, the trust model selection problem was formulated as a reinforcement learning task in a dynamic and partially observable Multi-Agent System.

To address this challenge, an adaptable trustor was introduced that is capable of selecting between the FIRE and CA trust models based on its observations of the local environment. Since the underlying environmental factors affecting trust model performance are not directly observable, the adaptable trustor relies on a set of nine environmental features derived from locally available information, including interaction histories, reputation information, neighboring agents, trust assessments, and location data.

The adaptable trustor employs a Deep Q-Network to approximate the optimal trust model selection policy. Through repeated interactions with the environment, the agent learns to associate particular environmental conditions with the trust mechanism that is expected to maximize utility. Unlike conventional trustors that rely permanently on a single trust model, the adaptable trustor can modify its behavior as environmental conditions evolve.

The experimental evaluation considered a broad range of scenarios involving provider population changes, consumer population changes, performance fluctuations, profile switching, mobility, combinations of concurrent environmental changes, and time-varying environmental dynamics. Across all experiments, the adaptable trustor consistently maintained performance above that of the worst-performing trust model, demonstrating its ability to learn an effective trust model selection policy.

The results further showed that the adaptable trustor frequently achieved performance levels close to those of the best-performing trust model and, in some scenarios, exceeded the performance of both FIRE and CA. These findings indicate that adaptive trust model selection enables agents to exploit local environmental conditions more effectively than approaches based on a fixed trust mechanism.

Overall, the results of this chapter demonstrate that reinforcement learning constitutes a viable approach for adaptive trust model selection in dynamic Multi-Agent Systems. By learning from experience and adapting to environmental changes, the proposed adaptable trustor improves robustness and decision-making effectiveness under a wide variety of operating conditions.

4.9. Research Directions

The results presented in this chapter demonstrate the potential of reinforcement learning for adaptive trust model selection. Nevertheless, several opportunities remain for further investigation and improvement.

4.9.1. Improved Exploration Strategies

The Adaptable trustor employs a fixed ε-greedy exploration policy with $\varepsilon = 0.05$ throughout training and execution. While this approach provides a simple mechanism for balancing exploration and exploitation, it may not be optimal throughout the learning process.

During the early stages of learning, extensive exploration is beneficial because the agent has limited knowledge regarding the consequences of its actions. As learning progresses, however, excessive exploration may lead to unnecessary suboptimal decisions. Future research could investigate adaptive exploration strategies, such as decaying ε-greedy policies or alternative exploration mechanisms, to determine whether improved utility gains can be achieved.

4.9.2. Multi-Agent Reinforcement Learning

The proposed approach adopts a single-agent reinforcement learning formulation, where each adaptable trustor learns independently based on its own observations and experiences. While this assumption simplifies the learning problem, it may underestimate the potential benefits of cooperation among agents.

An interesting direction for future work is the extension of the proposed framework to a Multi-Agent Reinforcement Learning (MARL) setting. Through information sharing and collaborative learning, adaptable trustors may be able to identify environmental changes more rapidly and improve trust model selection performance. Although such approaches introduce additional communication and scalability challenges, they may provide significant performance improvements in large-scale systems.

4.9.3. Hyperparameter Optimization and Resource Constraints

The Deep Q-Network architecture and training parameters employed in this work were selected through an informal tuning process. A more systematic hyperparameter optimization procedure could potentially identify configurations that improve learning performance and convergence behavior.

Furthermore, the computational requirements of Deep Q-Learning may be problematic in resource-constrained environments. Future work could investigate the trade-off between learning performance and computational cost, with the goal of identifying lightweight adaptable trust model selection mechanisms suitable for deployment on devices with limited processing capabilities.

4.9.4. Trust Disinformation and Malicious Information Sources

Trust and reputation systems are frequently vulnerable to dishonest or manipulated trust information. False rating, misleading recommendations, and malicious reputation reports can significantly affect trust assessments and decision making.

Although the CA trust model is less dependent on shared trust information than conventional trust models, the broader problem of trust disinformation remains largely unexplored in the context of adaptable trust model selection. Future research could investigate mechanisms that allow adaptable trustors to detect unreliable information sources and learn when different trust models are more resilient to disinformation attacks.

4.9.5. Adaptive Selection Among Multiple Trust Models

The present study considers only two trust mechanisms, namely FIRE and CA. However, the proposed learning framework is not inherently limited to these models.

Future work could investigate adaptive trustors capable of selecting among a larger set of trust and reputation mechanisms. Such an extension would transform trust model selection into a more general decision-making problem and could potentially improve performance across a broader range of environmental conditions.

4.9.6. Real-World Validation and Application Domains

The evaluation presented in this chapter is based exclusively on simulation experiments. While simulation provides a controlled environment for analyzing agent behavior, real-world validation remains an important step toward practical deployment.

Potential application domains include distributed healthcare teams, unmanned aerial vehicle (UAV) swarms, disaster response systems, mobile sensor networks, ad hoc robotic teams, and large-scale service-oriented organizations. In such environments, trust-based decision making often occurs under conditions of uncertainty, incomplete information, and continuous environmental change.

Investigating the applicability of adaptive trust model selection in real-world settings therefore represents an important direction for future research and may provide further insights into the practical benefits and limitations of reinforcement learning-based trust management approaches.

5. Improving CA for Dynamic Trustee Performance Changes

In the previous chapters, two trust models were investigated in detail: FIRE, a representative trustor-centric trust model, and CA, a biologically inspired trustee-centric trust model designed to address several challenges associated with openness and dynamicity.

The comparative evaluation presented in Chapter 3 demonstrated that CA offers several advantages in highly dynamic environments. In particular, the model exhibits strong resilience to changes in the consumer population, avoids the communication overhead associated with recommendation exchange, preserves privacy, and mitigates several forms of trust misinformation by storing trust information locally within service providers. However, the same evaluation also revealed a number of limitations. Most notably, CA was found to be less resilient than FIRE when providers frequently entered and exited the system or when providers changed their performance characteristics over time. Among all environmental changes examined, dynamic modifications of provider behavior, especially profile switching, had the strongest negative impact on the performance of CA.

These findings motivate the central research question addressed in this chapter: Can the CA algorithm be modified in order to improve its ability to cope with dynamic changes in provider performance while preserving its existing advantages?

To answer this question, this chapter begins with a semi-formal analysis of the original CA algorithm. The objective of this analysis is to understand the mechanisms responsible for the reduced performance observed under dynamic provider behavior and to identify opportunities for improvement.

5.1. Semi-Formal Analysis of the Original CA Algorithm

The experimental results presented in Chapter 3 indicated that the CA algorithm performs particularly well in environments characterized by frequent consumer population changes. However, the same evaluation revealed a significant weakness when providers

undergo continuous performance fluctuations or switch between different performance profiles.

To understand the reasons behind this behavior, a semi-formal analysis of the original CA algorithm was conducted. The objective of this analysis is not to provide a mathematical proof of correctness, but rather to examine how the trust update mechanism behaves under different task difficulties and provider performance levels. By analyzing the evolution of connection weights over time, it becomes possible to identify situations in which providers continue undertaking tasks despite exhibiting persistently poor performance.

Recall that in the original CA algorithm, each provider maintains a trust weight for every task-specific connection. Following task execution, the weight is updated according to the observed outcome. Successful executions strengthen the connection, while unsuccessful executions weaken it. A provider undertakes a task only if the corresponding connection weight remains above a predefined threshold.

Consequently, a provider stops attempting a task only after a sufficient number of unsuccessful executions causes the connection weight to fall below the threshold. This mechanism implicitly assumes that poor performance will eventually be reflected in the connection weight and therefore lead to the provider's self-exclusion from future task executions.

The critical question is how quickly this process occurs when a provider's actual performance deteriorates. If the connection weight remains above the threshold for a prolonged period, the provider may continue accepting tasks despite being unable to perform them satisfactorily. Such behavior can generate repeated negative outcomes for consumers and reduce the overall effectiveness of the trust model.

To investigate this issue, the following analysis examines the evolution of connection weights for providers with different performance capabilities under tasks of varying difficulty levels.

5.1.1. Analysis under Different Task Difficulty Levels

While this analysis incorporates formal elements such as assumptions, definitions, propositions, and a proof by induction, it does not constitute a strictly formal analysis in the traditional mathematical or computer science sense. Instead, it represents a blend of formal reasoning and empirical evaluation for the following reasons:

1. Although induction is used to support Proposition 1, many conclusions rely more on empirical observation than on rigorous formal logic,
2. The conclusions are drawn from specific experimental conditions (e.g., Threshold = 0.5, $\alpha = \beta = 0.1$), whereas a purely formal analysis would typically strive to derive results that hold regardless of particular parameter values,
3. Several conclusions are derived from example scenarios rather than universally valid logical proofs.

In our testbed, each consumer requesting the service at a certain performance level waits for a period equal to WT for the service to be provided by a provider in its operational range (a nearby provider). If the requested task = (service_ID, performance_level), where $performance_level \in \{PERFECT, GOOD, OK, BAD, WORST\}$ is not performed, either because there are no nearby providers or none of the nearby providers are willing to perform the task, then the consumer requests the service at the next lower performance level, assuming that any consumer can manage with a lower-quality service.

Assumption 1. *WT is large enough so that if there is a capable provider in the operational range of the consumer, then the task will be successfully executed*.

Assumption 2. *The reason why a nearby provider may not be willing to perform a task is only because the weight of the relevant connection is less than the threshold value. We assume that there are no other reasons, i.e., the providers are not selfish or malicious. In other words, we assume that the providers are always honest and comply with the implemented protocols, having no intention to harm the trustors*.

Since every rational consumer aims to maximize its profit, the sequence of tasks that it may request until a provider is found to provide the service is as follows: $task_1$, $task_2$, $task_3$, $task_4$, and $task_5$. In Table 11, the requirements of each task are specified.

Table 11. Explanation of requirements.

Tasks	Explanation of the Requirement
$task_1 = (service_ID, performance_level = PERFECT)$	Performance must be equal to 10
$task_2 = (service_ID, performance_level = GOOD)$	Performance must be >= 5

$task_3 = (service_ID, performance_level = OK)$	Performance must be >= 0
$task_4 = (service_ID, performance_level = BAD)$	Performance must be >= -5
$task_5 = (service_ID, performance_level = WORST)$	Performance must be >= -10

Definition 1. *Given that C_j is a consumer requesting the successful execution of $task_i$, $i \in \{1, \cdots, 5\}$, P_k is a provider in the operational range of C_j receiving a request from C_j for the execution of $task_i$, and P_k does not yet have a connection for C_j and $task_i$, we define P_k as having learned its incapability to perform $task_i$ (meaning that it cannot always perform $task_i$ successfully) when it initializes the weight of a new connection for P_k and $task_i$ to a value smaller than the threshold value, which will result in not executing $task_i$ for C_j.*

Since only bad and intermittent providers have negative (causing damage to the consumers) performances, we analyze how bad providers learn their own capabilities in each of the five tasks, aiming to identify ways to reduce task executions with negative performances and thus improve the performance of the CA algorithm.

We consider a system of four consumers, C_1, C_2, C_3, and C_4, and only one provider: the bad provider BP_1. Each consumer has in its operational range the provider BP_1. The provider has no knowledge of its capabilities in performing tasks, meaning that it has not formed any connections yet.

5.1.1.1. Phase A: Provider BP_1 learns its incapability in performing $task_1$.

Assume that C_1 requests the execution of $task_1$ broadcasting the message $m_1 = (request, C_1, task_1)$. According to the CA algorithm, when BP_1 receives message m_1, it will create the connection $co_1 = (BP_1, C_1, 0.5, task_1)$, initializing its weight to the value 0.5. Since the condition $w \geq Threshold$ is satisfied, BP_1will perform $task_1$, but it will fail because its performance range is in $[-10,0]$, and $task_1$ requires a performance equal to 10. After it fails, BP_1will decrease the weight of $co_1$to the value $w' = 0.45$, by using equation $w' = w - \beta(1 - w)$, where $\beta = 0.1$ in our experiments.

Now, let C_2 require the execution of $task_1$ by sending the message $m_2 = (request, C_2, task_1)$. When BP_1 receives the message m_2 and because it already has the connection co_1, it will create the connection $co_2 = (BP_1, C_2, 0.45, task_1)$, initializing its weight to the average of the weights of the connections it has with other consumers for $task_1$. In this case, $average = \frac{w_{co_1}}{1} = 0.45$, where w_{co_1} denotes the weight of connection co_1. Because the weight of co_2 is less than the threshold value, BP_1 will decide not to perform $task_1$ for consumer C_2.

Assumption 3. *For simplicity, we assume that BP_1 will not change its ability to provide $task_1$ so that the weights of all connections will remain constant over time.*

Now, we can prove by induction the following proposition.

Proposition 1. *Given Assumption 3, every new connection that BP_1 will create for $task_1$ will be initialized to the average of the weights of the connections it has already created for $task_1$, which will remain constant and equal to 0.45.*

The proof is provided in Appendix C. By Proposition 1 and Definition 1, it follows that BP_1, in a single trial, has learned that it cannot successfully perform $task_1$. We can generalize every bad provider to the following conclusion.

Conclusion 1. *Given our experimental conditions (i.e., Threshold = 0.5, $a = \beta = 0.1$), every bad provider needs only one trial to learn that it cannot successfully perform $task_1$.*

5.1.1.2. Phase B: Provider BP_1 learns its incapability in performing $task_2$.

Following the same analysis as the Phase A, we can conclude as follows.

Conclusion 2. *Given our experimental conditions (i.e., Threshold = 0.5, $a = \beta = 0.1$), every bad provider needs only one trial to learn that it cannot successfully perform $task_2$.*

5.1.1.3. Phase C: Provider BP_1 learns its incapability in performing $task_3$.

Assume that C_1 requests the execution of $task_3$ broadcasting the message $m_1 = (request, C_1, task_3)$. When BP_1 receives message m_1, it will create the connection $co_1 = (BP_1, C_1, 0.5, task_3)$, initializing its weight to the value 0.5. Since the condition $w \geq Threshold$ is satisfied, BP_1 will perform $task_3$. Due to its performance range in $[-10, 0]$ and the task's requirement that performance must be greater than or equal to 0 in order to be successful, BP_1 has a small probability of having a performance equal to 0 and thus a successful execution of $task_3$.

So, consider a scenario where the performance of BP_1 on $task_3$ on its first trial is 0. After it succeeds, it will increase the weight of $co_1$to the value $w' = w + \alpha \cdot (1 - w)$, where $\alpha = 0.1$ in our experiments.

Now, let C_2 request the execution of $task_3$ broadcasting the message $m_2 = (request, C_2, task_3)$. When BP_1 receives message m_2 and because it already has the connection co_1, it will calculate the average weight of existing connections as $average = \frac{w_{co_1}}{1} = 0.55$. Then, it will create the connection $co_2 = (BP_1, C_2, 0.55, task_3)$, initializing its weight to the average just calculated. Since the condition $w \geq Threshold$ is satisfied, BP_1 will execute $task_3$ for C_2. Suppose that BP_1 fails this time and decreases the weight of co_2 to the value $w' = 0.505$. The scenario continues with two consecutive failed executions of $task_3$ for consumers C_3 and C_4. In Table 12, we can see the average weights of the connections after each task execution.

Table 12. Evolution of average connection weights after each execution of $\boldsymbol{task_3}$ for consumer $\boldsymbol{C_1}$ to $\boldsymbol{C_4}$.

Consumer	Result	Final Weight of the New Connection	Average Weight
C_1	success	$w_{co_1} = 0.55$	0.55
C_2	failure	$w_{co_2} = 0.505$	0.5275
C_3	failure	$w_{co_3} = 0.48025$	0.51175
C_4	failure	$w_{co_4} = 0.462925$	0.499544

In the scenario above, BP_1needed three consecutive failed executions after the successful first execution of $task_3$ to learn (according to Definition 1) that it is not capable of always performing this task in a successful way, because the average weight of its connections for $task_3$ is now less than the threshold value. This leads us to the following more general conclusion.

Conclusion 3. *The successful execution of $task_3$ on the first trial of a bad provider requires a number of consecutive failed executions to learn that it cannot always execute this task successfully*.

5.1.1.4. Phase D: Provider BP_1 learns its incapability in performing $task_4$.

Since BP_1 has a performance range in $[-10, 0]$ and $task_4$ requires a $performance \geq -5$ to be successful, BP_1 has a good probability of being successful in its first trial of $task_4$. If we repeat the analysis of phase C, we will be led to the following conclusion.

Conclusion 4. *The successful execution of* $task_4$ *on the first trial of a bad provider requires a number of consecutive failed executions to learn that it cannot always execute this task successfully*.

5.1.1.5. Phase E: Provider BP_1 learns its capability in performing $task_5$.

Since BP_1 has a performance range in $[-10, 0]$ and $task_5$ is successfully executed if $performance \geq -10$, BP_1 will always execute this task successfully.

Conclusion 5. *A bad provider will always execute* $task_5$.

Ideally, we would prefer that a bad provider not provide the service at all, because its poor performance harms the consumer, but providing the service at least once is required to assess the provider's capabilities. However, the previous analysis demonstrates that the CA algorithm allows the bad provider to provide the service multiple times. Despite its negative performance on $task_1$, BP_1 performed poorly with a negative performance in phase B as well. A more intelligent agent could consider its negative performance from the first time and decide not to execute $task_2$. Furthermore, in both phase C and phase D, the successful first execution of the task requires a series of consecutive failed executions, which we would like to avoid.

To this end, in the following section, we propose an improved CA algorithm designed to detect bad providers early on and prevent them from damaging the consumers with their negative performances.

5.2. The Proposed CA Algorithm for the Early Detection of Low-Performance Providers

The semi-formal analysis presented in the previous section revealed an important limitation of the original CA algorithm. Although connection weights are eventually adjusted to reflect a provider's capability, this adaptation may occur too slowly when providers experience a deterioration in their performance. As a result, providers may continue undertaking tasks for which they are no longer suitable, generating repeated

unsuccessful executions before the corresponding connection weights fall below the execution threshold.

To address this limitation, an enhanced version of the CA algorithm is proposed. The main objective of the modification is to enable providers to detect harmful performance degradation immediately after task execution rather than relying exclusively on the gradual adaptation of connection weights. More specifically, the proposed approach introduces a task-specific self-assessment mechanism that allows providers to continuously monitor their own performance and incorporate this information into future decision making.

Each provider maintains a task-specific self-assessment mechanism through the $i.bad_tasks$ data structure, which is initialized with a default value of false for each task (line 1). This means that when a provider is created, it implicitly assumes that it is not bad for any task it has not yet performed – there is no need to explicitly initialize entries for unknown tasks. After performing a task, the provider re-evaluates its performance (lines 30-34). If the performance is less than or equal to zero, it considers itself a bad provider for that specific task by setting $i.bad_tasks[task] \leftarrow true$; otherwise, it maintains or resets the value to false. This enables rapid, task-specific reassessment, allowing the provider to dynamically adapt its trustworthiness profile based on its most recent outcomes.

In lines 7-14, this self-assessment is used to initialize the trust weight of a new connection. If the provider believes it is bad ($i.bad_tasks[task] = true$) and the task requires a performance level of PERFECT, GOOD, or OK, the connection is initialized with a trust weight of 0.45 (line 9). Otherwise, the algorithm falls back to the original initialization logic (lines 10-15), which either averages existing weights from similar connections or sets a default trust of 0.5.

In lines 16-20, if the connection already exists, the provider re-evaluates whether it should adjust the weight based on its current self-assessment. If the provider considers itself bad and the task requires a performance level (i.e., in {PERFECT, GOOD, OK}, it lowers the weight of the existing connection to 0.45, reinforcing a cautious stance toward engaging in the task.

This modification enhances the context-awareness and reliability of the decision-making process by ensuring that each provider is consistently informed by its own recent experience when evaluating and responding to task requests.

Core functionalities from Algorithm 2, presented in Chapter 3 – such as broadcasting task requests, handling incoming messages, task selection and execution, connection weight updates, and dynamic profile adjustments – are preserved without modification.

The complete pseudocode of the enhanced algorithm is presented in Algorithm 3.

```
Algorithm 3 CA for the Early Detection of Bad Providers

# --- Initialize task-specific self-assessment memory ---
1:  define i.bad_tasks as a map with default value false  # assumes good unless
proven otherwise
2:  while True do

# --- Task perception and request broadcasting ---
3:      when a new task task = (c, r) is perceived do
4:          broadcast message m = (request, i, task)

# --- Request reception and connection initialization ---
5:      when message m = (request, j, task) is received do
6:          add m to message list M

7:          if no existing connection Co = (j, w, task) exists then
8:              if i.bad_tasks[task] = true and task.performance_level ∈ {PERFECT,
GOOD, OK} then
9:                  create Co = (j, 0.45, task)  # cautious trust level for
task-specific bad assessment
10:             else if ∃ similar connections Co' = (k, w', task) for other
trustors k then
11:                    avg_w ← average weight of all Co' for task
12:                 create Co = (j, avg_w, task)
13:             else
14:                 create Co = (j, 0.5, task)  # initialize to default trust
15:             end if
16:         else  #if connection Co = (j, w, task) exists

# --- modify an existing connection if certain conditions hold---
17:             if i.bad_tasks[task] = true and task.performance_level ∈
{PERFECT, GOOD, OK} then
18:                 modify Co = (j, 0.45, task)
19:             end if
20:         end if

# --- Task selection ---
21:     select m = (request, j, task) from M such that Co = (j, w, task) has the
highest trust value w among all stored connections

# --- Execution decision ---
22:     if task not completed and w ≥ θ then

23:         performance ← performTask(task)
24:         feedback ← rewardPunishmentEvaluation(performance,task)

# --- Update connection weight based on performance ---
25:         if feedback = reward then
26:             w ← min(1, w + α(1 − w))
27:         else
28:             w ← max(0, w − β(1 − w))
29:         end if

# --- Re-evaluate task-specific self-assessment based on latest performance ---
30:         if performance ≤ 0 then
31:             i.bad_tasks[task] ← true
32:         else
33:             i.bad_tasks[task] ← false
34:         end if

35:     end if

# --- Remove processed request ---
36:     delete m from M
```

```
# --- Dynamic trustee profile adaptation ---
37:     for all connections Co' = (j, w', task') where w' < θ do
38:         if task' = (c, r') and r' > r then
39:             if performance ≥ minSuccessfulPerformance(task') then
40:                 w' ← θ   # re-enable previously rejected tasks
41:             end if
42:         end if
43:     end for

44: end while
```

5.3. Experimental Methodology

The experimental evaluation presented in this chapter is conducted using the same simulation environment, provider population configuration, performance measure, and statistical analysis procedure described in Chapter 3. Consequently, all simulation parameters, trust model parameters, and testbed characteristics remain unchanged unless otherwise stated.

The objective of the present evaluation is to investigate whether the proposed modification to the CA algorithm improves its ability to cope with dynamic changes in provider performance. More specifically, the experiments examine whether the introduction of the self-assessment mechanism enables providers to detect harmful performance degradation earlier and consequently improve the utility obtained by consumers.

5.3.1. Consumer Groups

The evaluation compares the performance of three groups of consumer agents:

- FIRE: consumer agents that use the FIRE trust model.
- CA_OLD: consumer agents that use the original version of the CA algorithm presented in Chapter 3.
- CA_NEW: consumer agents that use the enhanced version of the CA algorithm proposed in this chapter.

The comparison between CA_OLD and CA_NEW enables a direct assessment of the proposed modification, while the comparison with FIRE provides an additional benchmark based on a well-established trust model for open multi-agent systems.

5.3.2. Statistical Evaluation

As in the previous chapters, each experiment is executed multiple times using independent simulation runs in order to reduce the influence of random fluctuations. The exact Number of Independent Simulation Runs (NISR) varies between experiments and is reported in the corresponding figures.

The effectiveness of each algorithm is measured using the average Utility Gain (UG) achieved by consumer agents throughout the simulations. After all simulation runs have been completed, the average UG values are compared using a two-sample t-test with a 95% confidence level. Statistical comparisons are performed between:

- CA_OLD and CA_NEW,
- FIRE and CA_NEW.

As in Chapter 3, the experimental results are presented using two-axis graphs. The left y-axis shows the average Utility Gain achieved by each consumer group, while the right y-axis presents the performance ranking derived from the statistical comparison. Consumer groups assigned the same rank exhibit no statistically significant difference in performance at the 95% confidence level.

5.4. Experimental Scenarios and Results

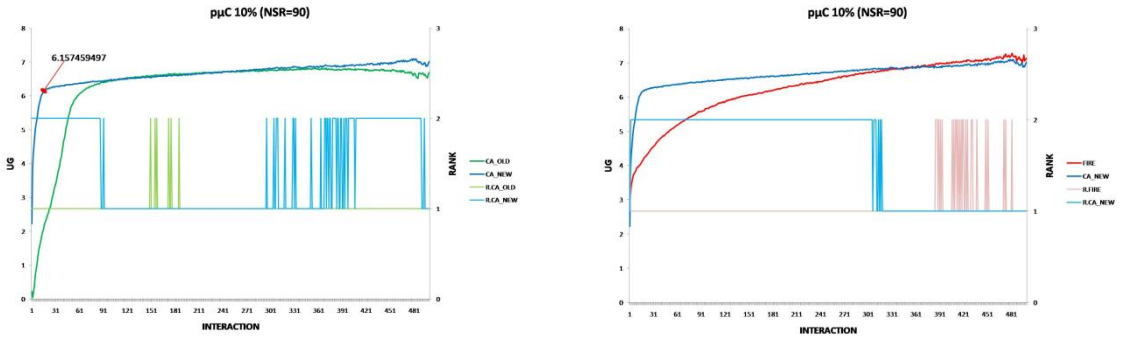
This section presents the results of fourteen experiments evaluating the performance of the enhanced CA algorithm (CA_NEW) in comparison to both CA_OLD and FIRE. Each subsection examines different environmental conditions.

The first two experiments (Section 5.4.1) investigate scenarios in which service provider performance fluctuates over time. Experiment 3 (Section 5.4.2) is conducted in a static environment. Experiments 4-8 (Section 5.4.3) investigate the impact of provider population changes, with the proportion of newly introduced providers increasing gradually up to 30%. Experiments 9-11 (Section 5.4.4) examine the effects of a consumer population change of up to 10%. Experiments 12 and 13 (Section 5.4.5) explore changes in the locations of consumers and providers. Experiment 14 (Section 5.4.6) evaluates performance under conditions where all dynamic factors vary simultaneously. Section 5.4.7 summarizes the findings across all experiments.

5.4.1. The Performance of the New CA Algorithm in Dynamic Trustee Profiles

This section presents the results of two experiments in which we compare the performance of the enhanced CA algorithm (CA_NEW) with the performance of the original version (CA_OLD) and that of FIRE, when the service providers' performance varies over time.

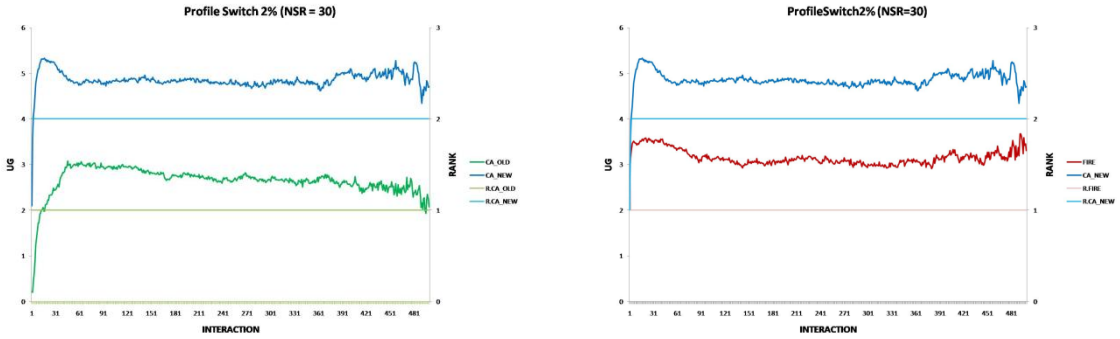
Experiment 1. A provider may alter its average performance at a maximum of 1.0 UG unit with a probability of 10% in every round $(p_{\mu C} = 0.10, M = 1.0)$. The results are shown in Figure 35. Figure 35a shows that CA_NEW generally outperforms CA_OLD, with the most significant improvement observed in the initial interactions. Figure 35b shows that CA_NEW, with the exception of the first interaction, outperforms FIRE in the initial interactions. However, FIRE adapts as the number of interactions increases and eventually achieves slightly better performance.



(**a**) (**b**)

Figure 35. (a) A performance comparison of CA_NEW and CA_OLD, when providers change performance with 10% probability per round: CA_NEW achieves better performance, especially in early interactions. (b) A performance comparison of CA_NEW and FIRE, when providers change performance with 10% probability per round: CA_NEW excels in early interactions (except for the first interaction), while FIRE adapts and eventually achieves slightly better performance.

Experiment 2. A provider may switch to a different performance profile with a probability of 2% in every round $(p_{ProfileSwitch} = 0.02)$. The results are depicted in Figure 36. Figure 36a demonstrates that CA_NEW consistently outperforms CA_OLD across all interactions, with an average difference of approximately 2 UG units. Figure 36b shows that CA_NEW performs better than FIRE in all interactions except the first one.

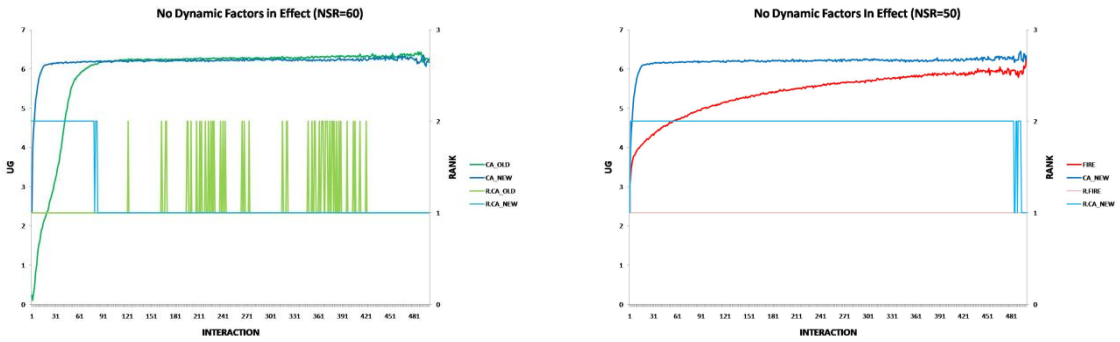


(a) **(b)**

Figure 36. (a) A performance comparison of CA_NEW and CA_OLD when providers switch performance profiles with 2% probability per round: CA_NEW consistently outperforms CA_OLD by an average of approximately 2UG units. (b) A performance comparison of CA_NEW and FIRE when providers switch performance profiles with 2% probability per round: CA_NEW outperforms FIRE in all interactions except for the first one.

5.4.2. The performance of the New CA Algorithm in the Static Setting

This section presents the results of **Experiment 3**, which evaluates the performance of the enhanced CA algorithm (CA_NEW) in a static environment without any dynamic factors. The findings are depicted in Figure 37. Figure 37a demonstrates that CA_NEW performs better than CA_OLD in the initial interactions. Figure 37b shows that CA_NEW surpasses FIRE in all interactions, except for the first one.



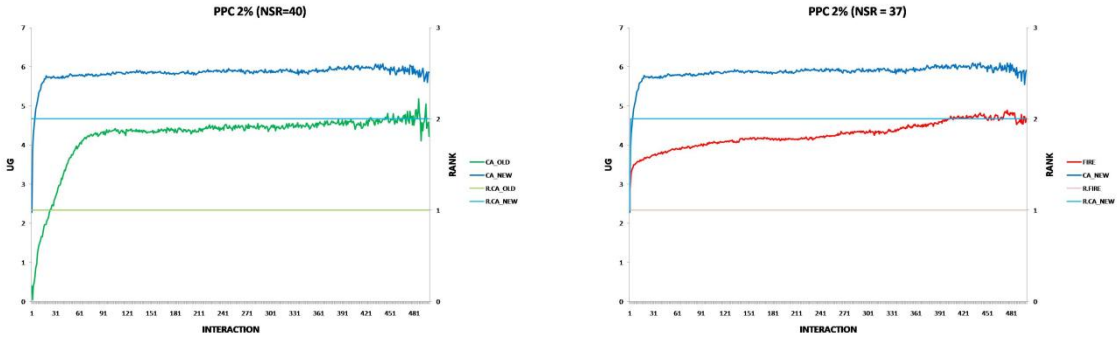
(a) **(b)**

Figure 37. (a) A performance comparison of CA_NEW and CA_OLD in a static environment: CA_NEW achieves better performance in early interactions. (b) A performance comparison of CA_NEW and FIRE in a static environment: CA_NEW outperforms FIRE in all interactions except for the first interaction.

5.4.3. The Performance of the New CA Algorithm in Provider Population Changes

This section evaluates the performance of the enhanced CA algorithm (CA_NEW) under conditions where the provider population gradually fluctuates by up to 30%, through a series of five experiments:

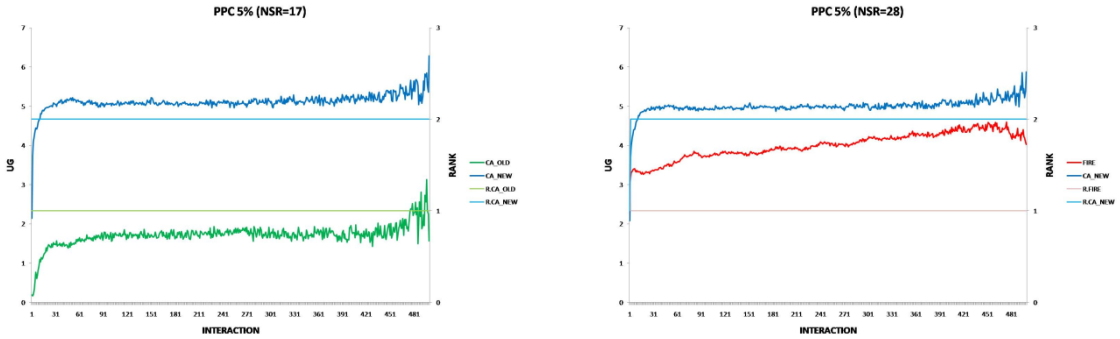
Experiment 4. The provider population changes at a maximum of 2% in every round ($p_{PPC} = 0.02$). The results are shown in Figure 38.



(a) **(b)**

Figure 38. (a) A performance comparison of CA_NEW and CA_OLD when the provider population changes with a 2% probability per round: CA_NEW outperforms CA_OLD in all interactions. (b) A performance comparison of CA_NEW and FIRE when the provider population changes with a 2% probability per round.

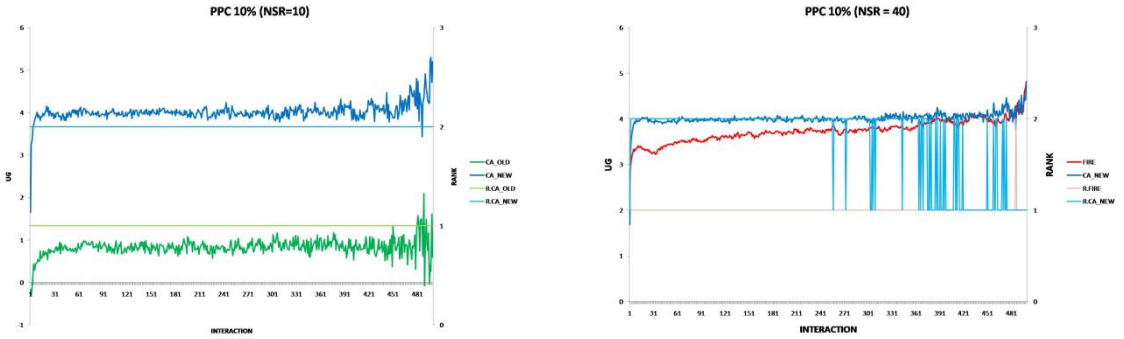
Experiment 5. The provider population changes by a maximum of 5% in every round ($p_{PPC} = 0.05$). The results are shown in Figure 39.



(a) **(b)**

Figure 39. (a) A performance comparison of CA_NEW and CA_OLD when the provider population changes with a 5% probability per round: CA_NEW outperforms CA_OLD in all interactions. (b) A performance comparison of CA_NEW and FIRE when the provider population changes with 5% probability per round.

Experiment 6. The provider population changes by a maximum of 10% in every round ($p_{PPC} = 0.10$). The results are shown in Figure 40.



(**a**) (**b**)

Figure 40. (a) A performance comparison of CA_NEW and CA_OLD when the provider population changes with a 10% probability per round: CA_NEW outperforms CA_OLD in all interactions. (b) A performance comparison of CA_NEW and FIRE when the provider population changes with a 10% probability per round.

Experiment 7. The provider population changes by a maximum of 20% in every round ($p_{PPC} = 0.20$). The results are shown in Figure 41.

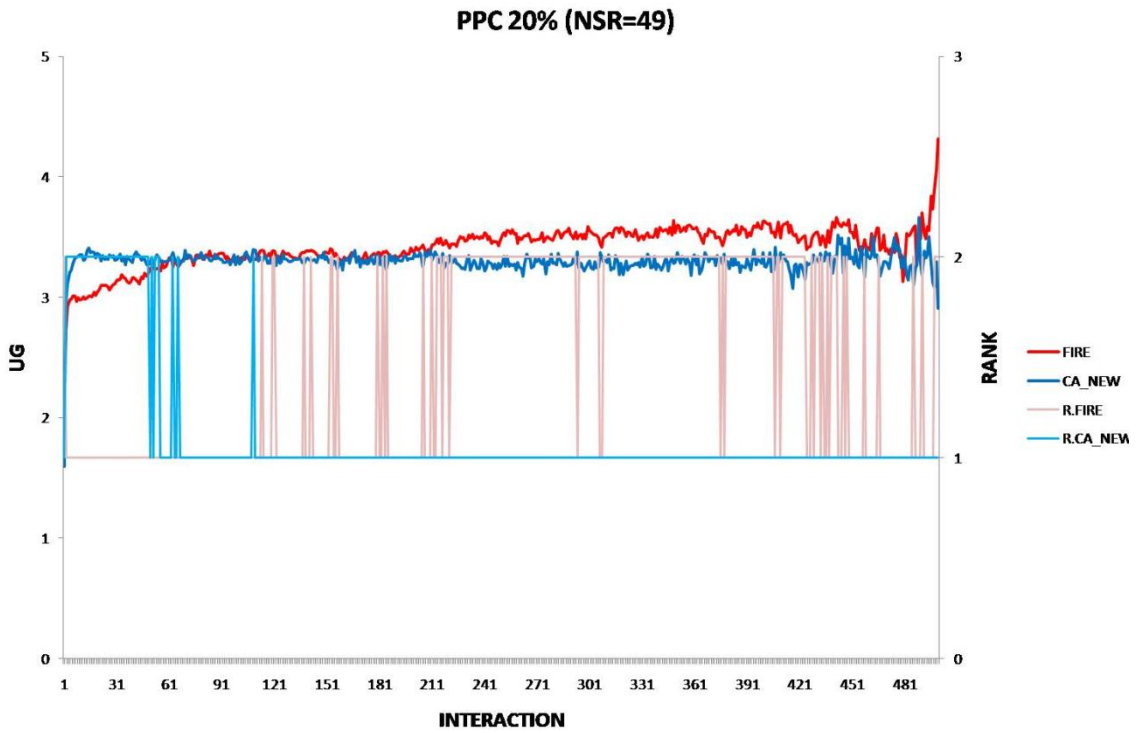


Figure 41. A performance comparison of CA_NEW and FIRE when the provider population changes with a 20% probability per round.

Experiment 8. The provider population changes by a maximum of 30% in every round ($p_{PPC} = 0.30$). The results are shown in Figure 42.

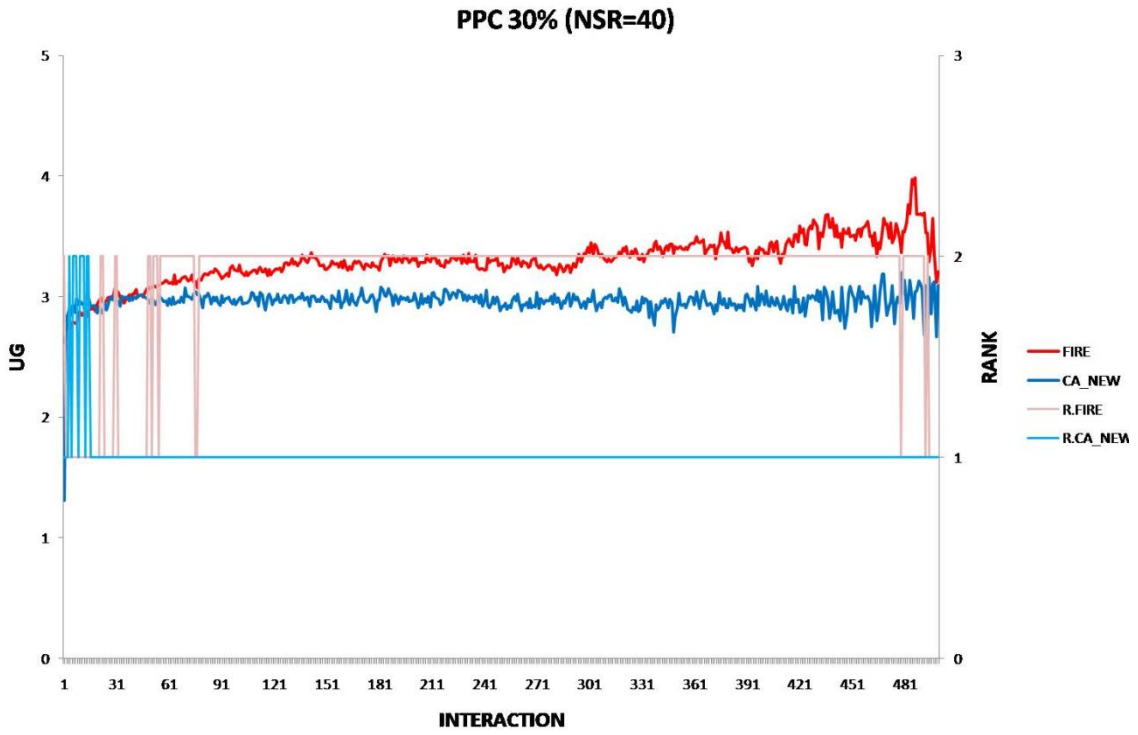


Figure 42. A performance comparison of CA_NEW and FIRE when the provider population changes with a 30% probability per round.

In Experiments 4-6, we compare the performance of CA_NEW to both CA_OLD and FIRE. Figures 38a, 39a and 40a show that CA_NEW significantly outperforms CA_OLD, achieving higher UG across all interactions. Figures 38b, 39b and 40b reveal that as the provider population change rate rises from 2% to 10%, CA_NEW maintains better performance than FIRE.

Figure 43a and 43b demonstrates that CA_NEW is more resilient than CA_OLD to changes in the provider population. Specifically, when the provider population change rate increases from 2% to 10%, CA_OLD's performance drops by 4 UG units (Figure 43a), whereas CA_NEW's performance drops by only 2 UG units (Figure 43b).

Figure 43b and 43c indicate that CA_NEW's performance drops more sharply than FIRE's, suggesting that FIRE is more resilient to this environmental change. This trend raises the expectation that FIRE will eventually surpass CA_NEW at a higher provider population change rate. To verify this hypothesis, we conducted Experiments 7 and 8. The results, shown in Figures 41 and 42, confirm that when the provider population change rate reaches 30%, FIRE outperforms CA_NEW.

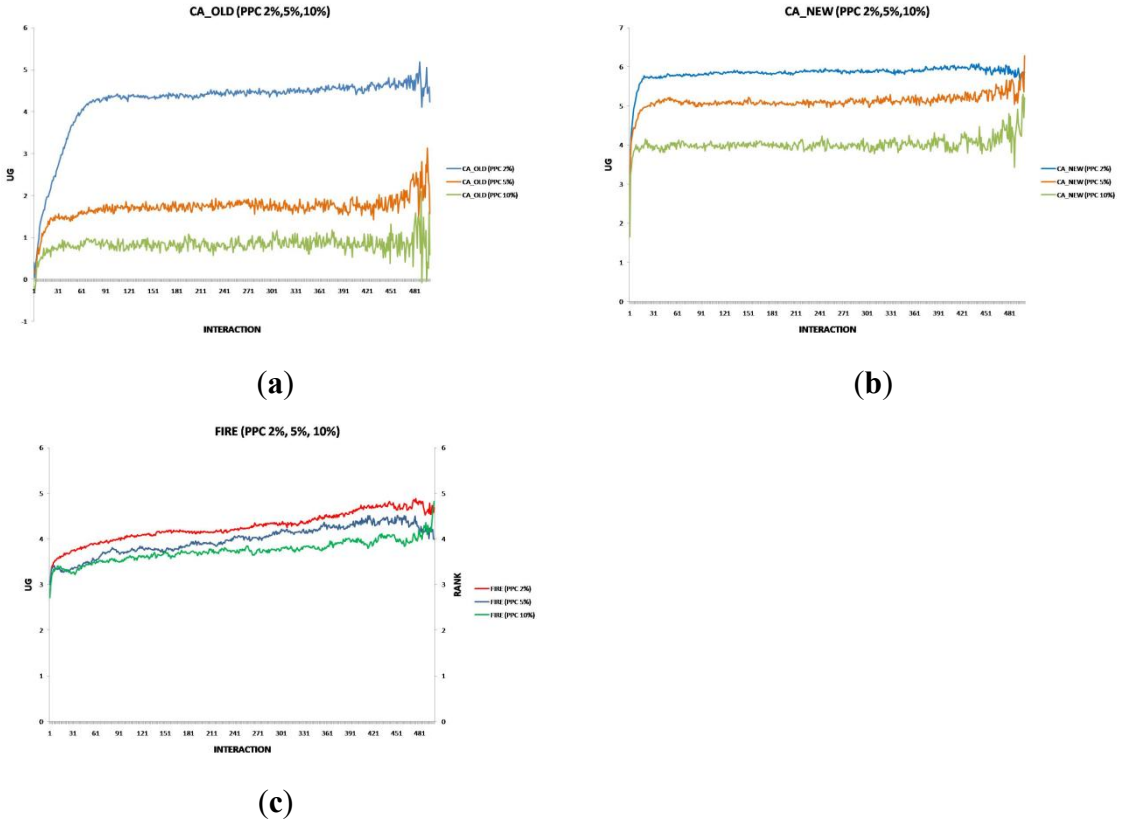


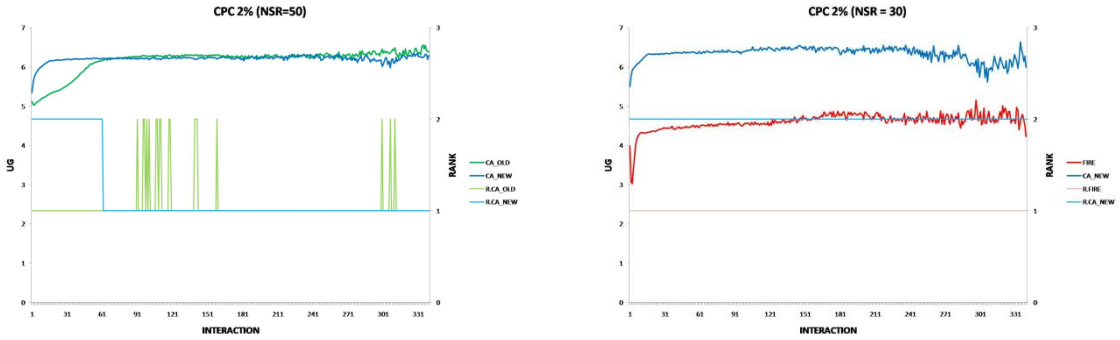
Figure 43. (a) Performance comparison of CA_OLD when the provider population changes with probabilities of 2%, 5%, and 10%; performance drops by 4 UG units. (b) Performance comparison of CA_NEW when provider population changes with probabilities of 2%, 5%, and 10%; performance drops by 2 UG units. (c) Performance comparison of FIRE when provider population changes with probabilities of 2%, 5%, and 10%; FIRE is more resilient than CA_NEW to this environmental change.

FIRE demonstrates greater resilience to changes in the provider population compared to CA_NEW. This is primarily due to FIRE's continuous adaptation over time, which enables it to maintain or even improve its performance. In these experiments, consumers rely on witness reputation, which gradually increases as witnesses (other consumers) remain in the system, contributing to FIRE's stability. In contrast, CA heavily depends on the provider's knowledge of its own capabilities, and when the provider population changes, newcomers must learn from scratch, leading to a more significant decline in performance. While CA_NEW shows an improvement over CA_OLD in scenarios involving provider population change, the fundamental differences in the underlying mechanisms still allow FIRE to outperform CA_NEW under these conditions.

5.4.4. The Performance of the New CA Algorithm in Consumer Population Changes

This section evaluates the performance of the enhanced CA algorithm (CA_NEW) in comparison to both CA_OLD and FIRE under conditions where the consumer population gradually fluctuates by up to 10%, through a series of three experiments.

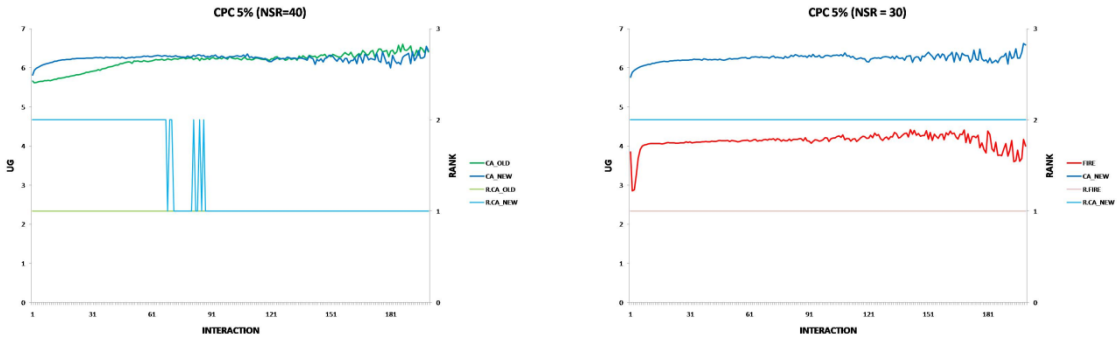
Experiment 9. The consumer population changes at a maximum of 2% in every round ($p_{CPC} = 0.02$). The results are shown in Figure 44.



(a) **(b)**

Figure 44. (a) Performance comparison of CA_NEW and CA_OLD when consumer population changes with a probability of 2% per round. (b) Performance comparison of CA_NEW and FIRE when consumer population changes with a probability of 2% per round.

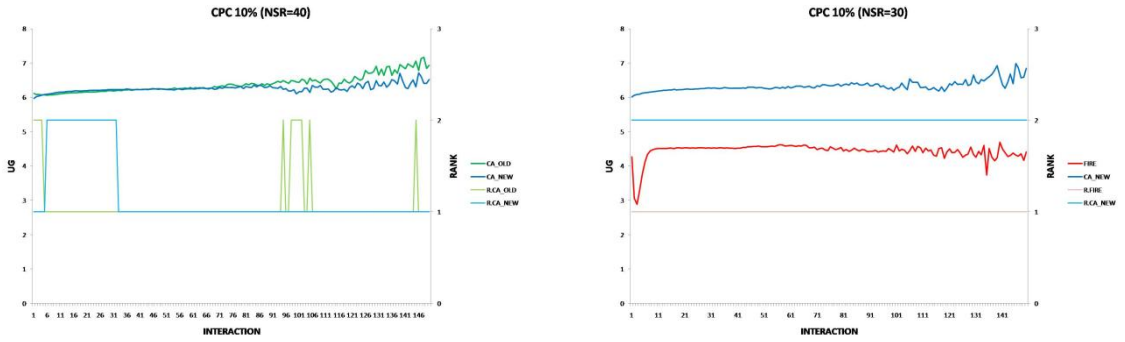
Experiment 10. The consumer population changes at a maximum of 5% in every round ($p_{CPC} = 0.05$). The results are shown in Figure 45.



(a) **(b)**

Figure 45. (a) Performance comparison of CA_NEW and CA_OLD when the consumer population changes with a probability of 5% per round. (b) Performance comparison of CA_NEW and FIRE when the consumer population changes with a probability of 5% per round.

Experiment 11. The consumer population changes at a maximum of 10% in every round ($p_{CPC} = 0.10$). The results are shown in Figure 46.

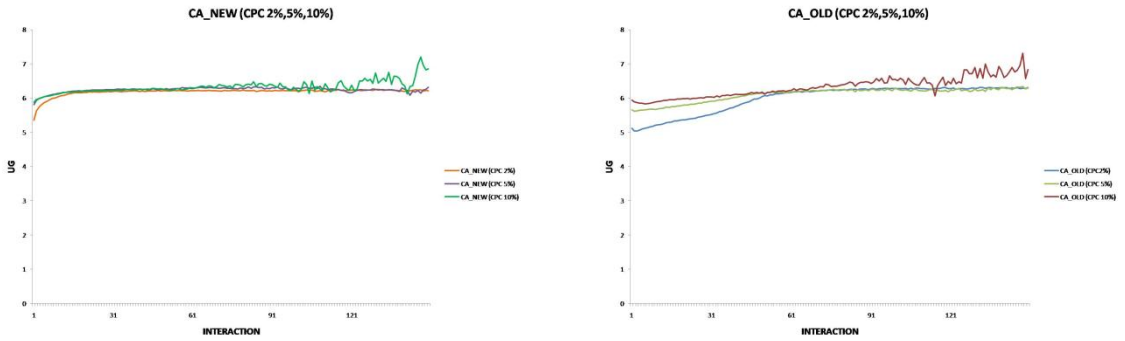


(a) **(b)**

Figure 46. (a) Performance comparison of CA_NEW and CA_OLD when the consumer population changes with a probability of 10% per round. (b) Performance comparison of CA_NEW and FIRE when the consumer population changes with a probability of 10% per round.

Figures 44a, 45a, and 46a illustrate that CA_NEW outperforms CA_OLD, generally achieving a higher UG in the initial interactions. Figures 44b, 45b and 46b demonstrate that CA_NEW consistently performs better than FIRE throughout all interactions in this environmental change.

Figure 47a, b show that in the first interactions, both CA_OLD and CA_NEW improve their performance as the consumer population change rate increases from 2% to 10%. Nevertheless, Figure 47a reveals that when CPC = 10%, CA_OLD slightly surpasses CA_NEW in the first interaction. A possible explanation for this is that when CPC = 10%, the average UG of the first interaction is influenced by a larger number of newcomer agents who have their first interaction in later simulation rounds. During these rounds, service providers have established more connections and can more accurately evaluate their ability to provide the service using CA_OLD, resulting in a higher UG for service consumers. This observation led us to hypothesize that CA_OLD, compared to CA_NEW, is a more suitable choice for service providers that have remained in the system longer and have gained more knowledge about their service-providing capabilities, assuming their capabilities remain unchanged over time.



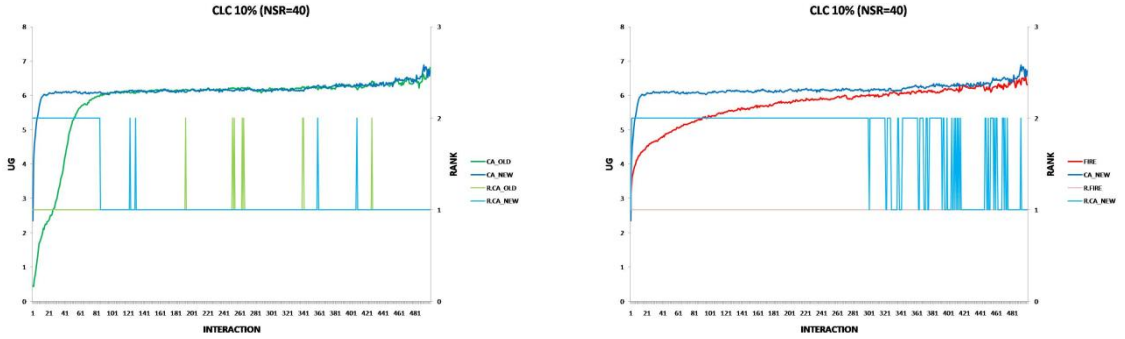
(a) **(b)**

Figure 47. (a) Performance comparison of CA_NEW in consumer population changes of 2%, 5%, and 10%. (b) Performance comparison of CA_OLD in consumer population changes of 2%, 5%, and 10%.

5.4.5. The Performance of the New CA Algorithm in Consumer and Provider Location Changes

In this section, we evaluate the performance of the enhanced CA algorithm (CA_NEW) in comparison to both CA_OLD and FIRE under conditions where consumers and providers change locations, by conducting the following two experiments.

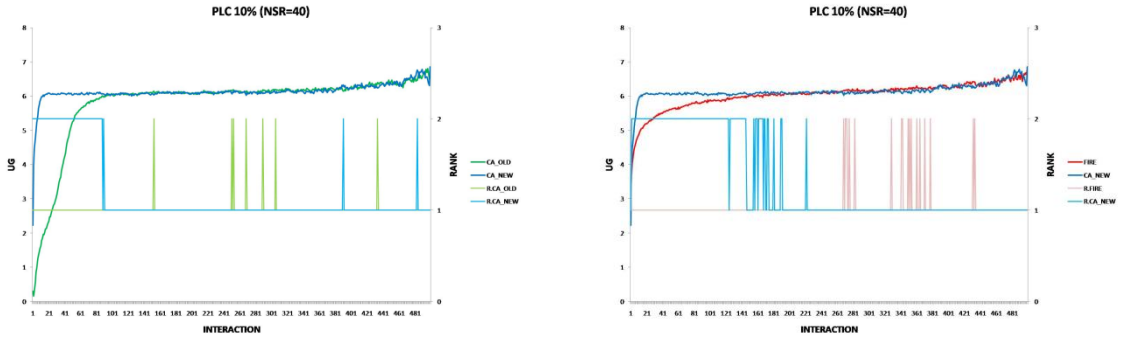
Experiment 12. A consumer may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($p_{CLC} = 0.10, \Delta\Phi = \pi/20$). The results are shown in Figure 48.



(a) **(b)**

Figure 48. (a) A performance comparison of CA_NEW and CA_OLD when consumers change locations with a probability of 10% per round. (b) A performance comparison of CA_NEW and FIRE when consumers change locations with a probability of 10% per round.

Experiment 13. A provider may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round ($p_{PLC} = 0.10, \Delta\Phi = \pi/20$). The results are shown in Figure 49.



(a) **(b)**

Figure 49. (a) A performance comparison of CA_NEW and CA_OLD when providers change locations with a probability of 10% per round. (b) A performance comparison of CA_NEW and FIRE when providers change locations with a probability of 10% per round.

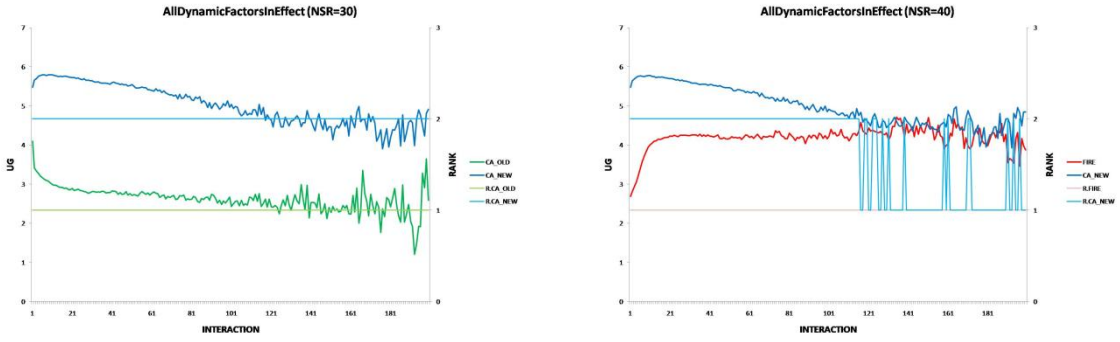
Figures 48a and 49a indicate that CA_NEW demonstrates improved performance in the initial interactions compared to CA_OLD under both environmental changes. Figures 48b and 49b show that CA_NEW outperforms FIRE in both experiments, except for the first interaction, where FIRE performs better.

5.4.6. The Performance of the New CA Algorithm Under the Effects of All Dynamic Factors

In this section, we report the results of the final experiment, which evaluates the performance of CA_NEW under the combined influence of all dynamic factors.

Experiment 14. A provider may alter its average performance at a maximum of 1.0 UG unit with a probability of 10% in every round $(p_{\mu C} = 0.10, M = 1.0)$. A provider may switch into a different performance profile with a probability of 2% in every round $(p_{ProfileSwitch} = 0.02)$. The provider population changes at a maximum of 2% in every round $(p_{PPC} = 0.02)$. The consumer population changes at a maximum of 5% in every round $(p_{CPC} = 0.05)$. A consumer may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round $(p_{CLC} = 0.10, \Delta\Phi = \pi/20)$. A provider may move to a new location on the spherical world at a maximum angular distance of $\pi/20$ with a probability of 0.10 in every round $(p_{PLC} = 0.10, \Delta\Phi = \pi/20)$.

The results, shown in Figure 50, demonstrate that CA_NEW consistently outperforms both CA_OLD and FIRE across all interactions.



(a) **(b)**

Figure 50. (a) Performance comparison of CA_NEW and CA_OLD when all dynamic factors are in effect: CA_NEW outperforms. (b) Performance comparison of CA_NEW and FIRE when all dynamic factors are in effect: CA_NEW outperforms.

5.4.7. An Overview of the Results

The simulation results reveal that the updated CA algorithm (CA_NEW) demonstrates superior performance over its predecessor (CA_OLD) and the FIRE model under various environmental conditions:

1. Dynamic Trustee Profiles: CA_NEW outperforms CA_OLD across all interactions and shows resilience in handling provider performance fluctuations. While FIRE adapts more quickly in some cases, CA_NEW remains very competitive.
2. Static Environment: CA_NEW surpasses CA_OLD in the initial interactions and consistently outperforms FIRE, except in the first interaction.
3. Provider Population Changes: CA_NEW is more resilient than CA_OLD when provider population fluctuations increase up to 10%, maintaining better performance. However, as changes reach 30%, FIRE eventually outperforms CA_NEW, indicating FIRE's resilience in this environmental change.
4. Consumer Population Changes: CA_NEW generally achieves a higher UG than CA_OLD, though at high levels of consumer population changes, CA_OLD performs slightly better in the first interaction, suggesting that CA_OLD may be better for long-term service providers with stable capabilities. CA_NEW consistently outperforms FIRE under consumer population changes.
5. Consumer and Provider Location Changes: CA_NEW shows improved performance over CA_OLD in initial interactions and outperforms FIRE in all interactions, except for the first one.

6. Combined Dynamic Factors: When all dynamic factors are in effect, CA_NEW maintains superior performance over both CA_OLD and FIRE across all interactions, demonstrating its robustness in complex environments.

Overall, CA_NEW is a significant improvement over CA_OLD, with better resilience and adaptability, although there are indications that CA_OLD may be a better choice for long-term service providers with stable behavior. While FIRE exhibits some advantages in extreme environmental changes, CA_NEW remains highly competitive across diverse scenarios. This leads to another direction for future work: to explore how trustees can recognize environmental changes and assess their own self-awareness to determine whether CA_OLD or CA_NEW would yield the best performance. Adopting this comprehensive RL approach could enhance adaptability and effectiveness across a wide range of scenarios.

5.5. Conclusions

This chapter investigated how the *Create Assemblies* (CA) algorithm can be improved to better cope with dynamic changes in trustee performance. The motivation for this work originated from the comparative evaluation presented in Chapter 3, which showed that although CA exhibits several desirable properties in open and dynamic multi-agent systems, its performance deteriorates when providers experience continuous performance changes or switch between different performance profiles. These environmental changes were found to be particularly challenging because providers may continue undertaking tasks even after their actual capability has declined, causing repeated unsuccessful interactions before the corresponding trust weights adapt sufficiently.

To better understand this behavior, a semi-formal analysis of the original CA algorithm was conducted. The analysis revealed that the adaptation mechanism based solely on connection-weight updates may react too slowly when providers experience performance degradation. In particular, providers may continue attempting tasks for which they are no longer suitable, especially when previous successful interactions have established high trust weights.

Based on these observations, an enhanced version of the CA algorithm was proposed. The new approach introduces a task-specific self-assessment mechanism that enables providers to continuously evaluate their own performance and rapidly identify

situations in which they should refrain from undertaking particular tasks. By incorporating self-assessment information into connection initialization and update decisions, providers become capable of reacting immediately to harmful performance changes rather than relying exclusively on the gradual adaptation of trust weights.

The experimental evaluation demonstrated that the proposed modification significantly improves the behavior of the CA model under dynamic conditions. Across the majority of the examined environmental changes, CA_NEW consistently outperformed the original CA_OLD algorithm. The improvement was particularly evident in scenarios involving dynamic trustee profiles, where providers continuously changed their performance characteristics or switched between different performance profiles. These results support the conclusion that the proposed self-assessment mechanism successfully improves the adaptability of the CA model while preserving its original design principles.

The results also showed that no single trust mechanism is universally optimal. Although CA_NEW substantially improves the performance of CA, there remain environmental conditions under which alternative approaches, such as FIRE or even the original CA algorithm, may achieve comparable or superior performance. This observation further reinforces the importance of adaptability in trust management systems operating under dynamic and uncertain conditions.

5.6. Research Directions

The results presented in this chapter suggest several promising directions for future research.

A natural extension of this work is the investigation of adaptive trustee behavior. The proposed CA_NEW algorithm introduces a self-assessment mechanism that allows providers to react more effectively to changes in their own performance. However, the choice between CA_OLD and CA_NEW remains fixed. An interesting research direction would be to develop providers capable of detecting environmental dynamics and autonomously selecting the most appropriate behavioral strategy according to current conditions, integrating machine learning techniques into the trustee side of the trust management process. While Chapter 4 investigated adaptive trustors capable of learning which trust model to employ, future research could explore whether providers can

similarly learn when and how to modify their self-assessment policies in order to maximize long-term utility.

The self-assessment mechanism introduced in this chapter relies on a predefined performance threshold for identifying unsuitable task executions. Future work could investigate more sophisticated approaches that allow providers to dynamically adjust these thresholds according to their experience, environmental conditions, or task characteristics.

The present study focused on performance adaptation under the assumption of willing and honest agents. Future work should investigate how the proposed mechanism behaves in environments containing dishonest, malicious, or strategically deceptive agents. Understanding the interaction between self-assessment mechanisms and adversarial behavior remains an important challenge for trust management in open multi-agent systems.

Finally, the experiments conducted in this chapter were performed in a simulation environment. Although simulation provides a controlled and reproducible means of evaluating trust mechanisms, applying the proposed approach to real-world distributed systems would provide valuable insights into its practical effectiveness. Potential application domains include Internet of Things environments, smart infrastructures, autonomous robotic teams, mobile sensor networks, and other highly dynamic distributed systems where trust assessment remains a fundamental requirement.

6. Towards a Comprehensive Evaluation of the Create Assemblies Trust Model

The previous chapters focused primarily on the design, improvement, and empirical evaluation of the *Create Assemblies* (CA) trust model. Experimental results demonstrated that the proposed approach exhibits several desirable properties in open and dynamic multi-agent systems, including resilience to consumer population changes, reduced communication requirements, and the ability to adapt to changing environmental conditions through the improvements introduced in the previous chapter.

However, the performance of a trust model cannot be assessed solely on the basis of utility gain or task allocation effectiveness. Trust management systems are expected to satisfy a broader set of requirements concerning decentralization, scalability, robustness, privacy preservation, explainability, and other quality characteristics that determine their suitability for deployment in real-world environments.

For this reason, the present chapter provides a comprehensive evaluation of the CA model using a set of thirteen trust management and Quality-of-Trust (QoT) criteria frequently discussed in the literature. These criteria cover both technical and operational aspects of trust systems and provide a broader perspective on the strengths and limitations of the proposed approach.

The objective of this chapter is twofold. First, it seeks to assess the extent to which the CA model satisfies established trust model requirements. Second, it aims to identify limitations and open challenges that can motivate future research. Rather than focusing exclusively on performance, the evaluation considers architectural properties, communication requirements, adaptability, robustness against malicious behavior, privacy implications, and practical deployment concerns.

6.1. Evaluation Criteria

The literature on computational trust and reputation systems proposes numerous properties that can be used to evaluate the quality and suitability of a trust model. While no universally accepted evaluation framework exists, several recurring criteria appear consistently across studies.

In [91], Wang et al. define Quality of Trust (QoT) as the quality of trust models and propose several evaluation criteria including subjectivity, dynamicity, context awareness, privacy preservation, scalability, robustness, overhead, explainability, and user acceptance. However, they highlight the absence of a uniform standard. Indeed, Fotia et al. in [145], focusing on edge-based IoT systems, provide a different set of evaluation criteria (accuracy, security, availability, reliability, Heterogeneity, Lightweightness, flexibility, and scalability), referring to them as trust management requirements, whereas in [146], integrity, computability, and flexibility are identified as the key properties of trust models.

To evaluate the CA model comprehensively, we selected thirteen trust model criteria synthesized from diverse but widely cited publications [91,145,146]. These criteria capture both technical properties (e.g., robustness, scalability, overhead) and human-centric qualities (e.g., explainability, user acceptance). Our selection reflects the recurring themes across the literature and the requirements of open, dynamic Multi-Agent System (MAS) environments. While all thirteen criteria are important, their relative priority may vary based on application needs. For example, scalability and overhead are critical in IoT settings with resource constraints, whereas explainability and user acceptance are more relevant in human-agent collaboration scenarios.

This section outlines a comprehensive evaluation of the CA model based on the thirteen evaluation criteria.

6.1.1. Decentralization

Centralized cloud-based trust assessment has significant drawbacks, including a single point of failure, scalability issues, and susceptibility to data misuse and manipulation by the companies that own cloud servers. As a response, decentralized-distributed solutions are becoming the norm in trust management [147]. The CA model follows the decentralized approach by allowing each trustee to compute trust independently, eliminating reliance on a central authority.

6.1.2. Subjectivity

Trust is inherently subjective, as different service requesters may have different service requirements in different contexts, affecting their perception of the trustworthiness of the same service provider [146]. Therefore, trust models should take into account the trustor's subjective opinion [91]. CA satisfies the subjectivity requirement, because the service requester defines and broadcasts the task and its requirements, and after the task's

execution, the service requester gives feedback to the service provider. Then the service provider decides whether the task was successfully executed, modifying the weight of the relevant connection, based on the feedback received.

6.1.3. Context Awareness

Since trust is context-dependent, a service provider may be trustworthy for one task but unreliable for another. The context typically refers to the task type, but it can also refer to an objective or execution environment with specific requirements [91]. CA ensures context awareness by allowing trustees to maintain distinct trust connections for different tasks requested by the same trustor.

6.1.4. Dynamicity

Trust is inherently dynamic, continuously evolving in response to events, environmental changes, and resource fluctuations. Recent trust information is much more important than historical information [91]. The CA model satisfies the dynamicity requirement, since trust update is event-driven, taking place after each task execution, making it well suited for dynamic environments, where agents may frequently join, leave, or alter their behavior, as evidenced by the experiments in this study.

6.1.5. Availability

A trust model should ensure that all entities and services remain updated [148] and fully available, even in the face of attacks or resource constraints [145]. In the CA approach, the service requester does not assign a task to a specific service provider but instead broadcasts the task to all nearby service providers. This ensures that even if a particular service provider is unavailable due to resource limitations (e.g., battery depletion), another service provider that receives the request can still complete the task. Consequently, the proposed approach effectively satisfies the availability requirement.

6.1.6. Integrity and Transparency

All transaction data and interaction outcomes should be properly recorded, efficiently stored, and easily retrievable [146]. Trust values should be accessible to all authorized network nodes whenever they need to assess the trustworthiness of any device in the system [147]. In the CA approach, trust information is stored locally by trustees in the form of connections (trust relationships). This ensures that trust information remains available even when entities cross application boundaries or transition between systems.

Additionally, trustees can readily access their own trust information to evaluate their reliability for a given task, thereby satisfying the integrity and transparency requirement.

6.1.7. Scalability

Since scalability is closely related to processing load and time, a trust model must efficiently manage large-scale networks while maintaining stable performance, regardless of network size, and operating correctly when devices are added or removed [91]. Large-scale networks require increased communication and higher storage capacity, meaning that trust models must adapt to the growing number of nodes and interactions [89,148,149]. However, many existing trust management algorithms struggle to scale effectively in massively distributed systems [104].

The CA model is designed for highly dynamic environments, ensuring strong performance despite continuous population changes. In Chapter 3 we have experimentally demonstrated CA's resilience to fluctuations in consumer populations. In Chapter 5, we presented simulation experiments showing that the enhanced CA algorithm exhibits significantly improved resilience, even when provider populations change. Additionally, since agents in the CA approach do not exchange trust information, the model avoids scalability issues related to agent communication. However, we have yet to evaluate how CA scales with an increasing number of nodes.

6.1.8. Overhead

A trust model should be simple and lightweight, as calculating trust scores may be impractical for IoT devices with limited computational resources [89]. It is essential to ensure that a trust model does not excessively consume a device's resources [103]. Both computational and storage overhead must be considered [91], as devices typically have constrained processing and storage capabilities and must prioritize their primary tasks over trust evaluation. Additionally, excessive computational overhead can hinder real-time trust assessment, negatively impacting time-sensitive applications. A trust model's efficiency can be analyzed using Big-O notation for time and space complexity. The CA model is relatively lightweight because its algorithm avoids complex mathematical computations. We provide a sketch of a computational complexity analysis of Algorithm 3 in Appendix D, outlining the core reasoning behind the CA model's efficiency using Big-O notation. However, a detailed complexity analysis, including a formal derivation of its time and space complexity, remains future work. Since all trust information is stored locally by

trustees, further research is needed to evaluate the storage overhead of the CA model and explore potential optimization techniques.

6.1.9. Accuracy

A trust model must effectively identify and prevent malicious entities by ensuring precise classification [89]. It should achieve a high level of accuracy, meaning that the computed trust value closely reflects the true value (i.e., the ground truth) [145]. Several studies [150–153] assess proposed trust models using metrics such as Precision, Recall, F1-score, accuracy, False Positive Rate (FPR), and True Positive Rate (TPR). However, the accuracy of the CA model has not yet been evaluated using these metrics.

6.1.10. Robustness

Trust models must withstand both anomalous behavior caused by sensor malfunctions [149] and trust-related attacks from insider attackers who deliberately act maliciously for personal gain or to disrupt system performance [89,154]. In heterogeneous networks, constant device connectivity and weak interoperability between network domains create numerous opportunities for malicious activities [91]. Traditional cryptographic and authentication methods are insufficient against insider attacks, where attackers possess valid cryptographic keys [149]. Existing trust models can only mitigate certain types of attacks, but none can fully defend against all threats [108]. Therefore, developing algorithms capable of detecting a broad range of malicious activity patterns is crucial [89]. In the following section, we examine common trust-related attacks discussed in the literature and evaluate the CA model's effectiveness in countering them.

6.1.11. Privacy Preservation

Ensuring a high level of privacy through data encryption is essential during trust assessment to prevent sensitive information leaks [147]. A trust model must safeguard both identity privacy (IP) and data privacy (DP) [91]. Specifically, feedback and interaction data must remain protected throughout all stages of data management, including collection, transmission, storage, fusion, and analysis. Additionally, identity details such as names and addresses should be shielded from unauthorized access, as linking trust information to real identities can lead to serious risks, such as Distributed Denial-of-Service (DDoS) Attacks [148]. In the CA approach, while entities generally do not exchange trust information (e.g., recommendations) interaction feedback is transmitted

from the trustor to the trustee, allowing the trustee to assess and locally store its own trustworthiness in the form of connections. Therefore, protecting the trustor's identity and ensuring data privacy during transmission, storage, and analysis are critical considerations in the CA approach.

6.1.12. Explainability

Trust models should be capable of providing clear and understandable explanations for their results. The ability to analyze and justify decisions is essential for enhancing user trust, compliance, and acceptance [155]. Additionally, it is important to clarify the processing logic and how trust metrics influence trust evaluations [91]. In this study, we made two key contributions to improve the explainability of our model. First, we conducted a semi-formal analysis to identify potential modifications to the CA algorithm's processing logic that could enhance its performance. Second, we carried out a series of simulation experiments to demonstrate the effectiveness of these improvements.

6.1.13. User Acceptance

The acceptance of a trust model depends on factors such as Quality of Service (QoS), Quality of Experience (QoE), and individual user preferences [91]. It can be assessed by gathering user feedback through questionnaires. However, the user acceptance of the CA model has not yet been evaluated.

Overall, the CA model satisfies the requirements of decentralization, subjectivity, context awareness, dynamicity, availability, integrity, and transparency, but several aspects require further enhancement and evaluation. Future research should focus on scalability, as the model's performance in large-scale environments with a high number of nodes remains untested. Additionally, while trust data are stored locally, their impact on system overhead and storage efficiency needs further investigation. Accuracy assessment using standard metrics such as Precision, Recall, and the F1-score is also necessary to validate the model's reliability. In terms of robustness, although the model resists false recommendations, its resilience against insider attacks requires deeper analysis. Privacy preservation remains an open challenge, particularly with respect to safeguarding trustor identity and feedback transmission. Finally, user acceptance has yet to be assessed, making it essential to evaluate the model's adoption based on Quality of Service (QoS) and Quality of Experience (QoE). Addressing these challenges will strengthen the CA model's effectiveness and applicability in dynamic environments. To the best of our

knowledge, comprehensive evaluations covering all thirteen criteria remain relatively uncommon in the trust-management literature.

6.2. CA's Resilience to Trust-Related Attacks

In this section, we examine the CA model's resilience against the most common trust-related attacks identified in various research studies [84,91,103–105,108,116,147–149,156,157]. A malevolent node is typically defined as a socially uncooperative entity that consistently seeks to disrupt system operations [149]. Its primary objective is to provide low-quality services to conserve its own resources while still benefiting from the services offered by other nodes in the system. Malicious nodes can employ various trust-related attack strategies, each designed to evade detection through different deceptive tactics [108].

6.2.1. Malicious with Everyone (ME)

In this attack, a node consistently provides low-quality services or misleading recommendations, regardless of the requester. This is one of the most fundamental types of attacks. To counter the ME attack, the CA approach could incorporate a contract-theoretic incentive mechanism. This mechanism would reward honest service providers with utility gains while penalizing dishonest service providers with utility losses, similarly to the approach in [101], by awarding or deducting credit coins.

6.2.2. Bad-mouthing Attack (BMA)

In this attack, one or more malicious nodes deliberately provide negative recommendations to damage the reputation of a well-behaved node, reducing its chances of being selected as a service provider. However, in the CA approach, agents do not exchange trust information through recommendations. Since nodes cannot act as recommenders, the CA model is inherently immune to the BMA.

Most studies assume that service requesters are honest, but as noted in [99], service requesters can also be "ill-intentioned" or "dishonest", deliberately giving low ratings to a service provider despite receiving good service. Additionally, some service requesters may be "amateurs" and incapable of accurately assessing service quality. This represents a specific type of bad-mouthing attack, against which the CA approach is also resilient. In the CA model, service providers have the autonomy to accept or reject service requests, allowing them to maximize their utility while avoiding dishonest or amateur service

requesters. If a service requester unfairly assigns a low rating to a high-quality service, the service provider may interpret this as evidence of an unreliable interaction and respond by decreasing the weight of its connection with that service requester, reducing the likelihood of future interactions. This serves as a built-in mechanism to penalize dishonest service requesters.

6.2.3. Ballot Stuffing (BSA) or Good-mouthing Attack

This attack occurs when one or more malicious recommenders (i.e., as a part of a collusion attack) falsely provide positive feedback for a low-quality service provider to boost its reputation and increase its chances of being selected, ultimately disadvantaging legitimate, high-quality service providers.

Since the CA approach does not involve nodes exchanging recommendations, it is inherently resistant to any BSA carried out by recommender nodes, which is common in other trust models.

However, a specific variation of this attack can occur when an ill-intended or amateur service requester assigns a service provider a higher rating than it actually deserves [99]. In this scenario, while the service provider benefits from an inflated rating, it also suffers by misjudging its actual service capability. Since the CA algorithm requires the service provider to compute the average weight of its existing connections to assess its performance, an inaccurate self-evaluation could lead to financial losses when dealing with honest service requesters. Thus, service providers have a strong incentive to identify and avoid dishonest or inexperienced service requesters.

To mitigate this risk, we could implement a mechanism similar to the comparison of Tlocal and Tglobal values from [99] or the Internal Similarity concept from [105], where the service provider would evaluate the consistency of a given rating by comparing it to the average or median weights of connections with other service providers for the same service.

Alternatively, a contract-theoretic incentive mechanism, as in [100], could deter dishonest service requesters by requiring them to pay a disproportionately high amount for low-quality services received. Additionally, integrating credit quotas (coins) as proposed in [101] could further discourage good-mouthing attacks. The choice of mitigation strategy would depend on the specific application environment in which the CA approach is implemented.

6.2.4. Self-promoting Attack

In this attack, a malicious node falsely provides positive recommendations about itself to increase its chances of being chosen as a service provider. Once selected, it then delivers poor services or exploits the network. This type of attack is especially effective against nodes that have not previously interacted with the attacker.

In the CA approach, service providers do not provide recommendations about themselves, and service requesters do not select service providers. As a result, traditional self-promotion tactics are ineffective.

However, a dishonest service provider could still choose to provide a service despite knowing that it would harm the service requester. To prevent such behavior, it is crucial to incorporate a penalty mechanism that imposes a financial loss on dishonest service providers.

Potential solutions include the fee charge concept from [99], the contract-theoretic mechanism from [100], or an incentive mechanism using credit quotas, such as TrustCoin [101]. The choice of mechanism would depend on the specific application scenario.

6.2.5. Opportunistic Service Attack (OSA)

This attack occurs when a node manipulates its behavior based on its reputation. When its reputation declines, it offers high-quality services, but once its reputation improves, it provides poor services. This strategy allows the node to sustain a sufficient level of trust to continue being chosen as a service provider. However, in the CA approach, OSA does not enhance a malicious node's chances of being selected as a service provider since trustors do not choose trustees. Consequently, our approach remains resilient against opportunistic service attacks.

6.2.6. Sybil Attack (SA)

This attack occurs when a malicious node generates multiple fake identities to manipulate the reputation of a target node unfairly by providing various ratings. In the CA approach, a malicious service requester could use this tactic to hinder a target service provider's ability to accurately assess its service capability, deplete its resources, and reduce its chances of being selected as a service provider. To counter this attack, we can apply the same mechanisms proposed for addressing the good-mouthing attack. Specifically, an approach based on the Internal Similarity concept [105] or a contract-theoretic mechanism based on incentives [100] could either deter attackers from carrying

out their attacks or assist the service provider in identifying and isolating malicious service requesters.

6.2.7. Whitewashing Attack (WA)

Also known as a Re-entry Attack, this attack occurs when a malicious node abandons the system after its reputation drops below a certain threshold and then re-enters with a new identity to erase its negative history and reset its reputation to a default value. In the CA approach, this attack is ineffective because changing a service provider's identity does not impact its likelihood of being selected as a service provider, as trustors do not choose trustees.

6.2.8. On-Off Attack (OOA)

Also called a Random Attack, this is one of the most challenging attacks to detect. In this attack, a malicious node alternates its behavior between good (ON) and bad (OFF) in an unpredictable manner, restoring trust just before launching another attack. During the ON phase, the node builds a positive reputation, which it later exploits for malicious activities. The CA approach is resilient against the OOA since trustors do not select trustees, meaning a high reputation does not increase a node's chances of being chosen as a service provider.

6.2.9. Discrimination or Selective Misbehavior Attack

This kind of attack occurs when an attacker selectively manipulates its behavior by providing legitimate services for certain network tasks while acting maliciously against others. To mitigate this attack in the CA approach, a penalty mechanism should be implemented to impose financial consequences on dishonest service providers. Depending on the application scenario, this could involve the "fee charge concept" [99], a contract-theoretic mechanism [100], or an incentive mechanism using credit quotas like in TrustCoin [101]. One form of this attack is selfish behavior, where service providers prioritize easier tasks that require less effort. Rational agents would evaluate each task based on expected utility – calculating the probability of success multiplied by the reward – and choose accordingly. To counter this in the CA approach, it is crucial to implement incentives that encourage capable agents to take on more challenging tasks. For example, a contract-theoretic mechanism can ensure that highly skilled agents prefer difficult tasks with higher expected utility.

6.2.10. Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks

These attacks aim to disrupt a service by overwhelming it with excessive traffic or resource-intensive requests. While a DoS attack originates from a single attacker, DDoS attacks involve multiple compromised devices working together to target a single system, making them more difficult to counter. In the CA approach, these attacks remain a potential threat. Therefore, future research should focus on developing effective mechanisms to mitigate DoS and DDoS attacks.

6.2.11. Storage Attack

This attack occurs when a malicious node manipulates stored feedback data by deleting, modifying, or injecting fake information. In the CA approach, an attacker may attempt to alter or remove the locally stored connections of a trustee. To counter this threat, future research should aim to find methods to ensure data integrity and prevent tampering.

6.3. Potential Mechanisms for Enhancing Attack Resilience

Overall, the CA model demonstrates strong resilience against a substantial subset of the trust-related attacks discussed in the literature, particularly those based on recommendation manipulation and reputation inflation. This resilience stems directly from its trustee-centric architecture and the absence of recommendation exchange.

At the same time, several attack categories require complementary mechanisms, while others remain open research challenges. Consequently, although the CA model cannot be considered universally secure against every form of malicious behavior, its architecture provides a robust foundation upon which additional protection mechanisms can be incorporated.

Several mechanisms proposed in the trust management literature could be integrated into the CA framework to further strengthen its resilience against dishonest or selfish behavior.

6.3.1. Incentive Mechanisms

Incentive mechanisms could be naturally integrated into the CA model through its existing connection-strengthening process (Figure 51). After evaluating the received service, a trustor may reward a provider when the delivered quality meets or exceeds the requested level. The provider can then use this reward as the reinforcement signal to

strengthen the connection associated with that trustor. Conversely, when the delivered quality is unsatisfactory, no reward is issued and the connection is not reinforced. Such a mechanism would create a direct incentive for providers to maintain high performance levels, since successful service provision would increase both the immediate and the long-term benefits received.



Figure 51. CA Approach with Incentives

6.3.2. Penalty Mechanisms

A complementary approach involves the introduction of penalty mechanisms (Figure 52). Providers that repeatedly deliver poor-quality services, behave dishonestly, or intentionally attempt to exploit the system could incur utility losses, lose accumulated credits, or face temporary or permanent exclusion from the system. Such mechanisms could discourage malicious behavior while reinforcing cooperation among participants.



Figure 52. CA Approach with Penalties

6.3.3. Contract-Theoretic Mechanisms

Another promising direction is the incorporation of contract-theoretic mechanisms (Figure 53). Prior to task execution, service requesters and providers could establish agreements specifying the expected service quality and the corresponding compensation. More demanding tasks would be associated with higher rewards, encouraging highly capable providers to undertake tasks that require greater expertise. Failure to satisfy contractual obligations would result in reduced trust and lower future rewards. Such mechanisms could also mitigate selfish behavior by aligning provider incentives with system objectives.

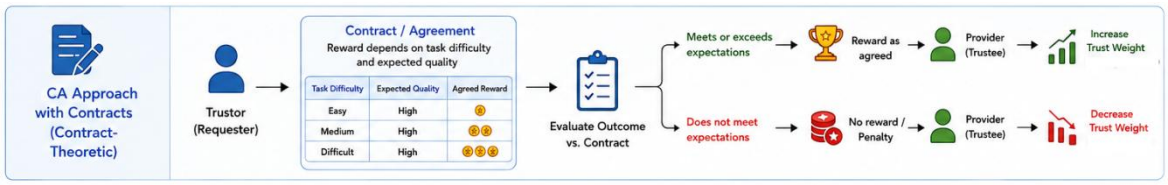


Figure 53. CA Approach with Contracts

6.3.4. Consistency-Checking Mechanisms

Finally, the CA model could be extended with consistency checking mechanisms designed to identify dishonest or inexperienced service requesters. Since providers already maintain local trust information, they can compare newly received evaluations against historical evaluations associated with similar services (Figure 54). Ratings that deviate significantly from the expected range could be considered suspicious and assigned lower credibility. This approach could help detect both bad-mouthing and good-mouthing behaviors originating from dishonest trustors, while also reducing the influence of inaccurate assessments provided by inexperienced users.

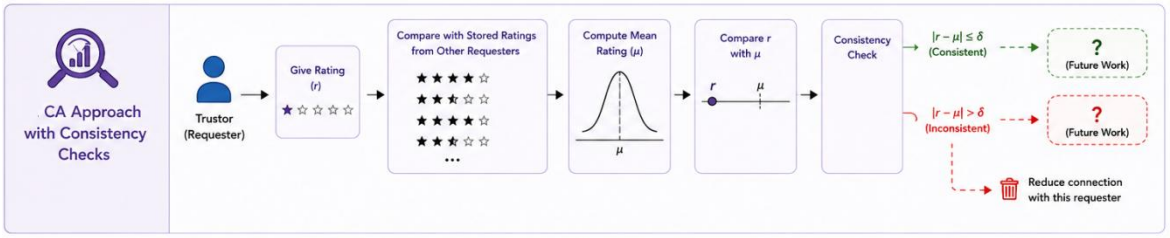


Figure 54. CA Approach with Consistency Checks

The precise design, implementation, and evaluation of these mechanisms remain open research directions. Nevertheless, their integration appears particularly compatible with the decentralized and trustee-centric architecture of the CA model and could substantially improve its robustness in adversarial environments.

6.4. Conclusions

This chapter broadened the evaluation of the CA model beyond performance-oriented simulation experiments. Rather than focusing exclusively on utility gains under dynamic environmental conditions, the analysis examined the model from the perspective of trust management requirements and security-related challenges.

The evaluation showed that the CA model satisfies several important criteria commonly associated with effective trust management systems, including decentralization,

subjectivity, context awareness, dynamicity, availability, integrity, transparency, and explainability. These properties emerge naturally from the trustee-centric design of the model, the local storage of trust information, and the task-specific representation of trust relationships. As a result, the CA approach offers a distinctive alternative to traditional trustor-centric trust and reputation models.

The results further indicate that the architectural choices underlying CA have implications that extend beyond trust estimation accuracy. In particular, the absence of recommendation exchange reduces communication requirements and naturally mitigates several forms of reputation manipulation and misinformation attacks. At the same time, the evaluation revealed that certain challenges remain only partially addressed and require additional mechanisms or architectural extensions.

Perhaps the most important outcome of this chapter is that it provides a clearer understanding of both the strengths and the limitations of the CA approach. Rather than evaluating the model solely through simulation performance, the analysis identifies the broader conditions under which the model can be expected to operate effectively and highlights the assumptions upon which its behavior relies.

Several open challenges remain, including resource exhaustion attacks, the protection of locally stored trust information against tampering, stronger defenses against insider attacks, and enhanced privacy-preserving mechanisms for the management of feedback and trust-related data. Nevertheless, these limitations should be viewed in the broader context of trust management research. Trust modeling remains an inherently difficult problem, and existing trust and reputation models typically provide protection against only a subset of known attacks rather than offering universal resilience against all forms of malicious behavior.

Overall, the evaluation supports the view that CA constitutes a promising foundation for trust management in open and dynamic multi-agent systems. The findings of this chapter therefore serve not only as an assessment of the current state of the model but also as a roadmap for its future evolution and refinement.

7. Trust-Based Decision Support for Emergency Department Operations: A Simulation Framework Case Study Based on the CA Model

Emergency Departments (EDs) operate under highly dynamic and resource-constrained conditions. Patients arrive with different levels of urgency, staff availability changes over time, and decisions must often be made under significant uncertainty. In such environments, assigning tasks to the most appropriate healthcare professional is critical for both patient safety and operational efficiency.

The task allocation problem in an ED shares several characteristics with those encountered in open multi-agent systems. Doctors generate requests for clinical tasks, nurses act as service providers, and both staff availability and performance may vary over time. Consequently, the ED constitutes an interesting application domain for evaluating computational trust mechanisms.

In previous chapters, the final version of the *Create Assemblies* (CA) algorithm was presented and evaluated. A key characteristic of this version is the ability of service providers to continuously self-assess their performance and avoid tasks for which they are no longer sufficiently competent. This capability is particularly relevant in healthcare environments, where assigning critical tasks to underperforming personnel may have serious consequences.

To investigate the practical implications of this approach, this chapter presents a simulation framework that models an emergency department as a multi-agent system. Doctors, nurses, and patients are represented as autonomous agents interacting within a dynamic environment implemented in the Unity game engine. The framework enables the evaluation of different task-allocation and staff-management strategies under controlled and repeatable conditions.

7.1. Simulation Framework

7.1.1. Overview of the Framework

To investigate the applicability of the *Create Assemblies* (CA) trust model in a healthcare setting, a simulation framework was developed using the Unity game engine.

The framework models an Emergency Department (ED) as a dynamic multi-agent environment in which healthcare professionals and patients interact under conditions of uncertainty, resource constraints, and continuously evolving workloads.

The primary objective of the framework is not to replicate the operation of a specific real-world emergency department, but rather to provide a controlled experimental environment for studying the impact of trust-based task allocation policies on operational performance and patient-related outcomes. By representing doctors, nurses, and patients as autonomous agents, the framework enables the systematic evaluation of alternative coordination strategies under repeatable conditions.

Within the simulation, doctors generate nursing task requests after examining patients, while nurses decide whether to undertake these tasks based on their perceived competence and current behavioral policy. The decision-making process is guided by the CA trust model presented in Chapter 5, allowing nurses to continuously assess their own performance and adapt their future participation accordingly. Consequently, task allocation emerges from the interaction between agent capabilities, task requirements, and trust-related mechanisms rather than being imposed by a centralized controller.

The framework was implemented in Unity due to its support for real-time visualization, agent navigation, event-driven interactions, and flexible scenario configuration. The three-dimensional environment allows for the observation of patient flow, staff movement, resource utilization, and task execution as the simulation progresses. Furthermore, the platform automatically records detailed performance metrics that can subsequently be analyzed to compare alternative operational policies.

A central feature of the framework is its support for multiple staff-management strategies. Beyond a baseline operational scenario, the simulation allows the evaluation of interventions such as personnel replacement and workforce training. These scenarios make it possible to investigate how different management decisions influence patient safety, operational efficiency, and staff development over time.

Overall, the proposed framework serves as a decision-support platform for exploring the consequences of alternative task-allocation and staffing policies before their potential application in real clinical environments. At the same time, it provides a realistic and challenging application domain for evaluating the behavior of the CA trust model in a safety-critical setting.

7.1.2. Agents and Roles

The simulation framework models the Emergency Department (ED) as a multi-agent system composed of four main agent categories: doctors, nurses, trainer nurses, and patients. Each agent type has a distinct role within the operational workflow and contributes to the overall dynamics of task generation, task execution, and trust-based decision-making.

7.1.2.1. Doctors

Doctors are responsible for examining patients and generating nursing task requests. After assessing a patient's condition, a doctor determines the level of performance required for the requested nursing intervention and broadcasts the corresponding task request to all available nurses. Each request contains information about the expected difficulty of the task and the maximum acceptable completion time.

In the current implementation, doctors do not directly select the nurse who will perform a task. Instead, they act solely as task requesters, allowing task-allocation decisions to emerge through the interactions between nurses and the CA trust mechanism. The design is consistent with the trustee-centric philosophy of the CA model, where service providers determine whether they are capable of undertaking a requested task.

7.1.2.2. Nurses

Nurses represent the primary service providers within the simulation. Upon receiving task requests from doctors, they evaluate whether they should undertake a task according to the active decision-making policy and their perceived capability to perform it successfully.

When a nurse accepts a task, the nurse travels to the corresponding patient, performs the required intervention, and subsequently updates its internal trust-related information based on the observed outcome. Through repeated interactions, nurses continuously refine their self-assessment and adapt their future behavior. In scenarios using the CA model, nurses that repeatedly perform poorly may classify themselves as unreliable for certain task categories and avoid undertaking tasks that exceed their capabilities.

This self-regulatory behavior constitutes the core mechanism through which the framework seeks to improve patient safety and reduce the likelihood of assigning complex tasks to unsuitable providers.

7.1.2.3. Trainer Nurses

Trainer nurses are introduced exclusively in the Training scenario. Their role is to assist nurses that have identified themselves as low-performing through the self-assessment mechanism of the CA model.

Rather than directly replacing the underperforming nurse, trainer nurses accompany and supervise task execution, enabling knowledge transfer and gradual skill development. Through repeated guided practice, the performance of the trainee nurse improves over time, allowing the nurse to eventually resume independent task execution.

This mechanism models workforce-development strategies frequently encountered in healthcare organizations, where mentoring and supervised practice are used as alternatives to personnel replacement.

7.1.2.4. Patients

Patients represent the recipients of healthcare services within the simulation. They arrive dynamically, occupy available beds, and require nursing interventions after being examined by a doctor.

Each patient is associated with a task characterized by a specific difficulty level and urgency. Patient outcomes are indirectly influenced by the quality and timeliness of task execution. Delays, unsuccessful task completions, and inefficient resource allocation contribute to the patient-related performance measures recorded by the simulation.

Although patients do not actively participate in decision-making, they provide the operational context that drives interactions among healthcare professionals and enables the evaluation of alternative staffing and coordination strategies.

7.1.3. Modeling Staff Heterogeneity

Healthcare professionals differ considerably in their experience, skills, judgment, and overall performance. Such variability plays an important role in emergency department operations, where decisions must often be made under time pressure and uncertainty. To capture these differences, the simulation framework incorporates heterogeneity among staff members through doctor evaluation styles and nurse quality attributes.

7.1.3.1. Doctor Evaluation Styles

Doctors are responsible for examining patients and estimating the performance level required for the successful execution of a nursing task. However, different doctors may not always evaluate task difficulty in the same manner. To represent this variability, each doctor is assigned an evaluation style that determines how task requirements are estimated.

Three evaluation styles are considered:

- **Estimates Correctly,** where the doctor accurately assesses the true performance level.
- **Overestimates,** where the doctor systematically assigns higher performance levels than the true value.
- **Underestimates,** where the doctor systematically assigns lower performance levels than the true value.

These alternative evaluation styles enable the simulation of realistic differences in clinical judgment. Overestimation may result in unnecessarily strict task requirements and underutilization of available staff, whereas underestimation may expose patients to increased risks by assigning complex tasks to insufficiently capable providers. Consequently, doctor evaluation styles influence both the perceived difficulty of tasks and the subsequent decisions made by nurses.

When a doctor examines a patient, the function *EvaluatePerformanceLevel()* maps the task's true difficulty to the estimated level according to the doctor's evaluation style. The pseudocode of the core decision logic is presented in Appendix E.1.

7.1.3.2. Nurse Quality Attributes

The second source of heterogeneity concerns the capabilities of nurses responsible for executing clinical tasks. At the beginning of each simulation run, nurses are assigned predefined quality attributes that represent their overall level of competence and performance.

In the current implementation, two categories are considered:

- **High-performing nurses**, who typically complete tasks more quickly and more reliably.
- **Low-performing nurses**, who require longer execution times and exhibit a higher probability of poor performance.

Nurse quality directly affects task execution outcomes and therefore influences the evolution of trust within the CA model. High-performing nurses are more likely to successfully complete demanding tasks and maintain positive self-assessments, whereas low-performing nurses are more likely to identify limitations in their capabilities and restrict their participation to tasks that better match their skills.

In the Training scenario, nurse quality is not static. Through repeated interactions with a trainer nurse, low-performing nurses gradually improve their capabilities, allowing the framework to model skill acquisition and workforce development over time.

Nurse performance is modeled in the *GetTaskDuration()* function. The logic governing task execution speed for different nurse qualities is presented in Appendix E.2.

7.1.3.3. Role of Heterogeneity in the Simulation

The combination of doctor evaluation styles and nurse quality attributes creates a heterogeneous environment in which task allocation decisions are influenced by both the characteristics of the requester and those of the provider. This heterogeneity is essential for evaluating the behavior of the CA trust model, as the benefits of self-assessment and adaptive task selection become particularly evident when agents differ in their capabilities and judgment.

By explicitly representing variability among healthcare professionals, the framework moves beyond simplified homogeneous agent simulations and provides a more realistic setting for studying trust-based coordination in emergency department operations.

7.1.4. Emergency Department Layout

The simulation environment represents a simplified emergency department (ED) implemented as a three-dimensional virtual space within the Unity game engine. The objective of the layout is not to reproduce a specific hospital facility, but to provide a realistic operational environment in which staff movement, patient flow, and task execution can be observed and analyzed.

The virtual ED consists of three main areas: the patient treatment area (bed zone), the staff waiting area, and the department exit. These areas, depicted in Figure 55, define the spatial context in which agents interact throughout the simulation.

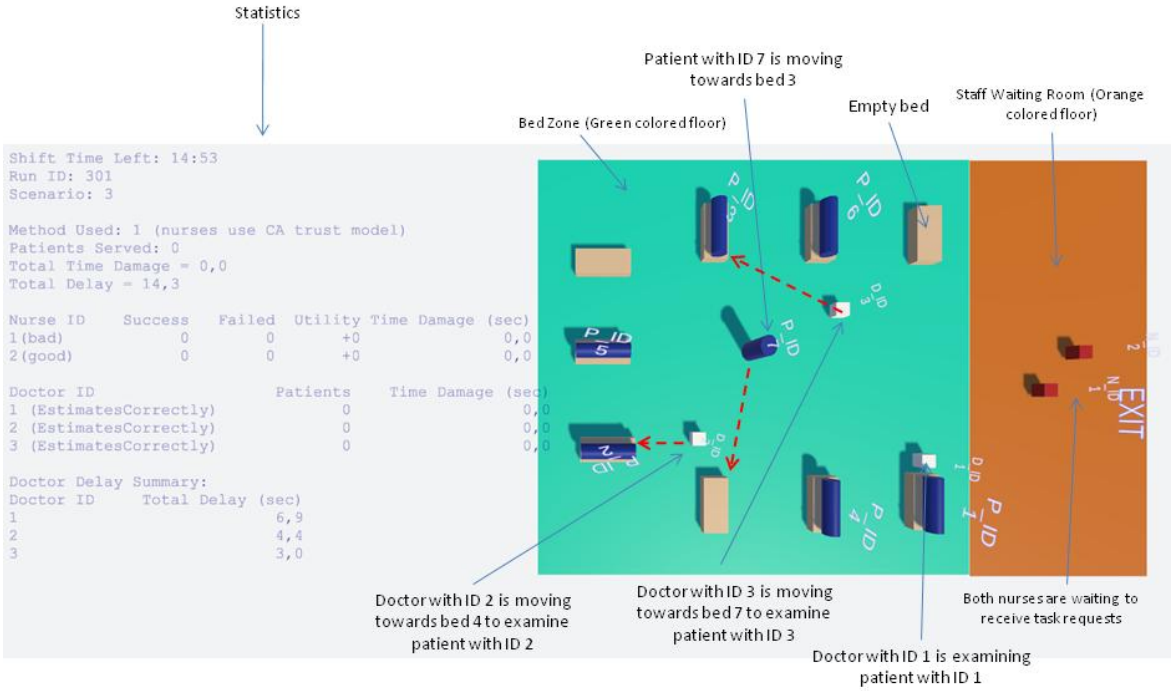


Figure 55. Simulation snapshot showing the bed zone (green area), staff waiting room (orange area), and the movement of agents. Red arrows indicate movement directions, while labels describe the current activity of agents. Dynamically updated statistics are displayed on the left side.

7.1.4.1. Patient Treatment Area

The treatment area (bed zone) contains nine patient beds where examinations and nursing interventions take place. Patients are dynamically admitted to available beds and remain there until all required nursing tasks have been completed.

To simplify workload management, each doctor is assigned responsibility for a fixed subset of beds. Specifically, the first doctor is responsible for Beds 1-3, the second doctor for Beds 4-6, and the third doctor for Beds 7-9. Consequently, each doctor examines only the patients assigned to their designated beds and generates the corresponding nursing task requests.

This arrangement ensures a balanced distribution of patients among doctors while maintaining a predictable workflow throughout the simulation.

7.1.4.2. Staff Waiting Area

The staff waiting area serves as the default location for doctors and nurses when they are not actively engaged in patient care. At the beginning of each simulation run, healthcare professionals are instantiated in this area and remain there until assigned to a task.

After completing a task, nurses return to the waiting area, where they remain available for future assignments.

7.1.4.3. Department Exit

A dedicated exit point is included within the environment to represent patient discharge from the emergency department. Once all required interventions have been completed, patients leave their assigned beds and navigate toward the exit, thereby releasing resources for newly arriving patients.

7.1.4.4. Agent Navigation and Visualization

All agents navigate autonomously within the environment using Unity's pathfinding mechanisms. Doctors move between beds to examine patients, nurses travel to patient locations to perform assigned interventions, and patients move between the entry point, beds, and the exit area.

The simulation also provides real-time visual feedback on agent activities. Floating labels and task timers display information such as agent identifiers, current actions, and remaining task execution times. Additionally, operational statistics are continuously updated during the simulation, allowing direct observation of patient flow, staff utilization, and system performance.

7.1.4.5. Simplifications and Future Extensions

The current implementation adopts a fixed department layout, a predefined number of beds, and static doctor-to-bed assignments. These design choices were made to facilitate controlled experimentation and reproducibility across simulation runs.

Although simplified, the environment provides sufficient realism for evaluating trust-based task-allocation strategies and staff-management policies. Future extensions may incorporate hospital-specific floor plans, dynamic bed allocation policies, and more detailed workflow representations, enabling the simulation of real emergency departments and supporting more comprehensive decision-support analyses.

7.1.5. Workflow Description

The simulation framework models the daily operation of a simplified emergency department through a sequence of interactions among patients, doctors, and nurses. Patient arrivals, medical examinations, task requests, and nursing interventions are represented as discrete events that collectively determine the evolution of the system over time.

Although all scenarios share a common operational workflow, the behavior of nurses differs depending on the selected staffing strategy. The following sections describe the baseline workflow and the additional mechanisms introduced in the Replacement and Training scenarios.

7.1.5.1. Base Workflow

At the beginning of each simulation run, patients are generated sequentially and assigned to available beds according to a predefined allocation order. This arrangement ensures that all doctors become engaged as early as possible and that the workload is distributed across the emergency department.

Once a patient occupies a bed, the responsible doctor travels to the patient's location and performs an examination. Following the examination, the doctor generates a nursing task request. In the current implementation, the only modeled nursing task is the insertion of an intravenous (IV) catheter, although the framework has been designed to support additional task categories in future extensions.

The task request contains information about the estimated difficulty of the intervention and the expected completion time. This information is then broadcast to all available nurses.

Upon receiving a task request, each nurse evaluates whether to undertake the task according to the selected decision policy. Under the CA model, nurses use their current self-assessment to determine whether the task matches their capabilities. Under the FIFO policy, requests are processed in arrival order without considering performance levels.

When a nurse accepts a task, the nurse travels to the corresponding patient, verifies that the task has not already been completed by another nurse, and performs the intervention. Task execution is represented by a time delay that depends on both task difficulty and nurse quality. After completing the intervention, the nurse returns to the staff waiting area and becomes available for future assignments.

Patients remain in the emergency department until all required interventions have been completed, after which they leave the system and release their assigned bed for incoming patients.

7.1.5.2. Replacement Workflow

The Replacement scenario extends the baseline workflow by introducing an additional staffing intervention aimed at mitigating the negative impact of low-performing

nurses on patient care. The intervention is triggered when a nurse, through the self-assessment mechanism of the CA model, determines that their performance has fallen below the required threshold for handling more demanding tasks.

Rather than attempting complex interventions that may lead to delays or failures, the low-performing nurse restricts their participation to tasks that match their current capabilities. To ensure that higher-difficulty requests are still addressed promptly, the system introduces an additional high-performing nurse who assumes responsibility for the remaining complex tasks.

Once activated, the replacement nurse operates in the same manner as any other nurse in the simulation, receiving pending requests and executing tasks according to their priority and difficulty. This mechanism reduces the likelihood that critical interventions will be delayed because of insufficient staff competence while simultaneously allowing the low-performing nurse to continue contributing to less demanding activities.

From an operational perspective, the Replacement scenario represents a staffing policy in which performance deficiencies are addressed through the temporary addition of experienced personnel. Such an approach can improve patient throughput and reduce delays associated with complex tasks, although it requires additional human resources and may increase operational costs. The scenario therefore provides a useful mechanism for evaluating the trade-off between service quality and staffing expenditure within the simulated emergency department.

7.1.5.3. Training Workflow

The Training scenario adopts a different strategy for addressing performance deficiencies. Instead of introducing additional personnel to handle difficult tasks, the framework focuses on improving the capabilities of low-performing nurses through supervised training.

When a nurse identifies, through the CA self-assessment mechanism, that their performance has fallen below the acceptable threshold, a trainer nurse is assigned to provide guidance and support. The trainer accompanies the trainee during task execution, observing the process and providing assistance throughout the learning period. Unlike the Replacement scenario, responsibility for task execution remains with the trainee nurse, while the trainer acts solely as a mentor.

As the trainee completes tasks under supervision, their performance gradually improves. This improvement is represented through incremental updates to the nurse's quality attributes, reflecting the acquisition of experience and practical skills. Once the nurse reaches the required performance level, the training phase is considered complete and the trainer leaves the system. The nurse then resumes normal operation without external assistance.

The Training scenario models a long-term workforce-development strategy in which performance problems are addressed through education rather than personnel replacement. Although the learning process may initially introduce additional delays and increase the risk of unsuccessful task execution, it enables sustainable improvement of staff capabilities and reduces future dependence on additional personnel. Consequently, this scenario is useful for studying the trade-off between short-term operational efficiency and long-term workforce development within emergency department environments.

7.1.6. Logged Metrics

To evaluate the impact of different staffing strategies and task allocation policies, the simulation framework automatically records a set of performance indicators throughout each simulation run. These metrics capture both operational efficiency and service quality, enabling quantitative comparison between alternative scenarios.

7.1.6.1. Number of patients served

This metric measures the total number of patients who have been fully attended to during the shift. A patient is considered "served" when all tasks requested by their assigned doctor have been successfully performed by the nurses. In addition to the total shift value, this metric is also recorded per doctor, enabling analysis of how the doctor's evaluation style (estimates correctly, overestimates, underestimates) relates to the number of patients served under their care.

7.1.6.2. Total time damage

It quantifies the cumulative extra time taken by nurses to complete tasks beyond the expected task duration. This excess time is considered detrimental to patient outcomes – either by reducing the patient's survival probability (e.g., in cases such as cardiac arrest where an IV catheter must be placed quickly to administer life-saving medication) or by increasing patient dissatisfaction due to prolonged waits. The simulation computes this

metric by summing all positive deviations (actual duration – expected duration) for completed tasks. In addition to the overall value for each shift, time damage is recorded per nurse – enabling analysis of how a nurse's quality attribute (e.g., high-performing, low-performing) correlates with the delays they cause – and per doctor, where it reflects the cumulative delays affecting that doctor's patients. This doctor-level recording allows investigation of how a doctor's evaluation style (estimates correctly, overestimates, underestimates) influences the delays, e.g., whether underestimation of task difficulty leads to higher time damage.

7.1.6.3. Total delay

It measures the total amount of time patients remain unattended by any nurse after they have been examined by their assigned doctor and one or more tasks requests have been sent to nurses. These are pure waiting periods in which the patient occupies a bed but receives no active nursing intervention. In addition to the overall shift value, total delay is also recorded per doctor, allowing the analysis of how each doctor's evaluation style influences the waiting times experienced by their patients.

7.1.6.4. Task success/failure counts

For each nurse, the simulation records the number of successfully and unsuccessfully completed tasks. A task is counted as successful if the actual execution time does not exceed the time estimated and requested by the doctor who examined the patient. These metrics therefore represent the doctor's assessment of each nurse's performance in executing assigned tasks. As a result, due to a doctor's systematic bias in evaluation style, (see Section 7.1.3.1), a task may be labeled unsuccessful despite meeting the gold-standard duration (in cases of overestimation), or labeled successful despite exceeding it (in cases of underestimation). Overestimation occurs when the doctor assigns higher performance levels than required, imposing a stricter time threshold, while underestimation relaxes the threshold. For instance, if the true required level is 3 (40 s) but the doctor overestimates and requests level 5 (20 s), a completion time of 30 s – although below the true gold standard – will be labeled unsuccessful. Conversely, underestimation may label the same 30 s as successful, despite exceeding the true optimal duration.

7.1.6.5. Utility values for each nurse

For each nurse, the simulation records a cumulative utility score based on successfully completed tasks. When a task is successful (i.e., its actual execution time does not exceed the time estimated and requested by the examining doctor), the nurse's utility increases by the task's *requestedLevel* – an integer from 1 (easiest) to 5 (most difficult) representing the task's difficulty. This metric can serve as a motivational indicator, with nurses potentially aiming to maximize their utility over the course of the shift.

7.1.6.6. Doctors' delays and evaluation accuracy

These metrics are displayed in the in-game UI and exported to external .csv files for post-simulation statistical analysis. These performance indicators can also serve as proxies for evaluating personalized decision-support performance, linking individual agent behaviors with patient-level outcomes.

7.1.7. Customizability and Reproducibility

The platform is designed for experimental flexibility:

- Agent counts, roles, and attributes are configured via the *GlobalSettings* class.
- Scenario-specific logic is encapsulated in a modular structure to enable extension.
- Run IDs and parameter logging ensure reproducibility of results.

This modular, data-driven design supports iterative development of new scenarios and facilitates rigorous comparative analysis.

7.2. Decision-Support Experimentation Methodology

7.2.1. Objectives and Rationale

The primary aim of the decision-support experiments is to evaluate how alternative nurse decision-making policies and workflow strategies influence key operational outcomes in a simulated emergency room (ER) environment. By systematically varying agent behaviors and resource allocation strategies, the framework allows us to identify policies that minimize patient delays, reduce detrimental time damage, and optimize the utilization of available staff. Beyond operational benchmarking, the proposed framework provides a foundation for AI-driven personalization of clinical workflows, allowing the adaptation of decision policies to specific staff profiles and patient characteristics.

This approach supports data-driven decision-making in hospital management, enabling stakeholders to test "what-if" scenarios before implementing changes in real clinical settings (i.e., real-world hospital environments).

7.2.2. Experimental Design

7.2.2.1. Scenarios

Three distinct operational scenarios within the simulation framework can be evaluated. Each represents a different policy for handling low-performing nurses.

- **Scenario 1 – Baseline**: Low – performing nurses remain active but only perform tasks below a predefined difficulty threshold. More complex tasks are skipped by these nurses and remain unassigned until another available nurse chooses to execute them. Unlike the Replacement scenario, where an additional high-performing nurse assists with complex tasks, and the Training scenario, where a trainer nurse accompanies and guides the low-performing nurse, no extra nursing staff is provided in the Baseline scenario.
- **Scenario 2 – Replacement**: Low-performing nurses remain active but only perform tasks below a certain difficulty threshold, and an additional high-performing nurse is spawned to execute pending higher-difficulty tasks, ensuring they are addressed without excessive delay.
- **Scenario 3 – Training with Mentor**: Low-performing nurses are accompanied by a trainer nurse who supervises and guides them, improving their performance over time.

The FIFO decision-making policy is only meaningful in the Baseline scenario, as it does not support performance self-assessment or adaptive responses to low quality. In contrast, the CA trust model is applicable across all scenarios and is essential in Replacement and Training settings, as it enables nurses to detect low performance and trigger the corresponding assistance or mentoring mechanisms.

7.2.2.2. Experimental Variables

To ensure meaningful comparisons between alternative task allocation policies and staffing strategies, a number of simulation parameters were held constant throughout the experimental study. Specifically, all experiments were conducted using the same emergency department layout, bed allocation rules, patient assignment sequence, and task

type. Unless explicitly stated otherwise, the number of doctors and nurses, as well as the predefined evaluation styles of doctors and quality attributes of nurses, remained unchanged across simulation runs. Maintaining these conditions allowed the observed differences in performance to be attributed primarily to the coordination policies and interventions under investigation rather than to changes in the underlying environment.

The experimental design included several independent variables. The first was the task allocation policy, comparing the proposed CA trust-based mechanism with the conventional First-In-First-Out (FIFO) approach. The second independent variable was the operational scenario, namely the Baseline, Replacement, and Training configurations. In addition, the framework allows the investigation of different doctor evaluation styles (accurate estimation, overestimation, and underestimation of task difficulty) and different nurse performance profiles (high-performing and low-performing), thereby supporting the analysis of heterogeneous staff behaviors.

The impact of these variables was assessed through a set of dependent variables automatically recorded by the simulation framework. These included the number of patients served, representing overall throughput; total delay, measuring the cumulative waiting time experienced by patients before receiving nursing care; and total time damage, quantifying the adverse effects of delayed task execution. Additional performance indicators included task success and failure counts, which reflect the effectiveness of nursing interventions, as well as the utility accumulated by individual nurses during a simulation run. Together, these metrics provide a multidimensional view of emergency department performance, enabling the evaluation of operational efficiency, patient safety, and workforce effectiveness under different decision-making policies.

7.2.3. Experimental Procedure

The experiments are conducted for four specific scenario-policy combinations:

- Baseline scenario – CA trust model;
- Baseline scenario – FIFO;
- Replacement scenario – CA trust model;
- Training scenario – CA trust model.

For each of these combinations, the simulation is executed 60 times to account for stochastic variations in task durations and agent behaviors, each time using a different random seed to ensure statistical robustness. In the Replacement and Training scenarios,

only the CA trust model is tested, as this decision-making policy is the only one that allows nurses to detect their low performance and either request assistance from an additional high-performing nurse (Replacement) or receive guidance from a trainer nurse (Training). The FIFO policy is applied only in the Baseline scenario.

Simulation parameters are initialized via the *GlobalSettings* class. A unique Run ID is assigned to each execution for reproducibility. Patient spawning, doctors' examinations, and nurse task executions proceed according to the workflows described in Section 7.1.5. Performance metrics are automatically logged both in the in-game UI and in external CSV files.

7.2.4. Data Analysis Approach

Collected data are analyzed using descriptive statistics and inferential statistical tests (e.g., ANOVA, t-tests, or non-parametric equivalents where appropriate) to determine whether observed differences between policies are statistically significant.

From a decision-support perspective, the results highlight trade-offs between different strategies. For instance, the Replacement scenario may significantly reduce total delay for complex tasks but could require additional staffing resources, while the Training scenario may offer long-term benefits by improving low-performing nurses without permanently increasing headcount.

This analysis framework enables hospital managers and policymakers to:

1. Predict the operational impact of specific interventions before real-world deployment;
2. Compare alternative policies under controlled and repeatable conditions;
3. Identify scenario-specific strengths and weaknesses, informing adaptive staffing and training strategies.

The analytical framework further enables the exploration of personalized decision patterns, where performance metrics are examined at the level of individual agents, highlighting how staff-specific adaptation can contribute to precision medicine-informed decision-support approaches.

7.3. Results

7.3.1. Case Study Example

To illustrate the practical use of the simulation framework for decision support, we present a case study of a small emergency room (ER) setup. The configuration includes

three doctors (all accurately estimating task requests), two nurses (one high-performing and one low-performing), and nine beds.

Three main interventions were compared across multiple simulation runs:

1. Baseline, comparing task allocation methods (CA trust model vs. FIFO);
2. Replacement, introducing an additional high-performing nurse for difficult tasks;
3. Training, pairing the low-performing nurse with a trainer to improve skills over time.

A full methodological description, including detailed statistical analyses, is provided in Appendix F. In this section, we focus only on the key findings, their interpretation, and the main limitations, and present the conclusions drawn from the case study. This case study also illustrates how individualized staff behaviors and adaptive coordination strategies can be represented computationally, providing a bridge toward personalized and precision-oriented decision support in emergency care.

7.3.2. Key Findings

7.3.2.1. Baseline comparison (CA trust vs. FIFO)

The FIFO policy served more patients and reduced overall waiting times, but substantially increased patient damage time and task failures by the low-performing nurse. In contrast, the CA trust model prioritized safety, reducing harmful delays and error rates but at the cost of longer waiting times. This demonstrates the trade-off between safety and operational efficiency. From a personalized medicine perspective, this trade-off reflects the dynamic balance between individual performance reliability and patient-specific safety outcomes, highlighting the importance of adaptive task allocation.

7.3.2.2. Replacement Scenario

Adding an extra high-performing nurse to handle complex tasks significantly improved patient throughput and reduced delays, achieving faster service and smoother patient flow. However, the increase in staff resources slightly raised cumulative patient damage time and implied additional staffing costs.

7.3.2.3. Training Scenario

Pairing the low-performing nurse with a trainer improved throughput and led to gradual skill gains. While error rates initially increased – resulting in higher short-term patient damage – the overall system performance stabilized over time, reflecting the long-term benefits of capability building.

7.3.2.4. Replacement vs. Training

Replacement produced immediate gains in efficiency and patient flow, whereas Training supported gradual skill development of low-performing nurses. The decision between these strategies depends on managerial priorities: short-term operational optimization versus long-term workforce development. In a broader sense, these contrasting strategies represent complementary dimensions of personalized decision-making – one emphasizing immediate, patient-level optimization (Replacement) and the other focusing on long-term personalization of staff performance (Training).

7.3.3. Interpretation and Limitations

This case study is based on a specific configuration of doctors, nurses, and beds. Results may vary significantly under different conditions – for example, if doctor evaluation styles differ (overestimating or underestimating tasks) or if the nurse composition changes (e.g., two high-performing nurses). Additionally, the type of task being simulated can strongly influence outcomes. In this study, IV catheter insertion was used – a task where improvement of low-performing nurses occurs gradually over time in the Training scenario. For other tasks, such as vital sign measurement, improvement may occur rapidly. In such cases, even a single demonstration might suffice for a low-performing nurse to reach a high performance, potentially making the Training scenario more effective than Replacement across all performance indicators for that task. Therefore, the findings should be interpreted as illustrative of the framework's capabilities rather than universally generalizable conclusions. Nevertheless, the ability to parameterize agent-level variability – such as doctor judgment bias or nurse learning curves – positions the framework as a foundation for data-driven personalization of decision-support models in clinical operations.

7.3.4. Summary of Findings

The case study highlights how the simulation framework supports evidence-based decision-making by allowing hospital managers to explore trade-offs between patient safety, operational efficiency, and staff development under controlled, repeatable conditions. This demonstrates that policy selection depends on strategic priorities, and highlights the potential of the framework as a decision-support tool to test alternative scenarios before real-world implementation. Moreover, by integrating adaptive trust-based coordination with individualized agent modeling, the framework contributes to the

emerging vision of AI-driven personalized medicine, where operational decisions are continuously tuned to both staff performance and patient-specific needs.

7.4. Conclusions

This chapter extended the evaluation of the CA algorithm beyond trust management and examined its applicability as a decision-support mechanism within a simulated emergency department environment. The study investigated how trust-informed task allocation influences operational outcomes such as patient safety, service efficiency, and workforce development.

The experimental results demonstrated that trust-based coordination can substantially alter the behavior of the system. The Baseline scenario revealed a clear trade-off between efficiency and safety, with FIFO achieving higher throughput and lower waiting times, whereas the CA-based policy reduced task failures and patient risk by preventing low-performing nurses from undertaking demanding tasks. The Replacement and Training scenarios further showed that performance deficiencies can be addressed through either additional staffing resources or workforce development interventions, each offering distinct advantages and limitations.

More importantly, the study demonstrated that the CA mechanism can support adaptive decision making rather than merely trust management. By enabling agents to assess their own capabilities and modify their behavior accordingly, the framework provides a foundation for exploring alternative staffing and coordination policies under controlled and repeatable conditions. In this sense, trust operates not only as a mechanism for selecting service providers, but also as a means of supporting organizational decision making in complex and dynamic environments.

Overall, the results indicate that the proposed framework constitutes a promising tool for evaluating healthcare management strategies and exploring the consequences of operational interventions before their deployment in real clinical settings. The chapter therefore illustrates how computational trust mechanisms can be applied beyond traditional multi-agent domains and contribute to evidence-based decision support in healthcare operations.

7.5. Research Directions

The simulation framework presented in this chapter demonstrates the applicability of the CA algorithm as a decision-support mechanism in emergency department operations. However, the current implementation should be viewed primarily as a proof of concept. Several opportunities exist for extending the framework toward more realistic, adaptive, and data-driven healthcare applications. The following research directions are considered particularly promising.

7.5.1. Learning Agent Behavior Models from Logged Performance Data

In the current implementation, both doctor evaluation styles and nurse performance profiles are represented through predefined rule-based functions. While this approach facilitates controlled experimentation and reproducibility, it cannot fully capture the complexity and variability of real clinical environments.

A promising extension is to replace these hard-coded rules with data-driven models inferred from the simulation's own detailed performance logs or from real-world clinical data, integrating machine learning components to capture behavioral variability and allow agents to continuously adapt their decision strategies to evolving clinical contexts – a key feature of AI-driven personalization.

7.5.1.1. Inferring Doctor Evaluation Styles

Doctor-level logs include total patient time damage, cumulative delays before treatment, and the number of patients served. Analyzing the relationships among these metrics could allow inference of doctor evaluation styles. For instance, systematic underestimation of task difficulty may correlate with increased patient damage, while overestimation may inflate perceived nurse error rates. Machine learning methods such as classification, clustering, or Bayesian inference could estimate probability distributions over evaluation styles, leading to more realistic physician-agent behaviors.

7.5.1.2. Inferring Nurse Quality Attributes

Nurse-level logs record task outcomes (success/failure counts), time damage, delay impacts, and utility values. While the current framework categorizes nurses as simply high-performing or low-performing, this binary classification oversimplifies reality. Regression models, clustering approaches, or reinforcement learning could be employed to

infer richer nurse skill profiles, capturing variability due to experience, fatigue, training, or workload.

By adopting this dual-level learning approach, the framework could evolve into a self-adaptive platform in which agent parameters are continuously updated based on observed behavior, supporting more accurate predictive modeling and more credible decision-support analyses.

7.5.2. Environment and Workflow Customization

Future extensions should allow the emergency department (ED) environment to be dynamically configured based on hospital-specific layouts, staffing structures, and operational policies. Incorporating digital twins or blueprints would enable site-specific replication of bed capacity, nurse stations, and patient flow paths, enhancing the ecological validity of the simulation and its value as a decision-support tool.

7.5.3. Integrating a Triage Mechanism for Task Prioritization

Although the current simulation framework does not explicitly include a triage module, aspects of triage are implicitly represented through the doctor's evaluation of required performance levels, which reflect both patient acuity and task complexity. This implicit mechanism ensures that more demanding or urgent tasks are indirectly associated with higher performance requirements. However, the current CA algorithm does not prioritize tasks according to case severity or patient urgency. Future extensions of the framework could incorporate an explicit triage mechanism that dynamically orders tasks based on patient acuity and clinical urgency.

7.5.4. Expanding the Range of Nursing Tasks

This current framework models only one nursing intervention (IV catheter insertion). Extending it to include a wider range of routine ED tasks – such as medication administration, vital signs measurement, or ECG – would increase realism and enable comparative evaluation of staff allocation strategies. This would also allow hospital managers to explore how Replacement and Training policies perform across different task complexities, balancing efficiency with long-term workforce development.

7.5.5. Towards AI-Driven Personalized Healthcare Operations

The long-term vision of this research extends beyond emergency department simulation toward the development of adaptive healthcare management systems capable of supporting operational decision making in real time.

By combining computational trust, agent-based simulation, predictive analytics, and machine learning, future systems could continuously adapt staffing assignments, workload distribution, and resource allocation according to both patient needs and staff capabilities. Trust-aware coordination mechanisms could therefore serve as a bridge between operational management and intelligent decision-support technologies.

In this broader context, the CA algorithm may provide a foundation for AI-driven healthcare operations in which coordination policies evolve dynamically in response to changing environmental conditions, workforce characteristics, and patient demands. Such systems could contribute to safer, more efficient, and more adaptive healthcare environments while preserving the human-centered nature of clinical decision making.

8. Conclusions and Future Research

8.1. Overview of the Dissertation

The increasing complexity of modern distributed systems has amplified the importance of effective trust management mechanisms. In open environments, service providers may exhibit heterogeneous capabilities, their performance may change over time, and traditional trust models often struggle to adapt to dynamic conditions. Most existing approaches focus primarily on the perspective of the trustor, treating trust as a mechanism through which service requesters evaluate and select service providers. While such approaches have demonstrated significant value across a wide range of domains, they often overlook the potential contribution of the trustee in the trust formation process.

This dissertation investigated an alternative perspective inspired by biological mechanisms observed in the human brain. More specifically, it introduced the *Create Assemblies* (CA) approach, a trust management paradigm in which service providers actively participate in trust formation through self-assessment and adaptive behavior. Instead of relying exclusively on external evaluations, trustees continuously evaluate their own capabilities and use this information when deciding whether to undertake a requested task.

The research was structured around the design, improvement, evaluation, and application of the proposed CA approach. Initially, a biologically inspired trust model was developed and experimentally evaluated in open multi-agent environments. Subsequently, the model was extended through the introduction of self-assessment capabilities, enabling trustees to identify performance degradation and adapt their behavior accordingly. Comparative experiments were then conducted to investigate the strengths and limitations of both versions of the model under different environmental conditions.

Finally, the dissertation explored the applicability of the CA paradigm beyond traditional trust-management domains. A simulation framework representing a hospital emergency department was developed to examine how trust-informed task allocation can support operational decision making in complex environments. Through this case study, the proposed approach was evaluated not only as a trust model but also as a mechanism for adaptive coordination and decision support.

Taken together, the results presented throughout this dissertation demonstrate the feasibility of incorporating self-awareness and trustee participation into trust management processes. More broadly, they suggest that trust can be viewed not merely as an evaluation mechanism, but as a dynamic process of continuous adaptation involving both service requesters and service providers. The following sections summarize the principal contributions of the dissertation, discuss its main findings and limitations, and outline promising directions for future research.

8.2. Main Research Contributions

The research presented in this dissertation contributes to the field of computational trust by introducing a novel trustee-oriented trust-management paradigm, extending it through adaptive self-assessment mechanisms, investigating its behavior under a variety of environmental conditions, and exploring its integration with reinforcement learning and decision-support systems. The main contributions are summarized below.

Contribution 1: Introduction of a Trustee-Centric Trust Management Paradigm

The primary contribution of this dissertation is the introduction of a novel trust-management paradigm that shifts part of the trust formation process from the trustor to the trustee. Unlike conventional trust models, where service requesters evaluate and select service providers based on trust estimates, the proposed *Create Assemblies* (CA) approach enables trustees to actively participate in the decision-making process by assessing their own suitability for a requested task.

This trustee-centric perspective offers several important advantages. Since trust-related information is maintained locally by each trustee, agents can carry their trust knowledge across different environments, making the approach particularly suitable for highly dynamic and mobile systems. Furthermore, the model significantly reduces communication overhead because trustees do not need to exchange trust evaluations or recommendation information. The absence of recommendation exchange also makes the approach inherently resistant to reputation manipulation attacks such as bad-mouthing and ballot-stuffing, while simultaneously improving privacy protection by preventing the dissemination of interaction histories and private evaluations.

Furthermore, the dissertation provided a systematic evaluation of the CA paradigm with respect to a broader set of trust-management requirements and security-related

challenges. Beyond performance-oriented simulation experiments, the analysis examined properties such as decentralization, subjectivity, context awareness, dynamicity, availability, integrity, transparency, and explainability. The results showed that many of these properties emerge naturally from the trustee-centric architecture of CA, while also identifying several open challenges that require further investigation. This evaluation provides a deeper understanding of the conditions under which the proposed paradigm can be expected to operate effectively and identifies several directions for its future evolution.

This contribution establishes a fundamentally different perspective on computational trust, where trust is not solely the result of external evaluation but also incorporates the self-awareness and self-regulation capabilities of service providers.

Contribution 2: Development and Evaluation of the *Create Assemblies* (CA) algorithm for Open and Dynamic Multi-Agent Systems

The second contribution is the design and evaluation of the CA algorithm which operationalizes the trustee-centric trust paradigm in open and dynamic multi-agent systems. Experimental results demonstrated that the CA algorithm performs particularly well in environments characterized by changes in consumer populations. Under such conditions, the algorithm was shown to outperform the widely used FIRE trust model. At the same time, the evaluation revealed that FIRE maintains advantages in environments dominated by changes in provider populations.

These findings demonstrate that the proposed trustee-centric approach constitutes a viable alternative to traditional trust-management mechanisms and that different trust paradigms may be advantageous under different environmental conditions.

Contribution 3: Development of the CA algorithm with a self-assessment mechanism for the early detection of poor performance

The third contribution extends the original CA approach through the introduction of a self-assessment mechanism for the early detection of low performance. The resulting CA algorithm enables trustees to continuously evaluate their own performance and adapt their behavior when significant changes in their capabilities are detected.

This enhancement addresses one of the most challenging problems in trust management: radical changes in trustee performance profiles. By continuously monitoring task execution outcomes, trustees can identify performance degradation and proactively avoid accepting tasks for which they are no longer suitable. Experimental evaluation

demonstrated that the new CA algorithm provides substantial improvements in environments where provider capabilities change over time.

At the same time, the results revealed that neither the original nor the new version of the CA algorithm is universally superior. The original version remains advantageous in environments with stable provider performance, whereas the new version performs better when trustee profiles evolve dynamically. This observation motivates the future research direction of adaptive self-aware trustees capable of dynamically selecting the most appropriate version of the CA algorithm according to environmental conditions and their own situation.

Contribution 4: Adaptive Trust Model Selection through Reinforcement Learning

The fourth contribution is the introduction of an adaptive trust-management framework capable of dynamically selecting between trust paradigms according to environmental conditions.

The experimental findings obtained from the evaluation of CA and FIRE demonstrated that no single trust model consistently outperforms all others across all environmental conditions. To address this limitation, this dissertation introduced an adaptable trustor capable of dynamically selecting between trust mechanisms using Deep Reinforcement Learning.

A Deep Q-Network (DQN) was employed to learn a trust model selection policy from environmental observations derived from locally available information. Rather than relying permanently on a single trust model, the adaptable trustor continuously adapts its strategy according to the characteristics of the environment.

Experimental results showed that the proposed approach consistently achieved performance above that of the worst-performing trust model and frequently approached or exceeded the performance of the best available model. These findings suggest that trust management should not be viewed as a competition between alternative trust paradigms, but rather as a collaborative process in which different trust mechanisms can be dynamically combined according to environmental requirements.

To the best of our knowledge, this constitutes one of the first applications of Deep Reinforcement Learning to the problem of dynamic trust model selection between a trustor-centric and the trustee-centric trust model in Multi-Agent Systems. The results indicate that the trust management problem cannot be effectively addressed through a

single universal trust mechanism. Instead, trustee-centric and trustor-centric approaches should be viewed as complementary paradigms whose effectiveness depends on the characteristics of the operating environment.

Contribution 5: Application of Trustee-Centric Trust to Decision Support in Healthcare

The final contribution is the extension of a trustee-centric CA paradigm beyond traditional trust-management domains through its application to healthcare decision support.

A simulation framework representing a hospital emergency department was developed to investigate how self-assessment and trust-informed task allocation can support operational decision making in environments where errors may have severe consequences. Within this framework, healthcare personnel were modeled as autonomous agents capable of evaluating their own competence before undertaking clinical tasks.

The results demonstrated that trust-informed coordination can be used not only for trust management but also for adaptive task allocation and policy evaluation. The framework enables hospital managers to compare alternative staffing and training strategies under controlled conditions before applying them in real clinical environments. Furthermore, the approach illustrates how trustee self-assessment can contribute to the broader vision of adaptive and personalized healthcare operations, where decisions are continuously adjusted according to both staff capabilities and patient needs.

Collectively, these contributions establish the *Create Assemblies* approach as a novel trustee-oriented trust-management paradigm and demonstrate that trust management can benefit from both trustee-centric and trustor-centric reasoning. The dissertation further shows that adaptive mechanisms, including self-assessment and reinforcement learning, can be used to dynamically combine alternative trust strategies according to environmental conditions. Finally, the applicability of the proposed approach beyond traditional trust-management problems is demonstrated through its use as a decision-support mechanism in healthcare operations.

8.3. Limitations of the Research

Despite the promising results obtained throughout this dissertation, several limitations should be acknowledged. These limitations do not diminish the value of the

proposed contributions but rather define the scope within which the findings should be interpreted and highlight opportunities for future research.

First, the majority of the experimental evaluations were conducted within simulation environments. Although simulation provides a controlled and reproducible setting for studying trust-management mechanisms, it cannot fully capture the complexity and unpredictability of real-world socio-technical systems. Factors such as human behavior, organizational constraints, incomplete information, and unforeseen environmental events may influence trust dynamics in ways that are difficult to model accurately. Consequently, the reported results should be interpreted as evidence of the effectiveness of the proposed approaches under controlled experimental conditions rather than as guarantees of identical performance in operational deployments.

Second, the evaluation of the trustee-centric CA paradigm was primarily performed in comparison with the FIRE trust model. FIRE was selected because it is a well-known trust and reputation model for open multi-agent systems and because it is a representative model of the conventional trustor-centric approach, substantially different from the trustee-centric philosophy adopted by CA. Nevertheless, many other trust models have been proposed in the literature, each with different assumptions and strengths. The conclusions regarding the relative advantages of CA should therefore be interpreted within the context of the specific models examined in this dissertation.

Third, the reinforcement learning framework introduced in Chapter 4 was designed to learn dynamic selection between two trust-management approaches, namely FIRE and CA. While the results demonstrated that adaptive trust model selection can improve decision-making effectiveness, the learning framework was not evaluated with a broader set of trust models. As a result, the generality of the learned selection policies remains limited to the trust mechanisms considered in the current study.

The healthcare case study presented in Chapter 7 also has several limitations. The emergency department simulation was intentionally designed as a proof-of-concept environment to investigate the applicability of trustee-centric trust principles to decision-support problems. The framework models a limited number of healthcare roles, a single nursing task, and a simplified emergency department workflow. Important real-world factors such as staff fatigue, multitasking, shift transitions, heterogeneous patient conditions, and institutional constraints were not explicitly represented. Therefore, the

healthcare results should be viewed primarily as an illustration of the broader applicability of the CA paradigm rather than as a clinically validated decision-support solution.

Finally, while the dissertation demonstrates that different trust mechanisms may be advantageous under different environmental conditions, the proposed approaches do not yet provide a fully autonomous solution capable of continuously adapting both the trust model and the trustee behavior simultaneously. The development of such adaptive and self-aware trust-management systems remains an open challenge and constitutes one of the most promising directions for future research.

Overall, these limitations reflect the exploratory nature of the research and should be considered as opportunities for extending and strengthening the proposed framework rather than as weaknesses of the underlying concepts. The results obtained throughout the dissertation provide a solid foundation for future investigations into adaptive trust management, reinforcement learning-based trust selection, and trust-informed decision-support systems.

8.4. Future Research Directions

The research presented in this dissertation opens several promising directions for future work. While the proposed trustee-centric trust-management paradigm, its self-assessment extensions, and the reinforcement learning-based trust model selection framework demonstrated encouraging results, many challenges remain open. These challenges provide opportunities for extending the proposed approaches and further advancing adaptive trust management in dynamic environments.

8.4.1. Adaptive and Self-Aware Trustees

One of the most important directions emerging from this dissertation is the development of adaptive trustees capable of dynamically selecting the most appropriate trust-management strategy according to their current circumstances.

The experimental evaluation demonstrated that neither the original CA algorithm nor its self-assessment extension consistently outperforms the other across all environmental conditions. The original CA approach performs well when trustee performance remains relatively stable, whereas the enhanced version provides advantages when trustee capabilities change over time. These findings suggest that future trustees

could continuously monitor both environmental conditions and their own performance characteristics and dynamically switch between alternative CA variants when appropriate.

Such adaptive self-aware trustees would extend the notion of trustee-centric trust by enabling providers not only to assess their current competence but also to reason about the most suitable decision-making strategy for a given context.

8.4.2. Expanding Reinforcement Learning-Based Trust Selection

The reinforcement learning framework developed in this dissertation considered trust model selection between FIRE and CA. Future research could extend this approach to larger collections of trust models and more complex decision spaces.

Instead of selecting between only two alternatives, future adaptable trustors could learn policies over a broad portfolio of trust mechanisms, dynamically selecting the model that best matches current environmental conditions. Such an approach would move trust management closer to a meta-reasoning framework in which agents learn not only whom to trust, but also how trust itself should be computed.

Furthermore, more advanced reinforcement learning techniques, including policy-gradient methods, actor-critic architectures, and multi-agent reinforcement learning, could be investigated to improve adaptability in highly dynamic environments.

8.4.3. Learning-Based Trustee Self-Assessment

The self-assessment mechanisms proposed in this dissertation are currently based on predefined rules and local observations. A natural extension is the incorporation of machine learning techniques capable of learning trustee performance models directly from experience.

By analyzing historical interaction outcomes, trustees could learn richer representations of their own capabilities and predict future performance under different conditions. Such learning-based self-assessment mechanisms may enable earlier detection of performance degradation, improved adaptation to changing environments, and more accurate task acceptance decisions.

The integration of self-assessment with predictive learning represents an important step toward autonomous trust-aware agents capable of continuously improving their decision-making behavior.

8.4.4. Real-World Validation and Deployment

Although the proposed approaches were evaluated extensively through simulation, future work should investigate their applicability in real-world environments.

Validation using operational data from practical multi-agent systems would provide valuable insights into the strengths and limitations of the trustee-centric trust paradigm under realistic conditions. Such studies could involve distributed service systems, collaborative robotic environments, Internet of Things applications, or organizational decision-support systems.

Real-world deployment would also allow investigation of issues that are difficult to model in simulation, including incomplete information, communication failures, strategic behavior, and organizational constraints.

8.4.5. Trust-Informed Decision Support Systems

The healthcare case study demonstrated that the CA paradigm can be applied beyond traditional trust-management problems. This observation opens opportunities for applying trustee-centric trust principles to a wider range of decision-support domains.

Future research could investigate applications in healthcare, disaster management, autonomous transportation, smart manufacturing, and other complex socio-technical systems where decisions depend on the capabilities and reliability of participating agents. In such environments, trust-informed self-assessment may provide a mechanism for balancing efficiency, safety, robustness, and resource utilization.

8.4.6. Toward Integrated Trust Management Frameworks

Perhaps the most significant conclusion of this dissertation is that no single trust-management approach appears to be universally optimal. Different trust models exhibit different strengths under different environmental conditions.

Consequently, future research should move beyond the search for a single superior trust model and focus instead on the development of integrated trust-management frameworks capable of combining multiple trust paradigms. Within such frameworks, trustee-centric and trustor-centric approaches would operate collaboratively, with adaptive mechanisms dynamically selecting, weighting, or combining trust information according to the current context.

The long-term vision emerging from this dissertation is therefore not the replacement of conventional trust models by the CA approach, but rather the incorporation

of trustee-centric trust as a complementary component within broader adaptive trust-management architectures. Such systems would be capable of continuously adapting to environmental changes, agent behavior, and application requirements, ultimately providing more robust and effective trust-based decision making in dynamic environments.

Appendix A. Synaptic Plasticity

This appendix presents the basic biological mechanisms of neuronal communication and synaptic plasticity.

A.1. Action Potential

When a neuron receives sufficient stimulation, an electrical signal is transmitted along the entire length of its axon. This phenomenon is known as an *action potential*, and the neuron is said to fire.

The transmission of a neuronal signal depends on the movement of ions (charged particles). Various ions, including sodium (Na^+), potassium (K^+), and Chloride (Cl^-) are unequally distributed between the interior and exterior of the neuron. When a neuron is not transmitting a signal, it is said to be at rest. In this state, the concentration of sodium ions is higher outside the cell than inside, whereas the concentration of potassium ions is higher inside the cell than outside. This unequal ionic distribution occurs across the cell membrane and creates a concentration gradient. As a result, the concentration of positively charged ions is greater outside the cell than inside (Figure 56).

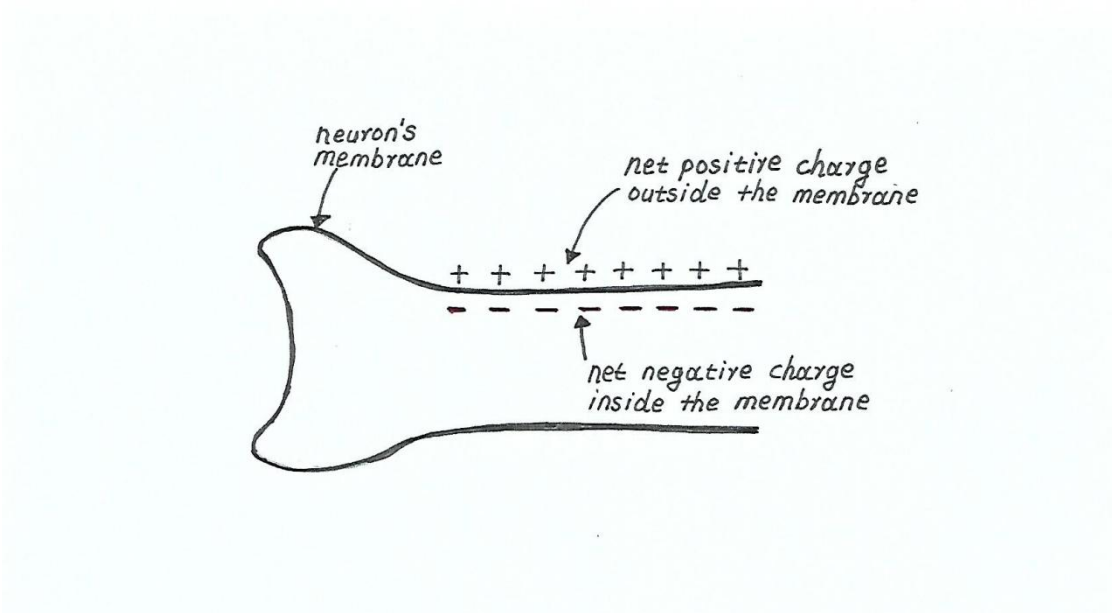


Figure 56: At rest, the concentration of positively charged ions is higher outside the cell than inside

This unequal distribution of ions also creates a difference in electrical charge across the membrane, known as the *electrical gradient*. The difference in electrical charge

between the inside and outside of the cell is referred to as the *membrane potential*. In a resting neuron, the membrane potential is approximately -70 mV, meaning that the interior of the cell is about 70 mV less positive than the exterior. At this point, the neuron is considered to have reached electrochemical equilibrium.

Ions cannot freely cross the cell membrane. Instead, their movement is facilitated by specialized proteins embedded within the membrane. Most ions cross the membrane through structures known as *ion channels*, moving by passive diffusion along their concentration gradients. Some ion channels remain permanently open, whereas others open or close in response to specific stimuli.

For example, *voltage-gated ion channels* open only when the membrane potential reaches a particular value. *Ligand-gated ion channels* are activated when a specific molecule binds to them, while *mechanically gated ion channels* respond to physical forces such as stretching or pressure. Most ion channels are selectively permeable, allowing only one or a limited subset of ions to pass through.

As ions move through these channels and cross the membrane, they alter the membrane potential. If the resulting change in membrane potential is relatively small, the signal is referred to as a *graded potential*. Graded potentials are transient, and the neuron rapidly returns to its resting state. This restoration is achieved primarily through the activity of the *sodium-potassium pump*, which continuously transports three sodium ions out of the cell and two potassium ions into the cell during each cycle. This unequal exchange contributes to the maintenance of the ionic gradients observed in resting neurons.

Only when the resting membrane potential and ion distributions are maintained within precise limits is the neuron capable of generating an action potential. When external stimulation is sufficiently strong to raise the membrane potential from approximately -70 mV to the threshold value of -55 mV, an action potential is initiated at the *axon hillock* and subsequently propagates along the axon.

An action potential consists of several phases, including *depolarization*, *repolarization*, and *hyperpolarization*. Voltage-gated sodium channels can exist in three states: closed, open, and inactivated. In the resting state, these channels remain closed. Once the membrane potential reaches the threshold value, the channels open and sodium ions rapidly enter the cell due to the electrochemical gradient. As positively charged

sodium ions enter the neuron, the membrane potential becomes progressively less negative and approaches 0 mV. This phase is known as *depolarization*.

The membrane potential may eventually reach approximately +30 mV, a phase referred to as the *overshoot*. As the membrane potential becomes positive, the sodium channels enter their inactivated state, preventing further sodium influx.

At the same time, the change in membrane potential triggers the opening of voltage-gated potassium channels. These channels open more slowly than sodium channels. Because of the potassium electrochemical gradient, potassium ions flow out of the cell, causing the membrane potential to return toward negative values. This phase is known as *repolarization*.

Since potassium channels close relatively slowly, the membrane potential briefly becomes more negative than its resting value. This phase is known as *hyperpolarization*. During this period, the potassium channels gradually close while the sodium-potassium pump continues to operate, restoring the ionic gradients and returning the membrane potential to its resting value.

During this period, the inactivated sodium channels cannot respond to further stimulation. Consequently, the neuron enters its *absolute refractory period*, a brief interval during which a second action potential cannot be generated regardless of the intensity of stimulation. The absolute refractory period prevents the generation of successive action potentials within a very short time interval and ensures the proper propagation of neuronal signals.

A.2. Synaptic Plasticity

Synaptic plasticity refers to the continuous modification and reorganization of neuronal connections in the human brain. The fundamental principle underlying this process is that when two neurons fire simultaneously or in a coordinated manner, the connection between them is strengthened, increasing the likelihood that they will fire together again in the future. Conversely, when two neurons repeatedly fire in an uncoordinated manner, the connection between them weakens, making future coordinated activity less likely. In simple terms, cells that fire together wire together, whereas cells that fire apart wire apart.

However, what do we mean by the strength of a connection? When an action potential (Figure 57) reaches the terminal of a presynaptic neuron, it triggers the release of neurotransmitters, which diffuse across the synaptic cleft and bind to receptors on the postsynaptic cell. These receptors may be either metabolic receptors or ion channels (Figure 57).

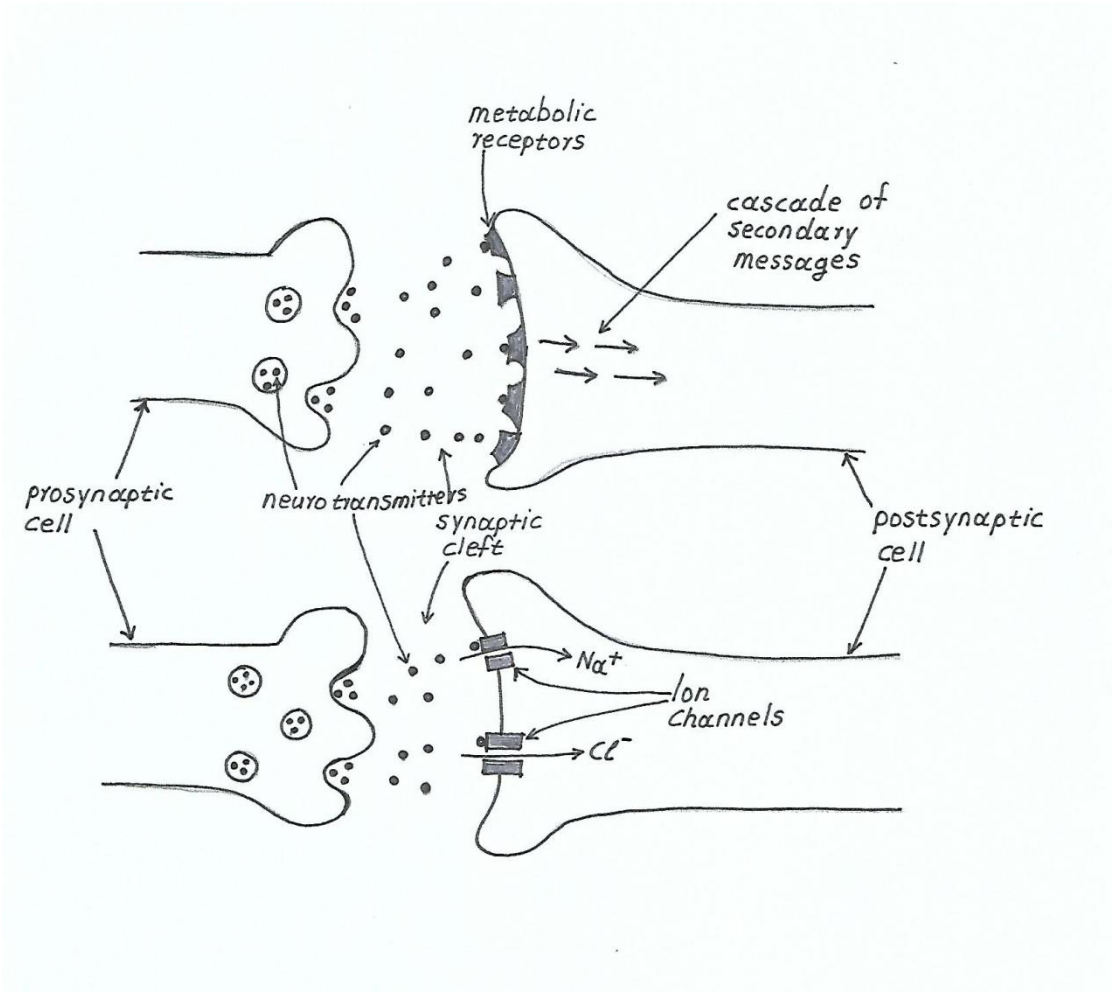


Figure 57: When an action potential reaches the end of a presynaptic cell, it causes the release of neurotransmitters, which cross the synaptic cleft and bind to receptors on the postsynaptic cell

When a neurotransmitter activates a metabolic receptor, it triggers a cascade of intracellular signaling processes that may alter a variety of proteins within the cell. In contrast, when a neurotransmitter activates an ion channel, it may cause the channel to open or close, thereby altering the flow of ions through the membrane and inducing a change in the membrane potential of the postsynaptic cell. The number of receptors present on the postsynaptic membrane – and consequently the magnitude of the voltage

change produced in the postsynaptic neuron – is what we refer to as the strength of a synaptic connection (Figure 58). The strengthening of a synapse is known as *long-term potentiation (LTP)*, whereas the weakening of synaptic strength is referred to as *long-term depression (LTD)*. Whether a synapse undergoes LTP or LTD depends primarily on the precise timing of neuronal activity.

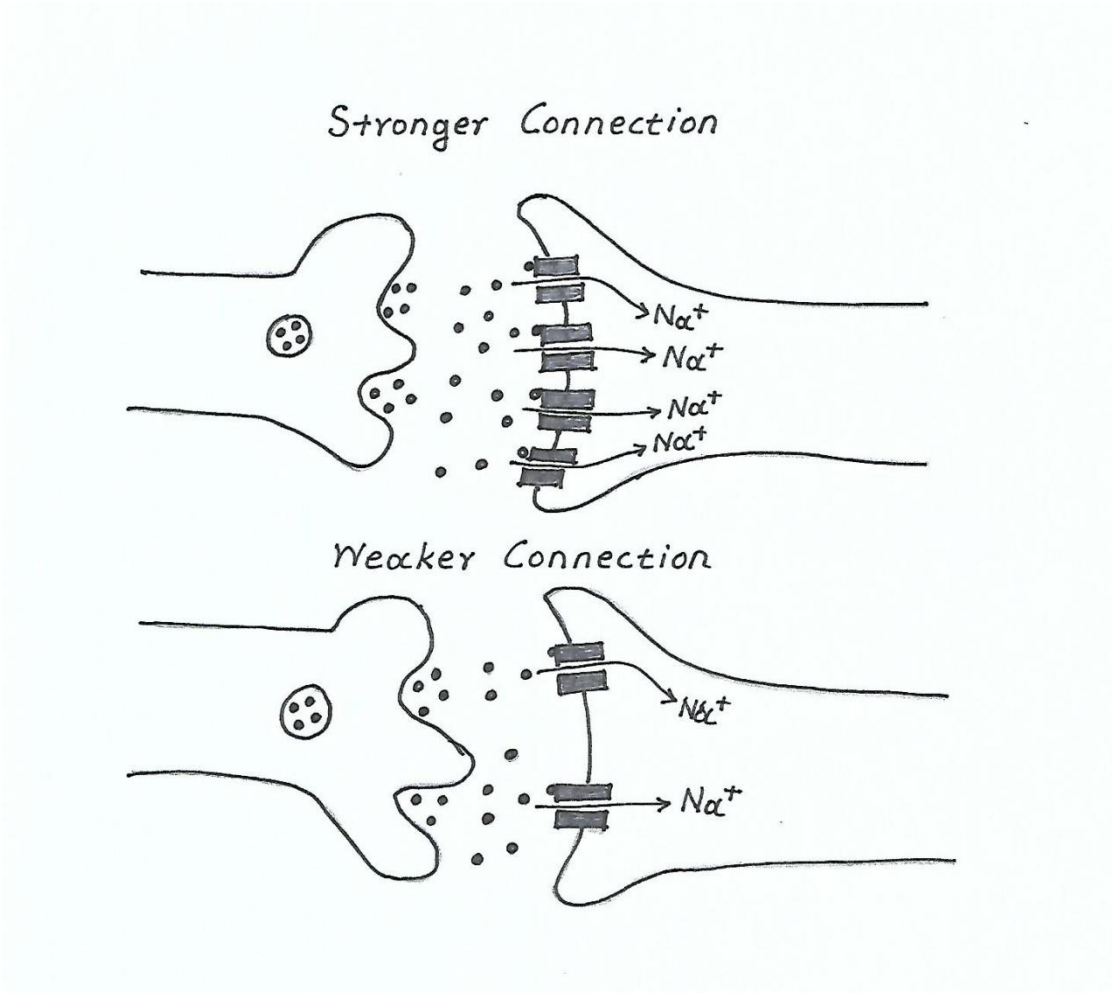


Figure 58: In the stronger connection, the number of receptors on the postsynaptic cell is higher, and therefore the degree of voltage change in the postsynaptic cell is greater, compared to the weaker connection

There exists a critical temporal window for synaptic plasticity, with the most significant changes in synaptic strength occurring within approximately 20ms before or after an action potential. If the presynaptic neuron fires up to 20 ms before the postsynaptic neuron, LTP occurs and the synapse is strengthened. Conversely, if the presynaptic neuron fires after the postsynaptic neuron within the same temporal window,

LTD occurs and the synapse is weakened. This phenomenon is known as *spike-timing-dependent plasticity (STDP)*.

Glutamate receptors play a central role in this process. Glutamate is the principal excitatory neurotransmitter in the brain, and two glutamate receptor types are particularly important: AMPA receptors and NMDA receptors. AMPA receptors are permeable to sodium and potassium ions, and it is the resulting inward flux of positively charged ions through these receptors that depolarizes the postsynaptic cell (Appendix A.1). In contrast, NMDA receptors are blocked by magnesium ions at negative membrane potentials and therefore do not significantly contribute to postsynaptic depolarization under resting conditions. However, once the cell becomes sufficiently depolarized, the magnesium block is removed, allowing ions to flow through the NMDA receptors.

Importantly, NMDA receptors are also permeable to calcium ions. The characteristics of this calcium influx are responsible for spike-timing-dependent plasticity. When the presynaptic neuron fires first, it becomes depolarized and releases glutamate. By the time glutamate reaches the postsynaptic neuron, the latter is in the process of repolarizing and is therefore at a lower membrane potential. Under these conditions, fewer NMDA receptors are available for activation, resulting in a relatively moderate calcium influx. A large calcium influx induces long-term potentiation (LTP), whereas a smaller calcium influx results in long-term depression (LTD).

AMPA receptors are continuously recycled within the cell. New receptors undergo exocytosis at perisynaptic sites and subsequently migrate to postsynaptic regions. At the same time, receptors located at postsynaptic sites migrate back to perisynaptic regions, where they undergo endocytosis and are internalized into the cell.

A calcium influx that exceeds a critical threshold activates *calcium-dependent kinases*. These kinases modify the recycling dynamics of AMPA receptors, primarily by increasing their exocytosis. In addition, they phosphorylate AMPA receptors, altering their structure and increasing their permeability. Consequently, when glutamate crosses the synapse, a greater number of AMPA receptors are available to open, more current flows through the membrane, and the resulting change in membrane potential is amplified.

By contrast, a more moderate calcium influx does not reach the threshold required to activate calcium-dependent kinases and instead activates *protein phosphatases*. These enzymes influence AMPA receptor recycling in the opposite direction by increasing

receptor endocytosis, thereby reducing the number of AMPA receptors available at the postsynaptic membrane. As a result, when glutamate crosses the synapse, fewer receptors are available to open, less current flows through the membrane, and the resulting change in membrane potential is diminished.

Appendix B. Reinforcement Learning Background

This appendix provides a concise overview of the Reinforcement Learning (RL) concepts that underpin the adaptive trust model selection mechanism presented in Chapter 4. The objective is not to provide a comprehensive treatment of RL, but rather to introduce the fundamental principles of Q-Learning and Deep Q-Learning that are employed by the proposed adaptable trustor.

B.1. Reinforcement Learning

Reinforcement Learning is a machine learning paradigm in which an agent learns how to act in an environment through trial-and-error interactions. Rather than being explicitly instructed which actions to perform, the agent receives feedback in the form of rewards and gradually learns a policy that maximizes long-term utility.

The classical formulation of single-agent Reinforcement Learning under full observability is based on the Markov Decision Process (MDP) framework introduced by Sutton and Barto [118]. An MDP is defined by the tuple:

$$\langle S, A, T, R, \gamma \rangle$$

where:

- S denotes the set of states,
- A denotes the set of actions,
- $T(s, a, s')$ defines the state transition probabilities,
- $R(s)$ represents the reward function,
- $\gamma \in [0,1)$ is the discount factor.

At time step t, the agent observes the current state $s_t \in S$, selects an action $a_t \in A$ according to a policy $\pi(a|s)$, and receives a reward r_t. The environment then transitions to a new state s_{t+1} according to the transition model T.

The objective of the agent is to learn an optimal policy π^* that maximizes the expected cumulative discounted reward, also known as the return:

$$R_t = \sum_{t'=t}^{H} \gamma^{(t'-t)} r_t$$

where H denotes the decision horizon.

A central concept in RL is the action-value function $Q(s,a)$, which estimates the expected return obtained by executing action a in state s and subsequently following policy π:

$$Q^{\pi}(s,a) = E[R_t|S_t = s, A_t = a]$$

The optimal policy is the policy that maximizes the action-value function:

$$\pi^* = arg\max_{\pi} Q^{\pi}(s,a)$$

and the corresponding optimal action-value function is denoted by $Q^*(s,a)$.

Reinforcement Learning methods can generally be categorized as either model-based or model-free. Model-based approaches attempt to learn the transition and reward functions before deriving an optimal policy. In contrast, model-free methods learn policies or value functions directly from experience without explicitly modeling the environment. Model-free approaches are often preferred in complex environments where the transition dynamics are unknown or difficult to estimate.

B.2. Q-Learning

Q-Learning, introduced by Watkins and Dayan [117], is one of the most widely used model-free RL algorithms.

The algorithm learns the optimal action-value function iteratively using the Bellman optimality equation. Following an observed transition (s,a,r,s'), the Q-value is updated according to:

$$Q(s,a) \leftarrow Q(s,a) + a\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

where:

- $a \in [0,1]$ is the learning rate,
- r is the immediate reward,
- γ is the discount factor.

The quantity

$$r + \gamma \max_{a'} Q(s',a') - Q(s,a)$$

is known as the Temporal Difference (TD) error and represents the discrepancy between the current estimate and the newly observed information.

For problems with relatively small state and action spaces, Q-values can be stored explicitly in a lookup table. Under appropriate conditions, tabular Q-Learning is guaranteed to converge to the optimal action-value function Q^*.

However, in many real-world problems the state space is extremely large or continuous, making tabular representations impractical. In such cases, function approximation techniques are commonly employed.

B.3. Deep Q-Learning

Deep Q-Learning [158] extends traditional Q-Learning by replacing the tabular representation of the action-value function with a neural network known as a Deep Q-Network (DQN).

Instead of storing a Q-value for every state-action pair, the network learns a parameterized approximation:

$$Q(s, a; \theta)$$

where θ denotes the neural network parameters.

For each experience tuple

$$\langle s, a, r, s' \rangle$$

the DQN computes a target value:

$$y_t = r_t + \gamma \max_a \bar{Q}(s', a'; \bar{\theta})$$

using a separate target network

$$\bar{Q}(s, a; \bar{\theta})$$

whose parameters are updated less frequently.

The online network produces the current estimate:

$$Q(s, a; \theta)$$

and the training objective is to minimize the squared loss:

$$L(\theta) = \big(y_t - Q(s, a; \theta)\big)^2$$

The use of separate online and target networks improves learning stability by reducing oscillations caused by rapidly changing target values.

Deep Q-Learning also employs an experience replay mechanism. Each interaction experience is stored in a replay memory, and training is performed using randomly

sampled mini-batches rather than sequential observations. This process reduces correlations between training samples and improves convergence.

The combination of neural-network-based value approximation, replay memory, and target networks has made Deep Q-Learning one of the most influential reinforcement learning algorithms and enables its application to problems involving large and complex state spaces.

B.4. Relation to Chapter 4

The adaptable trustor proposed in Chapter 4 formulates trust model selection as a Reinforcement Learning problem. The environmental features described in Section 4.3 constitute the state representation, while the available actions correspond to selecting either the FIRE or the CA trust model. The utility gain obtained after each interaction serves as the reward signal.

Because the state space generated by the environmental features is too large for a tabular representation, Deep Q-Learning is employed to approximate the optimal action-value function and learn a policy capable of adapting trust model selection to changing environmental conditions.

Appendix C. Selected Proof Sketches

Proof of Proposition 1. It suffices to show by mathematical induction that the n-th connection for $task_1$ will be initialized to the value of 0.45.

Base Case. $(n = 2)$. We know that BP_1 has initialized its second connection for co_2 to the value 0.45.

Induction hypothesis: Assume that Proposition 1 is correct for the kth connection that BP_1 will create for $task_1$.

Induction step: We will show that Proposition 1 holds for the $(k + 1)$th connection that BP_1will create for $task_1$.

BP_1 will calculate the average of its k connections for $task_1$as follows:

$$average = \frac{w_{co_1} + w_{co_2} + \cdots + w_{co_{k-1}} + w_{co_k}}{k} \quad (3)$$

where w_{co_n} denotes the weight of the n-th connection of $BP_1$for $task_1$.

From the induction hypothesis, we have the following:

$$w_{co_k} = 0.45 \quad (4)$$

The average of the weights of the $k - 1$ connections is equal to 0.45:

$$\frac{w_{co_1} + w_{co_2} + \cdots + w_{co_{k-1}}}{k-1} = 0.45 \Rightarrow w_{co_1} + w_{co_2} + \cdots + w_{co_{k-1}} = (k-1) \cdot 0{,}45 \quad (5)$$

From (3)-(5), we have the following: $average = \frac{(k-1)\cdot 0.45 + 0.45}{k} = 0.45$

Appendix D. Sketch of Computational Complexity Analysis of CA Algorithm

This sketch highlights the dominant cost contributors. A full formal proof would require additional assumptions and details regarding data structures and implementation models.

Time Complexity Analysis. We analyze the time complexity per iteration of the main loop of Algorithm 3, focusing on dominant, input-dependent operations. Constant time steps and scalar updates are noted but not emphasized in the final complexity.

Notation:

- t: Number of distinct known tasks (c, r) stored or processed by the agent, where:
  - c is a task category.
  - r is a set of requirements.
- n: Number of agents in the system.
- q: Number of incoming messages in the agent's message list.
- m: Number of connections currently held by the agent. In the worst case, $m = O(n \cdot t)$.

Step-by-Step Breakdown

1. Initialization:
   - Creating an empty structure: $O(1)$.
2. Message Handling and Connection Setup:
   - Broadcast to all agents: $O(n)$.
   - Append message: $O(1)$.
   - Connection handling:
     - Worst case:
       - Linear search in connections: $O(m)$.
       - Find similar connections and average weights: $O(m)$.

     Total: $O(m)$.
   - Alternative case: $O(1)$.

   Overall block cost: $O(n + m)$.

3. Select and Attempt Task:
    - For each message (q in total), find the corresponding connection (worst-case linear search): $O(q \cdot m)$.
    - Other steps (task status, threshold check, performance evaluation): Each is $O(1)$.
    - Message deletion (search + shift in list): $O(q)$.

    Total for this block: $O(q \cdot m + q) = O(q \cdot m)$.
4. Update Connection Weight: Direct updates and conditionals: $O(1)$.
5. Dynamic Profile Update:
    - Iterate and filter all connections: $O(m)$.
    - Search through task list: $O(t)$.

    Total: $O(m + t)$.

Total Time Complexity: $O(n + m + q \cdot m + m + t) = O(n + q \cdot m + t) = O(q \cdot n \cdot t)$, since, as stated, in the worst case, it is $m = O(n \cdot t)$.

Space Complexity. We estimate the memory requirements per agent, focusing on the key data structures and how they scale with input parameters:

- $i.bad_tasks$: Map from tasks to Boolean: $O(t)$.
- Message list M: Up to q messages: $O(q)$.
- Connection list: At most one per known task per agent: $O(n \cdot t)$.
- Local scalars and flags: $O(1)$.

Total Space Complexity: $O(q + n \cdot t)$.

Appendix E. Algorithms

E.1. Doctor's performance level estimation based on evaluation style

```
Algorithm 4. Doctor's performance level estimation based on evaluation style
function EvaluatePerformanceLevel(trueLevel, evaluationStyle):
	switch(evaluationStyle):
		case "overestimates":
			if trueLevel <= 2: return 3
			if trueLevel >= 3: return 5

		case "underestimates":
			if trueLevel >= 4: return 3
			if trueLevel <= 3: return 1

		case "estimatesCorrectly":
			return trueLevel
		default:
			return trueLevel
```

E.2. Nurse task execution time determination based on quality and training status

```
Algorithm 5. Nurse task execution time determination based on quality and
training status
function GetTaskDuration(nurseId, truePerformanceLevel, observedTasks):
roll = random(0, 1)

       if nurseId == 1:  //Low-performing nurse
              isTrainingScenario = (methodToChooseRequest == 1 AND Scenario == 3)

              if isTrainingScenario:
                     bonusChance = clamp(observedTasks × 0.1, 0, 1) // each
              observed task adds 10%

                     if roll <= bonusChance:
                            return baseDurationForLevel(truePerformanceLevel)
              else:
                     return randomDurationAboveBase(truePerformanceLevel)
       else:
              return randomDurationAboveBase(truePerformanceLevel)

       else if nurseId == 2: // High-performing nurse
              rollGood = random(0, 1)

              if rollGood <= 0.90:
                     return baseDurationForLevel(truePerformanceLevel)
              else:
                     return randomDurationAboveBase(truePerformanceLevel)

       return 40 // default fallback


function baseDurationForLevel(level):
       switch(level):
              case 1: return 60
              case 2: return 50
              case 3: return 40
              case 4: return 30
              case 5: return 20
              default: return 40

function randomDurationAboveBase(level):
       // returns a duration in a higher range to simulate delay
       switch(level):
              case 1: return random(60, 70)
              case 2: return random(50, 70)
              case 3: return random(40, 50)
              case 4: return random(30, 40)
              case 5: return random(20, 30)
              default: return 40
```

Appendix F. Decision Support Case Study

This appendix provides full methodological and statistical details for the decision-support case study summarized in chapter 7. All data correspond to 60-run simulation averages and include statistical test results (Shapiro-Wilk, Wilcoxon rank-sum, Monte Carlo x^2, etc.).

F.1. Comparative Evaluation of Nurse Task Selection Methods in the Baseline Scenario

Objective

This case study investigates how two alternative nurse task selection policies affect patient care quality and operational performance metrics in the Baseline scenario:

- Method 1: CA trust model;
- Method 2: FIFO.

Design

- Scenario: Baseline (no replacement nurses, no training interventions);
- Task selection policies: CA trust model vs. FIFO;
- Simulation runs: 60 independent runs per method;
- Metrics analyzed: Total patients served, total patient damage time, total patient delay, failed and successful tasks per nurse, nurse utility, nurse time damage, patient distribution across doctors.

Statistical Approach

All metrics were tested for normality using the Shapiro-Wilk test. Due to significant deviations from normality in most cases ($p<0.001$), non-parametric Wilcoxon rank-sum tests were applied unless otherwise specified. Effect sizes, medians, means and p-values are reported for each comparison.

Key Results

Total Patients Served: Since both groups exhibited significant deviations from normality (Shapiro Wilk $p<0.001$), the non-parametric Wilcoxon rank-sum test was applied. The analysis showed that the FIFO policy served significantly more patients than the CA trust model ($W = 0, p < 2.2 \times 10^{-16}$). The mean difference was approximately

11 patients per shift (CA trust: 22.18, FIFO: 33.22), indicating that prioritizing tasks by arrival order rather than trust scores substantially increases throughput in the Baseline scenario.

Total Patient Damage Time: Both groups deviated significantly from normality (Shapiro-Wilk p<0.001), so the non-parametric Wilcoxon rank-sum test was applied. The CA trust model resulted in dramatically lower cumulative patient damage time compared to the FIFO policy ($W = 0, p < 2.2 \times 10^{-16}$). On average, the damage time was reduced by nearly 200 seconds per shift (CA trust: 65.38 s, FIFO: 262.77 s), indicating that trust-based task allocation substantially mitigates patient exposure to harmful delays in treatment.

Total Patient Delay: Both groups showed strong deviations from normality (Shapiro-Wilk p<0.001), so the Wilcoxon rank-sum test was used. FIFO produced significantly lower total patient delay compared to the CA trust model ($W = 90000, p < 2.2 \times 10^{-16}$). The mean reduction was about 642 seconds per shift (CA trust: 7095.77 s, FIFO: 6453.23 s), suggesting that FIFO's task allocation leads to shorter waiting periods for patients before nursing intervention. However, as earlier results show, this comes at the cost of increased patient damage time.

Failed Tasks by the Low-performing Nurse: Both groups exhibited strong deviations from normality (Shapiro-Wilk, p<0.001), so the non-parametric Wilcoxon rank-sum test was applied. Results showed that the low-performing nurse failed significantly more tasks under FIFO (Method 2) than under the CA trust model (Method 1) ($W = 3600, p < 2.2 \times 10^{-16}$). The mean failure count was 13.53 for Method 2 and 3.60 for Method 1, a difference of nearly +10 failed tasks per shift. This confirms that FIFO's inability to avoid high-difficulty tasks for low-performing nurses leads to a substantial increase in failed task executions.

Failed Tasks by the High-performing Nurse: Normality tests indicated significant deviations from the normal distribution in both groups (Shapiro-Wilk p<0.001), so the Wilcoxon rank-sum test was used. The test did not reveal a statistically significant difference in failed tasks between FIFO and the CA trust model ($W = 1852.5, p = 0.389$). Mean failure counts were very similar (Method 1: 1.92, Method 2: 1.98), and the distributions showed substantial overlap. These results suggest that high-performing nurses maintain a low and stable error rate regardless of the task allocation policy.

Successful Tasks by the Low-performing Nurse: Both groups contained only zero values for the number of successful tasks, making normality testing infeasible and rendering statistical comparisons trivial. The Wilcoxon rank-sum test confirmed no difference between Method 1 (CA trust model) and Method 2 (FIFO) ($W = 1800, p = 1$). In other words, low-performing nurses were unable to complete any tasks successfully under either task allocation policy, highlighting that policy choice alone does not improve task success when nurse performance is intrinsically poor.

Successful Tasks by the High-performing Nurse: Shapiro-Wilk tests indicated that Method 1 data significantly deviated from normality ($p<0.001$), while Method 2 data showed no significant deviation ($p=0.075$). Given the distributional mismatch, the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference in the number of successful tasks between Method 1 (median = 17) and Method 2 (median = 18) ($W = 1853, p = 0.391$). These findings suggest that highly skilled nurses maintain similar success rates regardless of whether patient allocation follows the CA trust model or a FIFO policy.

Nurse Utility by the Low-performing Nurse: Shapiro-Wilk tests indicated that Method 1 data significantly deviated from normality ($p=0.006$), while Method 2 data showed no significant deviation ($p=0.517$). Given the distributional mismatch, the non-parametric Wilcoxon rank-sum test was applied. Results showed that the low-performing nurse had significantly lower utility in Method 2 (median = -44.50) compared to Method 1 (median = -9.97) ($W = 0, p < 2.2 \times 10^{-16}$). These findings suggest that the low-performing nurse performs substantially worse under Method 2 conditions.

Nurse Utility by the High-performing Nurse: Shapiro-Wilk tests indicated that Method 1 data significantly deviated from normality ($p=0.027$), while Method 2 data showed no significant deviation ($p=0.916$). Given the distributional mismatch, the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference in nurse utility between Method 1 (median = 50.28) and Method 2 (median = 53.25) ($W=1749$, $p=0.607$). These findings suggest that highly skilled nurses maintain similar utility regardless of whether patient allocation follows Method 1 or Method 2.

Nurse Time Damage by the Low-performing Nurse: Shapiro-Wilk tests indicated that Method 1 data significantly deviate from normality ($p=0.002$), while Method 2 data showed no significant deviation ($p=0.582$). Given the distributional mismatch, the non-

parametric Wilcoxon rank-sum test was applied. Results showed a significant difference in NurseTimeDamage between Method 1 (median = 56.48) and Method 2 (median = 252.37) ($W = 0, p < 2.2 \times 10^{-16}$). These findings suggest that low-performing nurses cause substantially lower time damage when patient allocation follows Method 1 compared to Method 2.

Nurse Time Damage by the High-performing Nurse: Shapiro-Wilk tests indicated that NurseTimeDamage data for both Method 1 (p=0.003) and Method 2 (p=0.003) significantly deviated from normality. Given the non-normal distributions, the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference in NurseTimeDamage between Method 1 (median = 8.87) and Method 2 (median = 10.32) (W=1679, p=0.263). These findings suggest that highly skilled nurses maintain similar time damage regardless of whether patient allocation follows Method 1 and Method 2.

Patient Distribution Across Doctors: Nurses prefer some doctors in Method 1, while patients are served uniformly in Method 2. Using the Monte Carlo x^2 test for each simulation run, in Method 1, 18 out of 60 runs showed a statistically significant nurse preference for certain doctors (p<0.05). In contrast, in Method 2, none of the runs showed a significant preference. The variance in the number of patients served per doctor per run was much higher in Method 1 (mean variance = 21.44) compared to Method 2 (mean variance = 0.6). A paired t-test confirmed that this difference is statistically significant $p \approx 1.46e - 09$). The conclusion is that nurses appear to favor certain doctors in Method 1, resulting in an uneven distribution of patients. In Method 2, patients are distributed uniformly across doctors, as expected.

Conclusions

Method 1 (CA trust model) prioritizes patient safety and reduces the risk from low-performing nurses but can result in longer patient delays and potential doctor preference bias.

Method 2 (FIFO) improves fairness and reduces delays but increases exposure to nurse-related damage.

These results highlight a trade-off between safety and operational efficiency. Hospitals can select the appropriate method depending on their priorities: safety-focused institutions may prefer Method 1, while those emphasizing throughput and fairness may prefer Method 2.

F.2. Baseline (CA trust model) vs. Replacement (CA trust model)

Objective

Evaluate the impact of introducing a high-performing replacement nurse for tasks exceeding the difficulty threshold of low-performing nurses.

Design

- Scenarios: Baseline vs. Replacement;
- Task selection policy: CA trust model (applied in both scenarios);
- Metrics analyzed: Total patients served, total patient damage time, total patient delay, failed and successful tasks per nurse, nurse utility, nurse time damage, patient distribution across doctors.

Statistical Approach

All metrics were tested for normality using the Shapiro-Wilk test. Due to significant deviations from normality in most cases, non-parametric Wilcoxon rank-sum tests were applied. For metrics with no variance, no statistical test was meaningful. Effect sizes, medians, means, and p-values are reported for each comparison.

Key Results

Total Patients Served: Since both groups exhibited significant deviations from normality (Shapiro-Wilk p<2.2e-16), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario served significantly more patients ($W = 1935, p < 2.2 \times 10^{-16}$) compared to the baseline. The mean difference was approximately 19.4 patients per shift (Baseline: 22.18, Replacement: 41.58), indicating that adding an extra nurse to handle high-difficulty tasks substantially increases the overall patient throughput.

Total Patient Damage Time: Since both groups showed significant deviations from normality (Shapiro-Wilk test: Baseline $p = 7.29 \times 10^{-9}$, Replacement $p = 1.63 \times 10^{-4}$, the non-parametric Wilcoxon rank-sum test was applied. Results revealed that the Replacement scenario had a significantly higher total time in damage (W=27360, $p < 2.2 \times 10^{-16}$) compared to the Baseline. The mean difference was approximately 30.6 seconds per shift (Baseline: 65.38 sec, Replacement: 95.97 sec), indicating that the staffing change, while improving throughput, also increased the cumulative time that patients spent in a critical condition.

Total Patient Delay: Shapiro-Wilk tests indicated that both the Baseline (CA trust) data (W=0.787, p<0.001) and the Replacement (CA trust) data (W=0.652, p<0.001) significantly deviated from normality. Given the non-normal distributions, the non-parametric Wilcoxon rank-sum test was applied, revealing a significant difference in total patient delay between the two scenarios ($W = 106605,\ p < 0.001$). Specifically, the Replacement scenario exhibited a substantially lower mean total delay (M=6284.12 s) compared to the Baseline scenario (M=7095.77 s) representing an average reduction of approximately 811.65 seconds per shift. These findings suggest that adding an extra nurse to handle high-difficulty tasks significantly decreases overall patient delay in the emergency room.

Failed Tasks by the Low-performing nurse: Since both groups exhibited significant deviations from normality (Shapiro-Wilk p<0.001), the non-parametric Wilcoxon rank-sum test was applied. Results indicated a small difference in the number of failed tasks between the Baseline and Replacement scenarios (W=1416.5, p=0.037), with the mean number of failed tasks being slightly higher in the Replacement scenario (Baseline: 3.6, Replacement: 4.0). This suggests that adding an extra nurse for high-difficulty tasks did not substantially reduce failed tasks, and the practical difference between the two scenarios is minimal.

Failed Tasks by the High-performing nurse: Since both groups exhibited significant deviations from normality (Shapiro-Wilk p<0.001), the non-parametric Wilcoxon rank-sum test was applied. Results showed no statistically significant difference in the number of failed tasks between the Baseline and the Replacement scenario (W=1744.5, p=0.7642). The mean number of failed tasks was identical across both scenarios (Baseline: 1.92, Replacement: 1.92), indicating that replacing one nurse did not affect the failure rate of high-performing nurse.

Successful Tasks by the Low-performing nurse: Since both groups exhibited no variance (all values were identical at 0), neither the Shapiro-Wilk test for normality nor inferential statistical tests (Wilcoxon, Welch's t-test) could be meaningfully applied. Both the Baseline and Replacement scenarios resulted in a mean of 0 successful tasks per shift, indicating that the low-performing nurse was unable to complete any successful tasks under either condition.

Successful tasks by the High-performing nurse: Since both groups exhibited significant deviations from normality (Shapiro-Wilk $p<0.001$), the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference between the Baseline and Replacement scenarios ($W=1566.5$, $p=0.218$). The mean difference was approximately 1.25 successful tasks (Baseline: 16.67, Replacement: 17.92), indicating that the high-performing nurse achieved a comparable number of successful tasks in both conditions.

Nurse Utility by the Low-performing nurse: Since both groups exhibited significant deviations from normality (Shapiro-Wilk $p=0.006$ for Baseline, $p = 0.0009$ for Replacement), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario did not differ significantly in Nurse Utility of the low-performing nurse compared to the Baseline ($W= 1606$, $p=0.154$). The mean Nurse Utility was slightly lower for Replacement (-10.47) than for Baseline (-9.97), with a mean difference of approximately 0.5 units per nurse, indicating that replacing the low-performing nurse in Scenario 2 did not substantially change the overall nurse utility under Method 1.

Nurse Utility by the High-performing Nurse: Since both groups showed deviations from normality (Shapiro-Wilk $p<0.05$ for both Baseline and Replacement), the non-parametric Wilcoxon rank-sum test was applied. Results indicated that the Replacement scenario had significantly higher nurse utility ($W=2129.5$, $p=0.042$) compared to the Baseline. The mean difference was approximately 6.17 units of utility (Baseline: 50.28, Replacement: 56.45) suggesting that the Replacement scenario improves the efficiency of the high-performing nurse in Method 1.

Nurse Time Damage by the Low-performing Nurse: Since both groups exhibited significant deviations from normality (Shapiro-Wilk test, Baseline: $W=0.930$, $p=0.0021$, Replacement: $W=0.938$, $p=0.0045$), the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference between Baseline and Replacement ($W=1955.5$, $p=0.794$), indicating that the time damage caused by the low-performing nurse remained similar across scenarios. The mean NurseTimeDamage was slightly higher in the Replacement scenario (Baseline: 56.48, Replacement: 58.07), but this difference was not statistically significant. Thus, replacing the nurse team configuration did not reduce the time damage caused by the low-performing nurse.

Nurse Time Damage by the High-performing Nurse: Since both groups exhibited significant deviations from normality (Shapiro-Wilk $p<0.05$), the non-parametric Wilcoxon rank-sum test was applied. Results indicated no significant difference in Nurse Time Damage between the Baseline and Replacement scenarios ($W=1853$, $p=0.611$). The mean values were highly comparable (Baseline: 8.87, Replacement: 9.22), suggesting that introducing an additional nurse in the Replacement scenario does not substantially affect the workload-related time damage for the high-performing nurse.

Patient Distribution Across Doctors: In the Baseline scenario (Scenario 1), some nurses show a preference for certain doctors, while in the Replacement scenario (Scenario 2), the distribution is slightly more even. Using a Monte Carlo x^2 test for each simulation run, 18 out of 60 runs in the Baseline scenario showed a statistically significant nurse preference for specific doctors ($p<0.05$). In the Replacement scenario, 12 out of 60 runs showed a significant preference.

The variance in the number of patients served per doctor per run was 21.44 for the Baseline scenario and 29.71 for the Replacement scenario. A paired t-test comparing these variances found no statistically significant difference ($p=0.938$).

Conclusion: Nurses occasionally favor certain doctors in both scenarios, but the evidence is not strong enough to indicate a systematic preference. Patient distribution across doctors is generally uneven in both Baseline and Replacement scenarios, with no significant difference in variance between them.

Conclusions

The Replacement scenario, which adds an extra nurse to handle high-difficulty tasks, substantially increases overall patient throughput and reduces patient delay. However, these operational gains come with the implicit cost of employing additional staff. While total patient damage time rises slightly, the main trade-off to consider is the financial and logistical burden of the extra nurse. Decision-makers should weigh the benefits of higher efficiency and reduced patient delays against the increased staffing costs, particularly in resource-constrained settings.

F.3. Baseline (CA trust model) vs. Training (CA trust model)

Objective

Evaluate the impact of assigning trainer nurses to accompany low-performing nurses within the CA Trust model. The focus is on whether this intervention improves overall system throughput and efficiency, while accounting for potential trade-offs in error rates, patient safety, and nurse performance outcomes.

Design

- Scenarios: Baseline vs. Training;
- Task selection policy: CA trust model applied in both scenarios;
- Metrics analyzed: Total patients served, total patient damage time, total patient delay, failed and successful tasks per nurse, nurse utility, nurse time damage, patient distribution across doctors.

Statistical Approach

All metrics were tested for normality using the Shapiro-Wilk test. Given that most metrics significantly deviated from normality, the non-parametric Wilcoxon rank-sum test was applied. For metrics with no variance in one group, the test was applied against the non-normal counterpart. Effect sizes, means, and p-values are reported. For patient distribution, Monte Carlo chi-square tests and variance comparisons via paired t-tests were conducted.

Key Results

Total Patients Served: Since both groups exhibited significant deviations from normality (Shapiro-Wilk $p < 2.2 \times 10^{-16}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario served significantly more patients ($W = 10850, p < 2.2 \times 10^{-16}$) compared to the baseline. The mean difference was approximately 6.93 patients per shift (Baseline: 22.18, Training: 29.12), indicating that assigning trainer nurses to accompany low-performing nurses substantially increases overall patient throughput.

Total Patient Damage Time: Since both groups deviated significantly from normality (Shapiro-Wilk $p < 2.2 \times 10^{-16}$ for Baseline, $p = 6.73 \times 10^{-5}$for Training), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario had significantly higher total patient damage time ($W = 7025, p < 2.2 \times 10^{-16}$) compared to the Baseline. The mean difference was approximately 83.9 seconds per shift (Baseline: 65.38 sec, Training: 149.28 sec), indicating that assigning trainer nurses increased the overall patient damage time.

Total Patient Delay: Since both groups exhibited significant deviations from normality (Shapiro-Wilk $p < 2.2 \times 10^{-16}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario had significantly lower total patient delay (W=83100, $p < 2.2 \times 10^{-16}$) compared to the Baseline. The mean difference was approximately 380.9 seconds per shift (Baseline: 7095.77 s, Training: 6714.88 s), indicating that implementing the training scenario reduces patient delays and potentially improves overall efficiency in the emergency department.

Failed Tasks by the Low-performing nurse: Since failures in the Baseline group exhibited significant deviations from normality (Shapiro-Wilk W=0.914, p=0.00045), while the Training group did not (W=0.967, p=0.102), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the low-performing nurse committed significantly more failed tasks in the Training scenario ($W = 140, p < 2.2 \times 10^{-16}$) compared to the Baseline. The mean difference was approximately 4.28 failed tasks per shift (Baseline: 3.60, Training: 7.88). This outcome suggests that, although trainer nurses improve overall system throughput, low-performing nurses do not immediately reduce their error rate; instead, the improvement process requires gradual, long-term capability building.

Failed Tasks by the High-performing nurse: Since both Baseline and Training scenarios showed significant deviations from normality (Shapiro-Wilk p<0.001), the non-parametric Wilcoxon rank-sum test was applied. Results indicated no statistically significant difference in the number of failed tasks between the Baseline and Training scenarios (W=1675.5, p=0.5005). The mean number of failed tasks was similar across scenarios (Baseline: 1.92, Training: 2.08). These results suggest that the performance of the high-performing nurse remains stable between Baseline and Training, and the presence of trainers or the Training scenario itself does not significantly affect the number of failed tasks for this nurse.

Successful Tasks by the Low-performing nurse: Since the baseline group had identical values (all zeros), normality could not be assessed. The training group significantly deviated from normality (Shapiro-Wilk $W = 0.848, p = 2.72 \times 10^{-6}$), so the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario achieved significantly more successful tasks ($W = 480, p = 9.86 \times 10^{-16}$) compared to the baseline. The mean difference was approximately 2.43 tasks per shift

(Baseline: 0.00, Training: 2.43), indicating that training substantially improves the performance of low-performing nurses.

Successful Tasks by the High-performing nurse: Since both Baseline and Training scenarios exhibited significant deviations from normality (Shapiro-Wilk $p<0.001$ for both), the non-parametric Wilcoxon rank-sum test was applied. Results showed no statistically significant difference in the number of successful tasks between Baseline and Training ($W=1958.5$, $p=0.403$). The mean difference was minimal (Baseline: 16.67, Training: 16.72), indicating that training interventions to the High-performing nurse did not substantially affect task performance.

Nurse Utility by the Low-performing nurse: Since the Baseline group showed a significant deviation from normality (Shapiro-Wilk $p=0.006$) while the Training group did not ($p=0.941$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario had significantly lower Nurse Utility ($W = 551.5, p < 2.72 \times 10^{-11}$) compared to the Baseline. The mean difference was approximately -10.07 units (Baseline: -9.97, Training: -20.03), indicating that the Training scenario substantially reduces the utility of the low-performing nurse.

Nurse Utility by the High-performing nurse: Since both groups showed deviations from normality (Shapiro-Wilk $p<0.05$ for Baseline and Training), the non-parametric Wilcoxon rank-sum test was applied. Results indicated no significant difference in utility between the Training and Baseline scenarios ($W=1588$, $p=0.868$). The mean utility values were 50.28 for Baseline and 47.95 for Training. This suggests that assigning a trainer nurse did not substantially change the utility of the high-performing nurse in the simulation.

Nurse Time Damage by the Low-performing Nurse: Since the Baseline group exhibited significant deviation from normality (Shapiro-Wilk, $p=0.0021$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario did not reduce Nurse Time Damage ($W=3347.5$, $p=1$) compared to the Baseline. The mean time damage actually increased from 56.48 in Baseline to 140.33 s in Training. This outcome aligns with findings on failed tasks: although trainer nurses improve overall system throughput, low-performing nurses do not immediately reduce their error rate; instead, the improvement process requires gradual, long-term capability building. These

results highlight that training interventions may not immediately mitigate the inefficiencies caused by underperforming staff.

Nurse Time Damage by the High-performing Nurse: Since both groups deviated significantly from normality (Shapiro-Wilk $p<0.005$), the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference in Nurse Time Damage between the Training scenario and the Baseline ($W= 1811.5$, $p=0.5252$). The mean Nurse Time Damage was 8.87 s in the Baseline and 8.92 s in the Training scenario, indicating that the good nurse's performance remained stable. This aligns with our previous observation that while the trainer nurses can improve overall system throughput, individual low-performing nurses require gradual capability building; high-performing nurses maintain consistent efficiency regardless of scenario.

Patient Distribution Across Doctors: In the Baseline scenario, some nurses show a preference for certain doctors, while in the Training scenario, the distribution is slightly more even. Using a Monte Carlo chi-square test for each simulation run, 18 out of 60 runs in the Baseline scenario showed a statistically significant nurse preference for specific doctors ($p<0.05$). In the Training scenario, 15 out of 60 runs showed a significant preference.

The variance in the number of patients served per doctor per run was 21.44 for the Baseline scenario and 26.29 for the Training scenario. A paired t-test comparing these variances found no statistically significant difference ($p=0.838$).

Conclusion: Nurses occasionally favor certain doctors in both scenarios, but the evidence is not strong enough to indicate a systematic preference. Patient distribution across doctors is generally uneven in both Baseline and Training scenarios, with no significant difference in variance between them.

Conclusions

The Training scenario substantially increases the number of patients served and reduces total patient delays demonstrating a clear improvement in system throughput under the use of CA Trust model. However, this comes at the cost of higher patient damage time and increased failed tasks by the low-performing nurse, highlighting that the error rate does not improve immediately. Instead, capability building for underperforming staff appears to be gradual and long-term in nature. The presence of trainer nurses

stabilizes outcomes for high-performing nurses, whose performance and utility remain largely unaffected across scenarios.

From a managerial perspective, the Training scenario illustrates a critical trade-off: while system-level efficiency improves, short-term patient safety risks may increase due to persistent errors by low-performing nurses. Decision-makers should recognize that training interventions deliver operational gains but require sustained investment in long-term skill development to mitigate the risks associated with low-performing staff.

F.4. Replacement (CA trust model) vs. Training (CA trust model)

Objective

Compare two intervention strategies for handling low-performing nurses: immediate replacement vs skill development under training. Specifically, the study investigates how these interventions affect patient throughput, patient safety, nurse performance, and overall system efficiency.

Design

- Scenarios: Baseline vs. Training;
- Task selection policy: CA trust model applied in both scenarios;
- Metrics analyzed: Total patients served, total patient damage time, total patient delay, failed and successful tasks per nurse, nurse utility, nurse time damage, patient distribution across doctors.

Statistical Approach

All metrics were tested for normality using the Shapiro-Wilk test. Given that most metrics significantly deviated from normality, the non-parametric Wilcoxon rank-sum test was applied. For metrics with no variance in one group, the test was applied against the non-normal counterpart. Effect sizes, means, and p-values are reported. For patient distribution, Monte Carlo chi-square tests and variance comparisons via paired t-tests were conducted.

Key Results

Total Patients Served: Since both groups deviated significantly from normality (Shapiro-Wilk test, $p < 2.2 \times 10^{-16}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario served significantly more patients (W=104114, $p < 2.2 \times 10^{-16}$) compared to the Training scenario. The mean difference

was approximately 12.47 patients per shift (Replacement: 41.58, Training: 29.12), indicating that immediate replacement of low-performing nurses results in higher patient throughput compared to training.

Total Patient Damage Time: Since both groups exhibited significant deviations from normality (Shapiro-Wilk test, Replacement: $= 0.982, p = 0.00016$; $Training: W = 0.976, p = 6.73 \times 10^{-5}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario had significantly lower total patient damage time compared to the Training scenario ($W = 19695, p < 2.2 \times 10^{-16}$). The mean difference was approximately 53.3 seconds per shift (Replacement: 95.97 s, Training: 149.28 s), indicating that replacing low-performing nurses with more capable ones substantially reduces the overall damage exposure time of patients in the emergency department.

Total Patient Delay: Since both groups deviated significantly from normality (Shapiro-Wilk test, $p < 2.2 \times 10^{-16}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario had significantly lower patient delay (W=4440, $p < 2.2 \times 10^{-16}$) compared to the Training scenario. The mean difference was approximately 430.77 seconds per shift (Replacement: 6284.12, Training: 6714.88), indicating that immediate replacement of low-performing nurses reduces total patient delay more effectively than training.

Failed Tasks by the Low-performing nurse: Since the Replacement group deviated significantly from normality (Shapiro-Wilk test, $W = 0.908, p = 2.66 \times 10^{-4}$), while the Training group was approximately normal (W=0.967, p=0.102), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the low-performing nurse had significantly fewer failed tasks in the Replacement scenario compared to the Training scenario (W=156.5, $p < 2.2 \times 10^{-16}$). The mean difference was approximately 3.88 failed tasks per shift (Replacement: 4.00, Training: 7.88), indicating that immediate replacement of low-performing nurses reduces task failures compared to training.

Failed Tasks by the High-performing nurse: Since both groups deviated significantly from normality (Shapiro-Wilk test, Replacement: W=0.910, p=0.0003; Training: W=0.860, $p = 6.3 \times 10^{-6}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed no statistically significant difference between Replacement and

Training (W=1727.5, p=0.695). The mean number of failed tasks was similar across conditions (Replacement: 1.92, Training: 2.08).

Successful Tasks by the Low-performing nurse: Since the Replacement group consisted of identical values (all zero) and the Training group deviated significantly from normality (Shapiro-Wilk test, $W = 0.848, p = 2.72 \times 10^{-6}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Training scenario produced significantly more successful tasks ($W = 480, p < 9.86 \times 10^{-16}$) compared to the Replacement scenario. The mean difference was approximately 2.43 successful tasks per shift (Replacement: 0.00, Training: 2.43), indicating that training substantially improved the performance of low-performing nurses compared to immediate replacement.

Successful Tasks by the High-performing nurse: Since both groups deviated significantly from normality (Shapiro-Wilk test, Replacement: $W = 0.813, p = 2.97 \times 10^{-7}$; Training: $W = 0.747, p = 7.98 \times 10^{-9}$), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario yielded significantly more successful tasks compared to the Training scenario (W=2269, p=0.013). The mean difference was approximately 1.20 tasks per shift (Replacement: 17.92, Training: 16.72). The results suggest that immediate replacement of the low-performing nurse leads to a modest but statistically significant improvement in the performance of the high-performing nurse, as measured by the number of successful tasks.

Nurse Utility by the Low-performing nurse: Since the Replacement group deviated significantly from normality (Shapiro-Wilk test, W=0.922, p=0.0009), while the Training group did not (W=0.991, p=0.94), the non-parametric Wilcoxon rank-sum test was applied. Results showed that the Replacement scenario yielded significantly higher utility for the low-performing nurse compared to the Training scenario ($W = 3038, p = 7.85 \times 10^{-11}$). The mean difference was approximately 9.57 utility points (Replacement: -10.47, Training: -20.03).

Nurse Utility by the High-performing nurse: Since both groups significantly deviated from normality (Shapiro-Wilk test, Replacement: W=0.951, p=0.018; Training: W=0.958, p=0.036), the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant improvement of nurse utility in the Training scenario compared to Replacement (W=1189, p=0.999), with mean utility being higher under Replacement (56.45) than Training (47.95).

Nurse Time Damage by the Low-performing nurse: Since the Replacement group significantly deviated from normality (Shapiro-Wilk test, Replacement: W= 0.938, p=0.0045) while the Training group was approximately normal (W=0.974, p=0.230), the non-parametric Wilcoxon rank-sum test was applied. Results showed a significant increase in nurse time damage in the Training scenario compared to Replacement ($W = 3384, p < 2.2 \times 10^{-16}$), with mean time damage being substantially higher under Training (140.33) than Replacement (58.07).

Nurse Time Damage by the High-performing nurse: Since both groups significantly deviated from normality (Shapiro-Wilk test, Replacement: W=0.946, p=0.011; Training: W=0.937, p=0.004), the non-parametric Wilcoxon rank-sum test was applied. Results showed no significant difference in nurse time damage between the Training and Replacement scenarios (W=1750.5, p=0.604), with mean time damage being slightly lower under Training (8.92) than Replacement (9.22).

Patient Distribution Across Doctors: In the Replacement scenario, some nurses show a preference for certain doctors, while in the Training scenario, the distribution is slightly more even. Using a Monte Carlo chi-square test for each simulation run, 12 out of 60 runs in the Replacement scenario showed a statistically significant nurse preference for specific doctors ($p<0.05$). In the Training scenario, 15 out of 60 runs showed a significant preference.

The variance in the number of patients served per doctor per run was 29.71 for the Replacement scenario and 26.29 for the Training scenario. A paired t-test comparing these variances found no statistically significant difference (p=0.274).

Nurses occasionally favor certain doctors in both scenarios, but the evidence is not strong enough to indicate a systematic preference. Patient distribution across doctors is generally uneven in both Replacement and Training scenarios, with no significant difference in variance between them.

Conclusions

Replacement of low-performing nurses substantially improves patient throughput, reduces patient damage time, and lowers total patient delay compared to training. Training, however, enables low-performing nurses to achieve more successful tasks. The performance of high-performing nurses improves slightly under Replacement, while

overall nurse utility favors Replacement. Patient distribution across doctors remains uneven but comparable across scenarios.

The Replacement scenario provides clear operational benefits, including faster patient service and reduced cumulative damage and delay. Training improves the skills of low-performing nurses, increasing their successful tasks but at the cost of higher patient delay and damage. Decision-makers must weigh the trade-off between immediate efficiency gains from Replacement versus long-term skill development through Training, especially when staffing resources are limited.

References

1. Bao, F.; Chen, I.-R. Dynamic Trust Management for Internet of Things Applications. In Proceedings of the Proceedings of the 2012 international workshop on Self-aware internet of things; ACM: San Jose California USA, September 17 2012; pp. 1–6.
2. Lee, G.M.; Truong, N.B. A Reputation and Knowledge Based Trust Service Platform for Trustworthy Social Internet of Things. In Proceedings of the Innovations in Clouds, Internet and Networks (ICIN) 2016; 2016.
3. Darroux, A.; Jamont, J.-P.; Aktouf, O.-E.-K.; Mercier, A. An Energy Aware Approach to Trust Management Systems for Embedded Multi-Agent Systems. In Proceedings of the Software Engineering for Resilient Systems; Calinescu, R., Di Giandomenico, F., Eds.; Springer International Publishing: Cham, 2019; pp. 121–137.
4. Das, P.; Debnath, S. A Trust Computing Model for Future Generation Networks. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT); IEEE, 2020; pp. 1–4.
5. Alhandi, S.A.; Kamaludin, H.; Alduais, N.A.M. Trust Evaluation Model in IoT Environment: A Comprehensive Survey. *IEEE Access* **2023**, *11*, 11165–11182, doi:10.1109/ACCESS.2023.3240990.
6. Lu, Z.; Delaney, D.T.; Lillis, D. A Survey on Microservices Trust Models for Open Systems. *IEEE Access* **2023**, *11*, 28840–28855, doi:10.1109/ACCESS.2023.3260147.
7. Alghofaili, Y.; Rassam, M.A. A Trust Management Model for IoT Devices and Services Based on the Multi-Criteria Decision-Making Approach and Deep Long Short-Term Memory Technique. *Sensors* **2022**, *22*, 634, doi:10.3390/s22020634.
8. Huynh, T.D.; Jennings, N.R.; Shadbolt, N.R. An Integrated Trust and Reputation Model for Open Multi-Agent Systems. *Auton Agent Multi-Agent Syst* **2006**, *13*, 119–154, doi:10.1007/s10458-005-6825-4.
9. Hermoso, R.; Billhardt, H.; Ossowski, S. Trust-Based Role Coordination in Task-Oriented Multiagent Systems. *Knowledge-Based Systems* **2013**, *52*, 78–90, doi:10.1016/j.knosys.2013.07.004.
10. Jin, D.; Kannengießer, N.; Sturm, B.; Sunyaev, A. Tackling Challenges of Robustness Measures for Agent Collaboration in Open Multi-Agent Systems. *Hawaii International Conference on System Sciences 2022 (HICSS-55)* **2022**.
11. Hattab, S.; Chaari, W.L. A Generic Model for Representing Openness in Multi-Agent Systems. *The Knowledge Engineering Review* **2021**, *36*, e3, doi:10.1017/S0269888920000429.
12. Jumadinova, J.; Dasgupta, P.; Soh, L.-K. Strategic Capability-Learning for Improved Multiagent Collaboration in Ad Hoc Environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2014**, *44*, 1003–1014, doi:10.1109/TSMC.2013.2285527.
13. Chen, B.; Chen, X.; Timsina, A.; Soh, L.-K. Considering Agent and Task Openness in Ad Hoc Team Formation. In Proceedings of the AAMAS; 2015; pp. 1861–1862.
14. Weyns, D. *Architecture-Based Design of Multi-Agent Systems*; Springer Science & Business Media, 2010; ISBN 978-3-642-01064-4.

15. Eck, A.; Soh, L.-K.; Doshi, P. Decision Making in Open Agent Systems. *AI Magazine* **2023**, *44*, 508–523, doi:10.1002/aaai.12131.
16. dos Santos, F.; Bazzan, A.L.C. Towards Efficient Multiagent Task Allocation in the RoboCup Rescue: A Biologically-Inspired Approach. *Auton Agent Multi-Agent Syst* **2011**, *22*, 465–486, doi:10.1007/s10458-010-9136-3.
17. Nunoo-Mensah, H.; Boateng, K.O.; Gadze, J.D. The Adoption of Socio- and Bio-Inspired Algorithms for Trust Models in Wireless Sensor Networks: A Survey. *International Journal of Communication Systems* **2018**, *31*, e3444, doi:10.1002/dac.3444.
18. Schwarzrock, J.; Zacarias, I.; Bazzan, A.L.C.; de Araujo Fernandes, R.Q.; Moreira, L.H.; de Freitas, E.P. Solving Task Allocation Problem in Multi Unmanned Aerial Vehicles Systems Using Swarm Intelligence. *Engineering Applications of Artificial Intelligence* **2018**, *72*, 10–20, doi:10.1016/j.engappai.2018.03.008.
19. Brutschy, A.; Pini, G.; Pinciroli, C.; Birattari, M.; Dorigo, M. Self-Organized Task Allocation to Sequentially Interdependent Tasks in Swarm Robotics. *Auton Agent Multi-Agent Syst* **2014**, *28*, 101–125, doi:10.1007/s10458-012-9212-y.
20. Ferreira, P.R.; Boffo, F.S.; Bazzan, A.L.C. A Swarm Based Approximated Algorithm to the Extended Generalized Assignment Problem (E-GAP). In Proceedings of the Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems; ACM: Honolulu Hawaii, May 14 2007; pp. 1–3.
21. Zambonelli, F.; Omicini, A.; Anzengruber, B.; Castelli, G.; De Angelis, F.L.; Serugendo, G.D.M.; Dobson, S.; Fernandez-Marquez, J.L.; Ferscha, A.; Mamei, M.; et al. Developing Pervasive Multi-Agent Systems with Nature-Inspired Coordination. *Pervasive and Mobile Computing* **2015**, *17*, 236–252, doi:10.1016/j.pmcj.2014.12.002.
22. Skaltsis, G.M.; Shin, H.-S.; Tsourdos, A. A Survey of Task Allocation Techniques in MAS. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS); June 2021; pp. 488–497.
23. Jevtic, A.; Gutierrez, Á.; Andina, D.; Jamshidi, M. Distributed Bees Algorithm for Task Allocation in Swarm of Robots. *IEEE Systems Journal* **2012**, *6*, 296–304, doi:10.1109/JSYST.2011.2167820.
24. Tkach, I.; Edan, Y.; Jevtic, A.; Nof, S.Y. Automatic Multi-Sensor Task Allocation Using Modified Distributed Bees Algorithm. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics; July 2013; pp. 1401–1406.
25. Tkach, I.; Jevtić, A.; Nof, S.Y.; Edan, Y. A Modified Distributed Bees Algorithm for Multi-Sensor Task Allocation. *Sensors* **2018**, *18*, 759, doi:10.3390/s18030759.
26. Kanth, V.; Mceachen, J.; Tummala, M. A Bio-Inspired Trust Framework in Wireless Ad Hoc Networks. *Hawaii International Conference on System Sciences 2021 (HICSS-54)* **2021**.
27. Arya, V.; Rani, S.; Choudhary, N. Enhanced Bio-Inspired Trust and Reputation Model for Wireless Sensor Networks. In Proceedings of the Proceedings of Second Doctoral Symposium on Computational Intelligence; Gupta, D., Khanna, A., Kansal, V., Fortino, G., Hassanien, A.E., Eds.; Springer: Singapore, 2022; pp. 569–579.

28. Adam, M.; Hammoudeh, M.; Alrawashdeh, R.; Alsulaimy, B. A Survey on Security, Privacy, Trust, and Architectural Challenges in IoT Systems. *IEEE Access* **2024**, *12*, 57128–57149, doi:10.1109/ACCESS.2024.3382709.
29. Nkosi, M.T.; Adigun, M.O.; Emuoyibofarhe, J.O. Agent-to-Agent Reputation-Based Trust Management. In Proceedings of the IADIS International Conference Applied Computing; 2007; pp. 143–150.
30. Yan, Z.; Holtmanns, S. Trust Modeling and Management: From Social Trust to Digital Trust. In *Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions*; IGI Global Scientific Publishing, 2008; pp. 290–323 ISBN 978-1-59904-804-8.
31. Bidgoly, A.J. Robustness Verification of Soft Security Systems. *Journal of Information Security and Applications* **2020**, *55*, 102632, doi:10.1016/j.jisa.2020.102632.
32. Parhizkar, E.; Nikravan, M.H.; Holte, R.C.; Zilles, S. Combining Direct Trust and Indirect Trust in Multi-Agent Systems. In Proceedings of the Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence; International Joint Conferences on Artificial Intelligence Organization: Yokohama, Japan, July 2020; pp. 311–317.
33. Premalatha, B.; Prakasam, P. TwI-FTM: Two-Way IoT-FoG Trust Management Scheme for Task Offloading in IoT-FoG Networks. *Results in Engineering* **2024**, *22*, 102197.
34. Scarlat, E.; Maries, I. Towards an Increase of Collective Intelligence within Organizations Using Trust and Reputation Models. In Proceedings of the Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems; Nguyen, N.T., Kowalczyk, R., Chen, S.-M., Eds.; Springer: Berlin, Heidelberg, 2009; pp. 140–151.
35. Fortino, G.; Fotia, L.; Messina, F.; Rosaci, D.; Sarné, G.M.L. Trust and Reputation in the Internet of Things: State-of-the-Art and Research Challenges. *IEEE Access* **2020**, *8*, 60117–60125, doi:10.1109/ACCESS.2020.2982318.
36. Lim, J.W.Z.; Thing, V.L.L. Building Confidence in a World of Eroding Trust. *Digit. Gov.: Res. Pract.* **2024**, *5*, 16:1-16:17, doi:10.1145/3653075.
37. Truong, N.B.; Lee, H.; Askwith, B.; Lee, G.M. Toward a Trust Evaluation Mechanism in the Social Internet of Things. *Sensors* **2017**, *17*, 1346.
38. Saputra, D.E. Defining Trust in Computation. In Proceedings of the 2020 International Conference on Information Technology Systems and Innovation (ICITSI); July 2020; pp. 161–166.
39. Braga, D.D.S.; Niemann, M.; Hellingrath, B.; Neto, F.B.D.L. Survey on Computational Trust and Reputation Models. *ACM Comput. Surv.* **2018**, *51*, 101:1-101:40, doi:10.1145/3236008.
40. Huang, X.; Kwiatkowska, M.; Olejnik, M. Reasoning about Cognitive Trust in Stochastic Multiagent Systems. *ACM Trans. Comput. Logic* **2019**, *20*, 21:1-21:64, doi:10.1145/3329123.
41. Urbano, J.; Rocha, A.P.; Oliveira, E. The Impact of Benevolence in Computational Trust. In *Agreement Technologies*; Chesñevar, C.I., Onaindia, E., Ossowski, S., Vouros, G., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013; Vol. 8068, pp. 210–224 ISBN 978-3-642-39859-9.
42. Urbano, J.; Rocha, A.P.; Oliveira, E. Computing Confidence Values: Does Trust Dynamics Matter? In Proceedings of the Progress in Artificial Intelligence; Lopes,

L.S., Lau, N., Mariano, P., Rocha, L.M., Eds.; Springer: Berlin, Heidelberg, 2009; pp. 520–531.
43. Ferraris, D.; Fernandez-Gago, C.; Roman, R.; Lopez, J. A Survey on IoT Trust Model Frameworks. *J Supercomput* **2024**, *80*, 8259–8296, doi:10.1007/s11227-023-05765-4.
44. Azevedo-Sa, H.; Yang, X.J.; Robert, L.P.; Tilbury, D.M. A Unified Bi-Directional Model for Natural and Artificial Trust in Human–Robot Collaboration. *IEEE Robotics and Automation Letters* **2021**, *6*, 5913–5920, doi:10.1109/LRA.2021.3088082.
45. Amaral, G.; Sales, T.P.; Guizzardi, G.; Porello, D. Towards a Reference Ontology of Trust. In Proceedings of the On the Move to Meaningful Internet Systems: OTM 2019 Conferences; Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R., Eds.; Springer International Publishing: Cham, 2019; pp. 3–21.
46. Sabater, J.; Sierra, C. REGRET: Reputation in Gregarious Societies. In Proceedings of the Proceedings of the fifth international conference on Autonomous agents; ACM: Montreal Quebec Canada, May 28 2001; pp. 194–195.
47. Carter, J.; Ghorbani, A.A. Towards a Formalization of Value-Centric Trust in Agent Societies. *Web Intelligence and Agent Systems* **2004**, *2*, 167–183, doi:10.3233/WEB-2004-wia00039.
48. Deljoo, A.; van Engers, T.; Gommans, L.; de Laat, C. The Impact of Competence and Benevolence in a Computational Model of Trust. In Proceedings of the Trust Management XII; Gal-Oz, N., Lewis, P.R., Eds.; Springer International Publishing: Cham, 2018; pp. 45–57.
49. Patel, R.; Acharya, S. A Fuzzy Logic Based Trust Evaluation Model for IoT. In Proceedings of the Intelligent Systems; Udgata, S.K., Sethi, S., Gao, X.-Z., Eds.; Springer Nature: Singapore, 2024; pp. 147–156.
50. Griffiths, N. Task Delegation Using Experience-Based Multi-Dimensional Trust. In Proceedings of the Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems; Association for Computing Machinery: New York, NY, USA, April 25 2005; pp. 489–496.
51. Maximilien, E.M.; Singh, M.P. Agent-Based Trust Model Involving Multiple Qualities. In Proceedings of the Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems; Association for Computing Machinery: New York, NY, USA, April 25 2005; pp. 519–526.
52. Messina, F.; Rosaci, D.; Sarnè, G.M.L. Applying Trust Patterns to Model Complex Trustworthiness in the Internet of Things. *Electronics* **2024**, *13*, 2107, doi:10.3390/electronics13112107.
53. Truong, N.B.; Um, T.-W.; Zhou, B.; Lee, G.M. From Personal Experience to Global Reputation for Trust Evaluation in the Social Internet of Things. In Proceedings of the GLOBECOM 2017 - 2017 IEEE Global Communications Conference; September 2017; pp. 1–7.
54. Battah, A.; Iraqi, Y.; Damiani, E. Blockchain-Based Reputation Systems: Implementation Challenges and Mitigation. *Electronics* **2021**, *10*, 289, doi:10.3390/electronics10030289.
55. Beltzer, B.; Conchon, E.; Giroux, S. New Aggregators for Global Reputation on Bi-Lattice Based Trust Model. *IEEE Access* **2024**, *12*, 15713–15725, doi:10.1109/ACCESS.2024.3357079.

56. Sagar, S.; Mahmood, A.; Sheng, Q.Z.; Zaib, M.; Sufyan, F. Can We Quantify Trust? Towards a Trust-Based Resilient SIoT Network. *Computing* **2024**, *106*, 557–577, doi:10.1007/s00607-023-01236-z.
57. Arazzi, M.; Nicolazzo, S.; Nocera, A. A Novel IoT Trust Model Leveraging Fully Distributed Behavioral Fingerprinting and Secure Delegation. *Pervasive and Mobile Computing* **2024**, *99*, 101889, doi:10.1016/j.pmcj.2024.101889.
58. Ogunbunmi, S.; Chen, Y.; Blasch, E.; Chen, G. A Survey on Reputation Systems for UAV Networks. *Drones* **2024**, *8*, 253, doi:10.3390/drones8060253.
59. Keshavarz, M.; Gharib, M.; Afghah, F.; Ashdown, J.D. UASTrustChain: A Decentralized Blockchain- Based Trust Monitoring Framework for Autonomous Unmanned Aerial Systems. *IEEE Access* **2020**, *8*, 226074–226088, doi:10.1109/ACCESS.2020.3044844.
60. Xiao, Y.; Zhu, L.; Li, X. A Review on Trust and Reputation Management Systems in E-Commerce and P2P Network. In Proceedings of the 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT); March 2021; pp. 58–62.
61. Mohsenzadeh, A.; Nickray, M.; Bidgoly, A.J.; Farjami, Y. A New Task Scheduling Mechanism in Grid Computing Using Trust Model. *Multiagent and Grid Systems* **2020**, *16*, 135–152, doi:10.3233/MGS-200325.
62. Tian, H.; Tian, J. A Blockchain-Based Access Control Scheme for Reputation Value Attributes of the Internet of Things. *Computers, Materials, & Continua* **2024**, *78*, 1297.
63. Pinyol, I.; Sabater-Mir, J. Computational Trust and Reputation Models for Open Multi-Agent Systems: A Review. *Artif Intell Rev* **2013**, *40*, 1–25, doi:10.1007/s10462-011-9277-z.
64. Tyagi, H.; Kumar, R.; Pandey, S.K. A Detailed Study on Trust Management Techniques for Security and Privacy in IoT: Challenges, Trends, and Research Directions. *High-Confidence Computing* **2023**, *3*, 100127, doi:10.1016/j.hcc.2023.100127.
65. Jorquera Valero, J.M.; Sánchez Sánchez, P.M.; Gil Pérez, M.; Huertas Celdrán, A.; Martinez Perez, G. Cutting-Edge Assets for Trust in 5G and Beyond: Requirements, State of the Art, Trends, and Challenges. *ACM Comput. Surv.* **2023**, *55*, 1–36, doi:10.1145/3572717.
66. Sellami, C.; Baron, M.; Bechchi, M.; Hadjali, A.; Jean, S.; Chabot, D. Towards a Unified Framework for Computational Trust and Reputation Models for E-Commerce Applications. In Proceedings of the Research Challenges in Information Science; Cherfi, S., Perini, A., Nurcan, S., Eds.; Springer International Publishing: Cham, 2021; pp. 616–622.
67. Moyano, F.; Fernandez-Gago, C.; Lopez, J. A Conceptual Framework for Trust Models. In *Trust, Privacy and Security in Digital Business*; Fischer-Hübner, S., Katsikas, S., Quirchmayr, G., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; Vol. 7449, pp. 93–104 ISBN 978-3-642-32286-0.
68. Fernandez-Gago, C.; Ferraris, D.; Roman, R.; Lopez, J. Trust Interoperability in the Internet of Things. *Internet of Things* **2024**, *26*, 101226, doi:10.1016/j.iot.2024.101226.
69. Yang, Q.; Parasuraman, R. How Can Robots Trust Each Other For Better Cooperation? A Relative Needs Entropy Based Robot-Robot Trust Assessment

Model. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC); July 2021; pp. 2656–2663.
70. Cofta, P. Trust, Choice, and Self-Preservation: A Computational Approach. *Cogn Tech Work* **2021**, *23*, 85–101, doi:10.1007/s10111-019-00600-4.
71. Cheng, M.; Yin, C.; Zhang, J.; Nazarian, S.; Deshmukh, J.; Bogdan, P. A General Trust Framework for Multi-Agent Systems. In Proceedings of the Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems; 2021; pp. 332–340.
72. Aref, A.; Tran, T.T. A Trust Establishment Model in Multi-Agent Systems. In Proceedings of the AAAI Workshop: Incentive and Trust in E-communities; 2015.
73. Castelfranchi, C.; Falcone, R.; Marzo, F. Being Trusted in a Social Network: Trust as Relational Capital. In Proceedings of the Trust Management; Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F., Eds.; Springer: Berlin, Heidelberg, 2006; pp. 19–32.
74. Sen, S. A Comprehensive Approach to Trust Management. In Proceedings of the Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems; 2013; pp. 797–800.
75. Griffiths, N. Using Recency and Relevance to Assess Trust and Reputation. *Proceedings of AISB 2008 Symposium on Behaviour Regulation* **2008**.
76. Zytniewski, M.; Klement, M. Trust in Software Agent Societies. *Online Journal of Applied Knowledge Management* **2015**, *3*.
77. Siegert, V. Content- and Context-Related Trust in Open Multi-Agent Systems Using Linked Data. In Proceedings of the Web Engineering; Bakaev, M., Frasincar, F., Ko, I.-Y., Eds.; Springer International Publishing: Cham, 2019; pp. 541–547.
78. Burnett, C.; Norman, T.J.; Sycara, K. Trust Decision-Making in Multi-Agent Systems. In Proceedings of the Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume One; AAAI Press: Barcelona, Catalonia, Spain, April 16 2011; pp. 115–120.
79. Kramer, R.M.; Tyler, T.R. *Trust in Organizations: Frontiers of Theory and Research*; SAGE, 1996; ISBN 978-0-8039-5740-4.
80. Nguyen, M.H.; Tran, D.Q. A Trust Model for New Member in Multiagent System. *Vietnam J Comput Sci* **2015**, *2*, 181–190, doi:10.1007/s40595-015-0043-z.
81. Wang, J.; Jing, X.; Yan, Z.; Fu, Y.; Pedrycz, W.; Yang, L.T. A Survey on Trust Evaluation Based on Machine Learning. *ACM Comput. Surv.* **2020**, *53*, 107:1-107:36, doi:10.1145/3408292.
82. Jelenc, D. Toward Unified Trust and Reputation Messaging in Ubiquitous Systems. *Ann. Telecommun.* **2021**, *76*, 119–130, doi:10.1007/s12243-020-00792-1.
83. Player, C.; Griffiths, N. Improving Trust and Reputation Assessment with Dynamic Behaviour. *The Knowledge Engineering Review* **2020**, *35*, e29, doi:10.1017/S0269888920000077.
84. Jabeen, F.; Khan, Z.-R.; Hamid, Z.; Rehman, Z.; Khan, A. Adaptive and Survivable Trust Management for Internet of Things Systems. *IET Information Security* **2021**, *15*, 375–394, doi:10.1049/ise2.12029.
85. Kolomvatsos, K.; Kalouda, M.; Papadopoulou, P.; Hadjiefthymiades, S. Fuzzy Trust Modelling for Pervasive Computing Applications. *J. Data Intell.* **2021**, *2*, 101–115.

86. li, B.; Xing, M.; Zhu, J.; Che, T. A Dynamic Trust Model for the Multi-Agent Systems. In Proceedings of the 2008 International Symposiums on Information Processing; February 2008; pp. 500–504.
87. Wang, E.K.; Chen, C.-M.; Zhao, D.; Ip, W.H.; Yung, K.L. A Dynamic Trust Model in Internet of Things. *Soft Comput* **2020**, *24*, 5773–5782, doi:10.1007/s00500-019-04319-2.
88. Ahmad, I.; Yau, K.-L.A.; Loong Keoh, S. A Hybrid Reinforcement Learning-Based Trust Model for 5G Networks. In Proceedings of the 2020 IEEE Conference on Application, Information and Network Security (AINS); August 2020; pp. 20–25.
89. Kumari, S.; Kumar, S.; R, V. Trust Management in Social Internet of Things: Challenges and Future Directions. *IJCDS* **2023**, *14*, 899–920, doi:10.12785/ijcds/140170.
90. Sato, K.; Sugawara, T. Multi-Agent Task Allocation Based on Reciprocal Trust in Distributed Environments. In Proceedings of the Agents and Multi-Agent Systems: Technologies and Applications 2021; Jezic, G., Chen-Burger, J., Kusek, M., Sperka, R., Howlett, R.J., Jain, L.C., Eds.; Springer: Singapore, 2021; pp. 477–488.
91. Wang, J.; Yan, Z.; Wang, H.; Li, T.; Pedrycz, W. A Survey on Trust Models in Heterogeneous Networks. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 2127–2162, doi:10.1109/COMST.2022.3192978.
92. Qureshi, B.; Min, G.; Kouvatsos, D. Collusion Detection and Prevention with FIRE+ Trust and Reputation Model. In Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology; June 2010; pp. 2548–2555.
93. Khosravifar, B.; Bentahar, J.; Gomrokchi, M.; Alam, R. CRM: An Efficient Trust and Reputation Model for Agent Computing. *Knowledge-Based Systems* **2012**, *30*, 1–16, doi:10.1016/j.knosys.2011.01.004.
94. Khalid, R.; Samuel, O.; Javaid, N.; Aldegheishem, A.; Shafiq, M.; Alrajeh, N. A Secure Trust Method for Multi-Agent System in Smart Grids Using Blockchain. *IEEE Access* **2021**, *9*, 59848–59859, doi:10.1109/ACCESS.2021.3071431.
95. Oh, H.-K.; Jung, J.; Park, S.; Kim, S.-W. A Robust Reputation System Using Online Reviews. *COMPUTER SCIENCE AND INFORMATION SYSTEMS* **2020**, *17*, doi:10.2298/CSIS191122007O.
96. Bangui, H.; Ge, M.; Buhnova, B. Deep-Learning Based Reputation Model for Indirect Trust Management. *Procedia Computer Science* **2023**, *220*, 405–412, doi:10.1016/j.procs.2023.03.052.
97. Jurca, R.; Faltings, B. Towards Incentive-Compatible Reputation Management. In *Trust, Reputation, and Security: Theories and Practice*; Falcone, R., Barber, S., Korba, L., Singh, M., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg, 2003; Vol. 2631, pp. 138–147 ISBN 978-3-540-00988-7.
98. Samuel, O.; Javaid, N.; Khalid, A.; Imrarn, M.; Nasser, N. A Trust Management System for Multi-Agent System in Smart Grids Using Blockchain Technology. In Proceedings of the GLOBECOM 2020 - 2020 IEEE Global Communications Conference; September 2020; pp. 1–6.
99. Alam, S.; Zardari, S.; Shamsi, J.A. Blockchain-Based Trust and Reputation Management in SIoT. *Electronics* **2022**, *11*, 3871, doi:10.3390/electronics11233871.

100. Fragkos, G.; Minwalla, C.; Plusquellic, J.; Tsiropoulou, E.E. Local Trust in Internet of Things Based on Contract Theory. *Sensors* **2022**, *22*, 2393, doi:10.3390/s22062393.
101. Pan, Q.; Wu, J.; Li, J.; Yang, W.; Guan, Z. Blockchain and AI Empowered Trust-Information-Centric Network for Beyond 5G. *IEEE Network* **2020**, *34*, 38–45, doi:10.1109/MNET.021.1900608.
102. Muhammad, S.; Umar, M.M.; Khan, S.; Alrajeh, N.A.; Mohammed, E.A. Honesty-Based Social Technique to Enhance Cooperation in Social Internet of Things. *Applied Sciences* **2023**, *13*, 2778, doi:10.3390/app13052778.
103. Ali, S.E.; Tariq, N.; Khan, F.A.; Ashraf, M.; Abdul, W.; Saleem, K. BFT-IoMT: A Blockchain-Based Trust Mechanism to Mitigate Sybil Attack Using Fuzzy Logic in the Internet of Medical Things. *Sensors* **2023**, *23*, 4265, doi:10.3390/s23094265.
104. Kouicem, D.E.; Imine, Y.; Bouabdallah, A.; Lakhlef, H. Decentralized Blockchain-Based Trust Management Protocol for the Internet of Things. *IEEE Transactions on Dependable and Secure Computing* **2022**, *19*, 1292–1306, doi:10.1109/TDSC.2020.3003232.
105. Ouechtati, H.; Nadia, B.A.; Lamjed, B.S. A Fuzzy Logic-Based Model for Filtering Dishonest Recommendations in the Social Internet of Things. *J Ambient Intell Human Comput* **2023**, *14*, 6181–6200, doi:10.1007/s12652-021-03127-7.
106. Mianji, E.M.; Muntean, G.-M.; Tal, I. Trust and Reputation Management for Data Trading in Vehicular Edge Computing: A DRL-Based Approach. In Proceedings of the 2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB); June 2024; pp. 1–7.
107. Liu, Y.; Liu, Z.; Zhang, Q.; Su, J.; Cai, Z.; Li, X. Blockchain and Trusted Reputation Assessment-Based Incentive Mechanism for Healthcare Services. *Future Generation Computer Systems* **2024**, *154*, 59–71, doi:10.1016/j.future.2023.12.023.
108. Marche, C.; Nitti, M. Trust-Related Attacks and Their Detection: A Trust Management Model for the Social IoT. *IEEE Transactions on Network and Service Management* **2021**, *18*, 3297–3308, doi:10.1109/TNSM.2020.3046906.
109. Nguyen, M. A Trust-Based Mechanism for Avoiding Liars in Referring of Reputation in Multiagent System. *International Journal of Advanced Research in Artificial Intelligence* **2015**.
110. Chen, Y.; Feng, L.; Zhao, Q.; Tian, L.; Yang, L. ARS-Chain: A Blockchain-Based Anonymous Reputation-Sharing Framework for E-Commerce Platforms. *Mathematics* **2024**, *12*, 1480, doi:10.3390/math12101480.
111. Bouchiha, M.A.; Ghamri-Doudane, Y.; Rabah, M.; Champagnat, R. DARS: Empowering Trust in Blockchain-Based Real-World Applications with a Decentralized Anonymous Reputation System. In Proceedings of the Advanced Information Networking and Applications; Barolli, L., Ed.; Springer Nature Switzerland: Cham, 2024; pp. 48–61.
112. Castelfranchi, C.; Falcone, R.; Marzo, F. Cognitive Model of Trust as Relational Capital. In Proceedings of the 10th Workshop on Trust, Privacy, Deception and Fraud in Agent Societies, Honolulu, Hawaii; 2007.
113. Aref, A.; Tran, T. Acting as a Trustee for Internet of Agents in the Absence of Explicit Feedback. In Proceedings of the E-Technologies: Embracing the Internet of Things; Aïmeur, E., Ruhi, U., Weiss, M., Eds.; Springer International Publishing: Cham, 2017; pp. 3–23.

114. Tran, T.T.; Cohen, R.; Langlois, E.; Kates, P. Establishing Trust in Multiagent Environments: Realizing the Comprehensive Trust Management Dream. *TRUST@ AAMAS* **2014**, *1740*, 35–43.
115. Houhamdi, Z.; Athamena, B. Multi-Agents Collaboration in Open System. *IAJIT* **2021**, doi:10.34028/iajit/18/3A/2.
116. Latif, R. ConTrust: A Novel Context-Dependent Trust Management Model in Social Internet of Things. *IEEE Access* **2022**, *10*, 46526–46537, doi:10.1109/ACCESS.2022.3169788.
117. Watkins, C.J.C.H.; Dayan, P. Q-Learning. *Mach Learn* **1992**, *8*, 279–292, doi:10.1007/BF00992698.
118. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT press Cambridge, 1998; Vol. 1;.
119. Hamza, N.; Majid, M.A.; Hujainah, F. SIM-PFED: A Simulation-Based Decision Making Model of Patient Flow for Improving Patient Throughput Time in Emergency Department. *IEEE Access* **2021**, *9*, 103419–103439, doi:10.1109/ACCESS.2021.3098625.
120. Kim, J.-K. Enhancing Patient Flow in Emergency Departments: A Machine Learning and Simulation-Based Resource Scheduling Approach. *Applied Sciences* **2024**, *14*, 4264, doi:10.3390/app14104264.
121. Castanheira-Pinto, A.; Gonçalves, B.S.; Lima, R.M.; Dinis-Carvalho, J. Modeling, Assessment and Design of an Emergency Department of a Public Hospital through Discrete-Event Simulation. *Applied Sciences* **2021**, *11*, 805, doi:10.3390/app11020805.
122. De Santis, A.; Giovannelli, T.; Lucidi, S.; Messedaglia, M.; Roma, M. A Simulation-Based Optimization Approach for the Calibration of a Discrete Event Simulation Model of an Emergency Department. *Ann Oper Res* **2023**, *320*, 727–756, doi:10.1007/s10479-021-04382-9.
123. Dosi, C.; Iori, M.; Kramer, A.; Vignoli, M. Successful Implementation of Discrete Event Simulation: Integrating Design Thinking and Simulation Approach in an Emergency Department. *Production Planning & Control* **2023**, *34*, 1233–1247, doi:10.1080/09537287.2021.1996651.
124. Moyaux, T.; Liu, Y.; Bouleux, G.; Cheutet, V. An Agent-Based Architecture of the Digital Twin for an Emergency Department. *Sustainability* **2023**, *15*, 3412, doi:10.3390/su15043412.
125. Mesas, F.; Taboada, M.; Rexachs, D.; Epelde, F.; Wong, A.; Luque, E. A Customizable Agent-Based Simulation Framework for Emergency Departments. In Proceedings of the Computational Science – ICCS 2025; Lees, M.H., Cai, W., Cheong, S.A., Su, Y., Abramson, D., Dongarra, J.J., Sloot, P.M.A., Eds.; Springer Nature Switzerland: Cham, 2025; pp. 46–53.
126. Godfrey, T.; Zschaler, S.; Batra, R.; Douthwaite, S.; Edgeworth, J.; Edwards, M.; Miles, S. Supporting Emergency Department Risk Mitigation with a Modular and Reusable Agent-Based Simulation Infrastructure. In Proceedings of the 2023 Winter Simulation Conference (WSC); September 2023; pp. 162–173.
127. Xuan, X.; Wu, Q.; Yang, L.; Cao, W.; Zhang, Z.; Zheng, Y. Interaction of Nursing Efficiency and Spatial Design in Different Departments: An Agent-Based Modeling Approach. *Frontiers of Architectural Research* **2026**, *15*, 1–23, doi:10.1016/j.foar.2025.04.001.

128. Yuan, H.; Zhou, Y. The Impact of Crowds on Visibility in Emergency Department: Integrating Agent-Based Simulation and Space Syntax Analysis. *Frontiers of Architectural Research* **2025**, *14*, 1398–1414, doi:10.1016/j.foar.2025.01.007.
129. Evans, J.C.; Evans, M.B.; Lingard, L. Team Cognition in Healthcare Simulation: A Framework for Deliberate Measurement. *Adv Simul* **2025**, *10*, 12, doi:10.1186/s41077-025-00333-7.
130. McAvoy, S.; Staib, A.; Treston, G. Can a System Dynamics Model of the Emergency Department Show Which Levers Reduce Bottlenecks and Delays to Improve Access to Care? *Systems Research and Behavioral Science* **2021**, *38*, 61–79, doi:10.1002/sres.2663.
131. England, T.; Brailsford, S.; Evenden, D.; Street, A.; Maynou, L.; Mason, S.M.; Preston, L.; Burton, C.; Van Oppen, J.; Conroy, S. Examining the Effect of Interventions in Emergency Care for Older People Using a System Dynamics Decision Support Tool. *Age Ageing* **2023**, *52*, afac336, doi:10.1093/ageing/afac336.
132. Wong, A.H.; Sabounchi, N.S.; Roncallo, H.R.; Ray, J.M.; Heckmann, R. A Qualitative System Dynamics Model for Effects of Workplace Violence and Clinician Burnout on Agitation Management in the Emergency Department. *BMC Health Serv Res* **2022**, *22*, 75, doi:10.1186/s12913-022-07472-x.
133. Muttalib, F.; Ballard, E.; Langton, J.; Malone, S.; Fonseca, Y.; Hansmann, A.; Remy, K.; Hovmand, P.; Doctor, A. Application of Systems Dynamics and Group Model Building to Identify Barriers and Facilitators to Acute Care Delivery in a Resource Limited Setting. *BMC Health Serv Res* **2021**, *21*, 26, doi:10.1186/s12913-020-06014-7.
134. Yousefi, M.; Yousefi, M. Human Resource Allocation in an Emergency Department: A Metamodel-Based Simulation Optimization. *Kybernetes* **2019**, *49*, 779–796, doi:10.1108/K-12-2018-0675.
135. Ortiz-Barrios, M.; Cuenca, L.; Arias-Fonseca, S.; McClean, S.; Pérez-Aguilar, A. Nurse Staffing Management in the Context of Emergency Departments and Seasonal Respiratory Diseases: An Artificial Intelligence and Discrete-Event Simulation Approach. *J Med Syst* **2025**, *49*, 106, doi:10.1007/s10916-025-02242-8.
136. Wang, X.; Matta, A.; Geng, N.; Zhou, L.; Jiang, Z. Simulation-Based Emergency Department Staffing and Scheduling Optimization Considering Part-Time Work Shifts. *European Journal of Operational Research* **2025**, *321*, 631–643, doi:10.1016/j.ejor.2024.09.020.
137. Redondo, E.; Bélanger, V.; Ruiz, A. A Simulation–Optimization Approach for Defining Long-Term Staffing Levels at Healthcare Facilities. *Flex Serv Manuf J* **2025**, doi:10.1007/s10696-025-09607-8.
138. Anjana, S.; Singhal, S. Simulating Trust Dynamics in Healthcare Networks: A Multi-Agent Game-Theoretic Approach. In Proceedings of the Innovations in Communication Networks: Sustainability for Societal and Industrial Impact; Bhateja, V., Abdul Hameed, V., Udgata, S.K., Azar, A.T., Eds.; Springer Nature: Singapore, 2025; pp. 111–124.
139. Savchenko, E.; Bunimovich-Mendrazitsky, S. Investigation toward the Economic Feasibility of Personalized Medicine for Healthcare Service Providers: The Case of Bladder Cancer. *Front. Med.* **2024**, *11*, doi:10.3389/fmed.2024.1388685.
140. Gummadi, H.S.B. AI-Driven Multimodal Workflow Optimization for Personalized Patient-Centered Care. *World Journal of Advanced Engineering Technology and Sciences* **2025**, *15*, 555–563.

141. Hodgson, N.R.; Saghafian, S.; Martini, W.A.; Feizi, A.; Orfanoudaki, A. Artificial Intelligence-Assisted Emergency Department Vertical Patient Flow Optimization. *Journal of Personalized Medicine* **2025**, *15*, 219, doi:10.3390/jpm15060219.
142. Chan, C.W.; Huang, M.; Sarhangian, V. Dynamic Server Assignment in Multiclass Queues with Shifts, with Applications to Nurse Staffing in Emergency Departments. *Operations Research* **2021**, *69*, 1936–1959, doi:10.1287/opre.2020.2050.
143. Hebb, D.O. *The Organization of Behavior: A Neuropsychological Theory*; Psychology press, 2005;
144. Cohen, P.R. Empirical Methods for Artificial Intelligence.
145. Fotia, L.; Delicato, F.; Fortino, G. Trust in Edge-Based Internet of Things Architectures: State of the Art and Research Challenges. *ACM Comput. Surv.* **2023**, *55*, 182:1-182:34, doi:10.1145/3558779.
146. Wei, L.; Wu, J.; Long, C. Enhancing Trust Management via Blockchain in Social Internet of Things. In Proceedings of the 2020 Chinese Automation Congress (CAC); August 2020; pp. 159–164.
147. Amiri-Zarandi, M.; Dara, R.A. Blockchain-Based Trust Management in Social Internet of Things. In Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech); December 2020; pp. 49–54.
148. Hankare, P.; Babar, S.; Mahalle, P. Trust Management Approach for Detection of Malicious Devices in SIoT. *Teh. glas. (Online)* **2021**, *15*, 43–50, doi:10.31803/tg-20210204180217.
149. Talbi, S.; Bouabdallah, A. Interest-Based Trust Management Scheme for Social Internet of Things. *J Ambient Intell Human Comput* **2020**, *11*, 1129–1140, doi:10.1007/s12652-019-01256-8.
150. Fabi, A.K.; Thampi, S.M. A Psychology-Inspired Trust Model for Emergency Message Transmission on the Internet of Vehicles (IoV). *International Journal of Computers and Applications* **2022**, *44*, 480–490, doi:10.1080/1206212X.2020.1814557.
151. Bahutair, M.; Bouguettaya, A.; Neiat, A.G. Multi-Perspective Trust Management Framework for Crowdsourced IoT Services. *IEEE Transactions on Services Computing* **2022**, *15*, 2396–2409, doi:10.1109/TSC.2021.3052219.
152. Sagar, S.; Mahmood, A.; Sheng, Q.Z.; Zhang, W.E. Trust Computational Heuristic for Social Internet of Things: A Machine Learning-Based Approach. In Proceedings of the ICC 2020 - 2020 IEEE International Conference on Communications (ICC); June 2020; pp. 1–6.
153. Sagar, S.; Mahmood, A.; Sheng, M.; Zaib, M.; Zhang, W. Towards a Machine Learning-Driven Trust Evaluation Model for Social Internet of Things: A Time-Aware Approach. In Proceedings of the MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services; Association for Computing Machinery: New York, NY, USA, December 9 2021; pp. 283–290.
154. Aalibagi, S.; Mahyar, H.; Movaghar, A.; Stanley, H.E. A Matrix Factorization Model for Hellinger-Based Trust Management in Social Internet of Things. *IEEE*

Transactions on Dependable and Secure Computing **2022**, *19*, 2274–2285, doi:10.1109/TDSC.2021.3052953.

155. Ullah, F.; Salam, A.; Amin, F.; Khan, I.A.; Ahmed, J.; Zaib, S.A.; Choi, G.S. Deep Trust: A Novel Framework for Dynamic Trust and Reputation Management in the Internet of Things (IoT)-Based Networks. *IEEE Access* **2024**, *12*, 87407–87419, doi:10.1109/ACCESS.2024.3409273.
156. Alemneh, E.; Senouci, S.-M.; Brunet, P.; Tegegne, T. A Two-Way Trust Management System for Fog Computing. *Future Generation Computer Systems* **2020**, *106*, 206–220, doi:10.1016/j.future.2019.12.045.
157. Tu, Z.; Zhou, H.; Li, K.; Song, H.; Yang, Y. A Blockchain-Based Trust and Reputation Model with Dynamic Evaluation Mechanism for IoT. *Computer Networks* **2022**, *218*, 109404, doi:10.1016/j.comnet.2022.109404.
158. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533, doi:10.1038/nature14236.

Author's Statement:

I hereby expressly declare that, according to the article 8 of Law 1559/1986, this dissertation is solely the product of my personal work, does not infringe any intellectual property, personality and personal data rights of third parties, does not contain works/contributions from third parties for which the permission of the authors/beneficiaries is required, is not the product of partial or total plagiarism, and that the sources used are limited to the literature references alone and meet the rules of scientific citations.