Advancements in Federated Fog Architectures for Enhanced Quality-of-Service in IoT and IoV Applications

by

Ahmad HAMMOUD

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, OCTOBER 12, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Zbigniew Dziong, Thesis supervisor
Department of Electrical Engineering, École de technologie supérieure

Mr. Hadi Otrok, Thesis Co-Supervisor
Department of EECS, Khalifa University of Science and Technology

Mr. Azzam Mourad, Thesis Co-Supervisor
Department of Mathematics and Computer Science, Lebanese American University

Mr. Chamseddine Talhi, Chair, Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mr. Kim Khoa Nguyen, Member of the Jury
Department of Electrical Engineering, École de technologie supérieure

Mr. Chadi Assi, External Independent Examiner
Concordia Institute for Information Systems Engineering, Concordia University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON OCTOBER 4, 2023

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This dissertation aims to explore various aspects of federated fog computing which gained significant attention in the last several years. The research produced 4 journal articles. While the first two chapters of this dissertation provide an extensive introduction and background of federated fog architectures, the subsequent chapters are dedicated to the presentation of the journal articles without any modifications. Although each article focuses on a different aspect of federated fog computing, they collectively form a coherent whole and are closely interconnected, contributing to a comprehensive understanding of the main topic.

ACKNOWLEDGEMENTS

I am incredibly grateful for the opportunity to undertake this research and for the immense support I have received throughout this transformative journey. Many individuals have played a pivotal role in shaping my academic journey and providing unwavering support.

I would like to begin by expressing my deepest appreciation to my supervisor, Professor Zbigniew Dziong, for his deep guidance, support, and invaluable insights throughout the last 4 years. His academic knowledge was a tremendous asset in shaping my research direction. I was lucky to be given the opportunity to work under his supervision in this chapter of my life. He provided a very supportive work environment to help me realize the findings of this thesis.

I am also immensely grateful to my co-supervisor Professor Hadi Otrok and his integration to provide me with technical solutions for approaching the problems we were addressing in this research. His valuable feedback and constructive criticism were crucial to my work. I learned a lot from him as a critical thinker. Also, if ever in need of recommendations for an outstanding dining experience, Professor Hadi's expertise extends beyond academia to include an exceptional knowledge of gastronomy.

It is not often that a co-supervisor occupies a big part of a student's life, but Professor Azzam Mourad breached all traditional boundaries and became an integral part of my support system. Since 2015, he has been not only my academic supervisor, but my mentor, guide, therapist, and wise brother. I consider myself fortunate to have had the chance to work under his guidance for the past 9 years.

I would also like to extend my sincere gratitude to the members of my thesis committee, Professor Chamseddine Talhi and Professor Kim Khoa Nguyen, for their efforts and insightful comments on my dissertation. I was honored to be evaluated by such experienced Professors.

In addition, I would like to express my deepest gratitude to the person who holds the most profound place in my heart, my mother. Her unwavering support and belief in my abilities have been instrumental in my academic success. I am also immensely grateful to the rest of my family and friends for their continuous encouragement throughout this challenging academic endeavor. Their motivation has been a constant source of strength.

Thank you all for your unconditional support toward achieving the highest academic milestone of my life.

Avancées dans les architectures fédérées de brouillard pour une amélioration de la qualité de service dans les applications IoT et IoV

Ahmad HAMMOUD

RÉSUMÉ

Le monde devient de plus en plus connecté avec la présence croissante d'objets intelligents et les progrès des systèmes de transport intelligents (ITS). Cependant, les retards du réseau causent des perturbations et une réduction de la qualité de service (QoS) pour les applications Internet des objets (IoT) et Internet des véhicules (IoV) lorsque les utilisateurs exécutent des services en temps réel critiques. Le fog computing, avec son placement stratégique de serveurs, émerge comme une solution, mais il nécessite de résoudre le manque de puissance de traitement pour les applications IoT et IoV critiques. Parallèlement, le federated fog computing peut offrir une adaptation dynamique aux changements environnementaux et faciliter la communication.

Cette thèse vise à contribuer à une architecture de fédération de fog complète et sécurisée pour améliorer la QoS des applications IoT en général et de l'architecture des applications de conduite autonome en IoV en particulier. Elle se concentre sur plusieurs défis, notamment l'absence d'enquête sur une architecture de fédération de fog complète, l'instabilité potentielle des fédérations de fog, l'absence de prise en charge de la mobilité en IoV, les complications liées à l'apprentissage fédéré pour IoV et l'impact des fournisseurs de fog peu fiables.

Les objectifs de recherche sont les suivants : développer une architecture de fédération de fog complète et efficace avec un mécanisme de regroupement, créer un mécanisme de formation robuste pour empêcher les fournisseurs de changer de fédération, étendre la formation de fédération de fog pour prendre en charge la mobilité, soutenir l'apprentissage fédéré des véhicules et introduire un mécanisme de confiance et de réputation dans la formation de fédération de fog.

Nous complétons les progrès actuels de la recherche sur le federated fog computing dans la littérature existante en ajoutant différents modules pour faciliter l'objectif d'amélioration de la QoS des applications d'apprentissage fédéré en IoV. Tout d'abord, nous étudions une architecture nouvelle et complète pour le concept de fédération de fog et proposons une approche adaptative, intelligente et dynamique de formation de fédération en utilisant l'apprentissage automatique et les algorithmes génétiques. De plus, nous abordons le problème de l'instabilité au sein des fédérations de fog en proposant un algorithme décentralisé basé sur la théorie des jeux évolutifs. En outre, nous élargissons notre domaine pour couvrir un environnement plus dynamique : l'Internet des véhicules. Pour satisfaire les utilisateurs mobiles, la mobilité doit être prise en compte, nous nous appuyons donc sur un mécanisme de formation de fédération de fog en vol dans lequel nous prenons en compte la mobilité des appareils pour leur fournir une bonne qualité de service en utilisant la théorie des jeux. De plus, nous présentons une architecture d'apprentissage fédéré basée sur l'horizontale renforcée par les fédérations de fog pour prendre en charge la formation sur l'appareil avec la QoS demandée. Enfin, nous étendons le mécanisme de formation utilisé pour le federated fog computing en introduisant une infrastructure de

chaîne de blocs pour gérer les tâches administratives des fédérations et sécuriser le processus de formation.

Des ensembles de données réels sont utilisés pour évaluer l'architecture proposée et les mécanismes de formation. Les résultats montrent une amélioration notable du débit et une diminution du temps de réponse pour les services demandés, ainsi que la stabilisation des fédérations de fog.

Mots-clés: Informatique décentralisée fédérée, Apprentissage fédéré, Internet des véhicules, Regroupement, Stabilité, Qualité de service, Théorie des jeux

Advancements in Federated Fog Architectures for Enhanced Quality-of-Service in IoT and IoV Applications

Ahmad HAMMOUD

ABSTRACT

Network delays cause disturbance and reduction in the Quality-of-Service (QoS) for Internet-of-Things (IoT) and Internet-of-Vehicles (IoV) applications while end-users are running critical real-time services. Federated fog computing emerged as a viable solution to overcome such a problem. By merging resources from multiple fog providers and agreeing on a service level agreement, the federated infrastructure can offer the opportunity to adapt to dynamic environmental changes and facilitate vehicle-to-vehicle communication.

This thesis aims to contribute to the comprehensive and secure fog federation architecture to enhance the QoS for IoT in general, and autonomous driving applications in IoV in specific. It focuses on several challenges, including a lack of investigation into a comprehensive federated fog architecture, potential instability in fog federations, lack of support for mobility in IoV, complications in federated learning for IoV, and the impact of untrustworthy fog providers. The main objectives of this thesis include developing a comprehensive and efficient federated fog computing architecture, creating a robust formation mechanism to limit providers from switching federations, extending fog federation formation to support mobility, supporting vehicular federated learning applications, and ensuring that fog federation formation considers trust and reputation during the formation of the architecture and its maintaining phases.

We complement the current research progress about federated fog in the literature by adding various modules to facilitate the goal of enhancing the QoS of Federated Learning applications in IoV. First, we investigate a novel architecture for the federated fog concept and propose an adaptive, intelligent, and dynamic federation formation approach using Machine Learning and Genetic Algorithms. Moreover, we address the problem of instability within fog federations by proposing a decentralized algorithm based on evolutionary game theory. Furthermore, we expand our area to cover a more dynamic environment; the Internet-of-Vehicles. To satisfy mobile users, mobility should be considered, thus, we rely on a fog federation formation mechanism on the fly where we consider the mobility of the devices to provide them a good service quality using game theory. In addition, we present a horizontal-based federated learning architecture empowered by fog federations to support on-device training with the requested QoS. Finally, we extend the formation mechanism used for federated fog computing by introducing a Blockchain infrastructure to handle the federations' administrative tasks and secure the formation process.

Real datasets are used to evaluate the proposed architecture and formation mechanisms. The results show a notable improvement in the throughput and a decrease in the response time for the services requested, in addition to stabilizing the fog federations.

Keywords: Federated Fog Computing, Federated Learning, Internet-of-Vehicle, Clustering, Stability, Quality-of-Service, Game Theory

TABLE OF CONTENTS

Page

INTRODUCTION 1
0.1 Motivation 1
0.2 Problem Statement 3
0.3 Research Objectives 4
0.4 Contributions and Novelty 5
0.5 Author's Publication 9
0.5.1 Journal Publications 9
0.5.2 Conference Publications 9
0.5.3 Collaborative Publications 10
0.6 Thesis Outline 10

CHAPTER 1 PRELIMINARIES 13
1.1 Background 13
1.1.1 Smart Cities 13
1.1.2 Internet of Things 14
1.1.3 Internet of Vehicles 15
1.1.4 Service Provider 16
1.1.5 Federated Learning 17

CHAPTER 2 STABLE FOG FEDERATION FORMATION: AN EVOLUTIONARY GAME THEORETICAL APPROACH 19
2.1 Abstract 19
2.2 Introduction 19
2.3 Related Work 24
2.3.1 Cloud Federation Formation Approaches 24
2.3.2 Fog Service Deployment and Task-Scheduling Approaches 25
2.3.3 Fog Federation-Based Solutions 26
2.3.4 Analysis 27
2.4 System Model and Problem Formulation 27
2.5 Evolutionary Federated Fog Formation 30
2.5.1 Background 31
2.5.2 Game Characteristics 31
2.5.3 Stable Fog Federation Formation 33
2.5.3.1 Initialization 33
2.5.3.2 Player Strategy 35
2.5.3.3 Discussion 36
2.6 Numerical Example 36
2.7 Experimental Evaluation 41
2.7.1 Experimental Setup 41
2.7.2 Results and Discussion 41

2.8 Conclusion 44

CHAPTER 3 DYNAMIC FOG FEDERATION SCHEME FOR INTERNET OF VEHICLES 47
3.1 Abstract 47
3.2 Introduction 48
3.3 Related Work 52
3.3.1 Cloud-based Solutions 52
3.3.2 Fog-based Solutions 53
3.4 Proposed Architecture 55
3.5 Problem Formulation 57
3.5.1 Task Invocation 58
3.5.2 Vehicle Trajectory 59
3.5.3 Fog Federations 59
3.6 Merge & Split-based Fog Federation Formation 62
3.6.1 Preliminary Terminology 62
3.6.2 Formation Mechanism 63
3.7 Experimental Evaluation 67
3.7.1 Experimental Setup 67
3.7.2 Simulation Results 68
3.8 Conclusion 73

CHAPTER 4 ON DEMAND FOG FEDERATIONS FOR HORIZONTAL FEDERATED LEARNING IN IOV 75
4.1 Abstract 75
4.2 Introduction 76
4.2.1 Machine Learning in IoV 76
4.2.2 Federated Learning 77
4.2.3 Fog Federations 77
4.2.4 Problem Statement and Objectives 78
4.2.5 Contributions 80
4.3 Related Work 81
4.3.1 Enabling Federated Learning 82
4.3.1.1 Preserving Privacy Approaches 83
4.3.1.2 Active Sampling 83
4.3.1.3 Resource-Aware Solutions 84
4.3.2 Quality-based Solutions 85
4.3.2.1 Service Deployment Strategies 85
4.3.2.2 Cloud and Fog Federations 86
4.3.3 Discussion 87
4.4 Dynamic Horizontal Federated Learning Approach 87
4.4.1 Architecture Components 89
4.4.2 Architecture workflow 90
4.5 Hedonic Federations Formation 92

4.5.1 System Model ... 93
4.5.1.1 IoV Learning Applications ... 93
4.5.1.2 Application Users ... 94
4.5.1.3 Fog Providers ... 94
4.5.1.4 Fog Federations ... 94
4.5.2 Hedonic Games ... 95
4.5.3 Game Formulation ... 96
4.5.4 Federation Formation Algorithm ... 97
4.6 Experimental Evaluation ... 98
4.6.1 Experimental Setup ... 98
4.6.1.1 Dataset ... 99
4.6.1.2 Benchmarking models ... 100
4.6.1.3 Parameters and Applied Methods ... 100
4.6.2 Results and Discussion ... 101
4.6.2.1 Resource Availability ... 101
4.6.2.2 Federated Learning ... 103
4.7 Conclusion and Future Work ... 106

CHAPTER 5 A BLOCKCHAIN-BASED HEDONIC GAME SCHEME FOR REPUTABLE FOG FEDERATIONS ... 109
5.1 Abstract ... 109
5.2 Introduction ... 110
5.3 Related Work ... 114
5.3.1 Coalition-based Solutions ... 114
5.3.2 Blockchain-based Solutions ... 117
5.4 Proposed Framework ... 118
5.4.1 Architecture and Components ... 118
5.4.2 System Model ... 120
5.4.3 Reputable Hedonic Fog Federation Formation Scheme ... 122
5.4.4 Off-Chain Hedonic Algorithm Execution ... 123
5.4.5 Contract-Based Reputation Module ... 126
5.4.5.1 Provider Contract Functions ... 127
5.4.5.2 Coalition Contract Functions ... 127
5.4.6 Security Analysis ... 128
5.4.6.1 Misbehaving Providers ... 128
5.4.6.2 Detection and Mitigation ... 129
5.5 Experimental Evaluation ... 129
5.5.1 Setup ... 129
5.5.1.1 Tools ... 129
5.5.1.2 Datasets ... 130
5.5.1.3 Misc. Provider Mechanics ... 130
5.5.1.4 Benchmark Model ... 131
5.5.1.5 Comparison Metrics ... 131

5.5.2 Results 132
5.6 Conclusion 135

CONCLUSION AND RECOMMENDATIONS 139

LIST OF REFERENCES 141

LIST OF TABLES

Page

Table 1.1 Comparison between Cloud and Fog Computing 18

Table 2.1 Comparison among related work 28

Table 2.2 Definitions 29

Table 2.3 Available Fog Provider 38

Table 2.4 Fog Federations Using K-means 38

Table 2.5 Application Service Providers 38

Table 2.6 Fog Providers' Utility and Payoff 40

Table 2.7 Fog Providers' Utility and Payoff After Convergence 41

Table 3.1 Summary of Notations 58

Table 3.2 Parameters used in simulation 67

Table 4.1 Summary of Notations 93

Table 5.1 Related Work Summary 115

Table 5.2 Provider Contract Functions 127

Table 5.3 Coalition Contract Functions 127

LIST OF FIGURES

Page

Figure 0.1 Knowledge sharing in ITS 2

Figure 0.2 Thesis Novelty Graph 6

Figure 1.1 A smart city 13

Figure 1.2 IoT Adoption 14

Figure 1.3 IoV Networks 15

Figure 2.1 Fog Federations vs. Single For Providers 21

Figure 2.2 Response time of requests with vs. without federations 22

Figure 2.3 Evolutionary Game Theory Flowchart 32

Figure 2.4 Stability 43

Figure 2.5 Total Federations' Payoff 44

Figure 2.6 Response time of requests 45

Figure 2.7 Availability of requests 45

Figure 3.1 Motivation for a Dynamic Fog Federation Formation 50

Figure 3.2 Dynamic Federation Formation Architecture 55

Figure 3.3 Adaptive Coalition Formation Algorithm 66

Figure 3.4 Satisfaction Rate of the end users 68

Figure 3.5 Stability of the formed federations 70

Figure 3.6 Execution time of forming the federations 71

Figure 3.7 Average execution time of forming the federations versus the number of providers 72

Figure 4.1 Migrating services on-demand among federation members 79

Figure 4.2 Dynamic Horizontal Federated Learning Architecture 88

Figure 4.3 Framework Processes Timeline 91
Figure 4.4 Traffic Signs Samples 100
Figure 4.5 Invocation Delay 102
Figure 4.6 Invocation Satisfaction Rate 103
Figure 4.7 Federated Learning Simulation Results: IID settings 105
Figure 4.8 Federated Learning Simulation Results: non-IID settings 106
Figure 5.1 Fog federations interchanging services in IoV 112
Figure 5.2 Reputation-Based Fog Coalitions Architecture for IoV 119
Figure 5.3 Reputable Fog Federation Formation Scheme: Process Timeline 124
Figure 5.4 Reputation Prediction Accuracy 133
Figure 5.5 Service Availability 134
Figure 5.6 Serviced User Rate 136
Figure 5.7 Average Payoff 137

LIST OF ALGORITHMS

Page

Algorithm 2.1 Initial Clustering 34

Algorithm 2.2 Services Deployment 35

Algorithm 2.3 Player Preference 37

Algorithm 2.4 Greedy Player Preference 42

Algorithm 4.1 Federation Formation Mechanism 98

Algorithm 5.1 Federation Formation Mechanism 125

INTRODUCTION

0.1 Motivation

It is undeniable how the world we are living in is becoming more connected. Smart objects are everywhere now, and the underlying computing infrastructure has a lot of potentials to enable seamless communication across devices. In this context, as the world is witnessing advancements in the Intelligent Transportation System (ITS), we can ask ourselves: Are we able to let autonomous cars learn by themselves to adapt to their environments and make use of the driving habits of the drivers? Are we able to transfer knowledge from one car to another while preserving the privacy of these cars? What are the enabling technologies to allow such a seamless supportive learning system in IoV?

Intuitively, when these questions are asked, Artificial Intelligence (AI) comes into our mind. AI has become a key component of the IoV paradigm that allows the development of complex services such as Autonomous Driving systems for improving road safety. The safety of intelligent vehicles' trips strongly relies on how well-trained and prepared the integrated AI systems are. Particularly, the vehicle scans its surroundings using various sets of sensors, including cameras, beams of radar, lidar, ultrasound, GPS navigation, etc... Then, it passes the sensed data to the AI system which, in turn, analyzes it and makes the best decision under given circumstances (e.g., speed up, stop, turn left, etc...). In order for such a complex model to be ready for deployment, a huge data sets is required, and an analytical machine-learning procedure must be carried out to discover statistically significant patterns in such data. Training such a huge stream of data requires capable computing servers such as the ones deployed in Cloud Computing. By having the cloud collecting the data from a significant number of vehicles, it is able to perform intensive computations to extract useful patterns and combine these patterns into one single optimized machine learning model that could be forwarded to the vehicles for deployment. As the number of autonomous vehicles increases, the centralized cloud computing model is not designed to

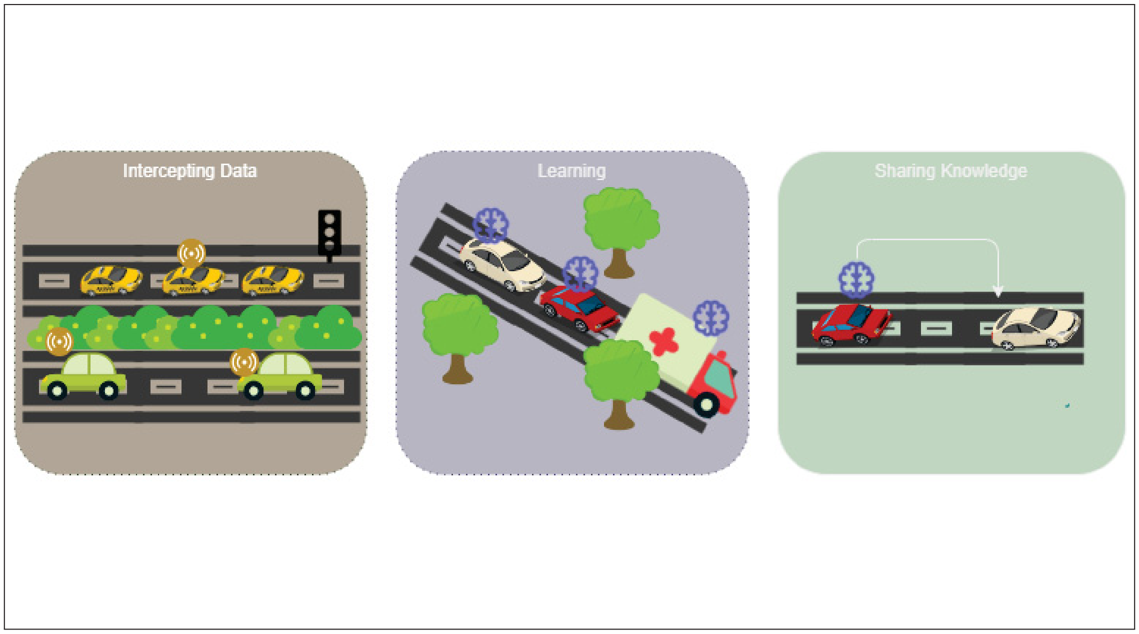


Figure 0.1 Knowledge sharing in ITS

deliver the required real-time processing and analysis capabilities, and also it is not the best candidate to support local vehicle communication (vehicle-to-everything). Fog computing can support such an architecture due to its strategic server placements. However, the main critical challenge is to compensate for the lack of processing power needed when deployed to serve critical IoV applications. In response to this challenge, the concept of federated fog computing emerged as a viable solution. By merging resources from multiple fog providers and agreeing on a service level agreement, the federated infrastructure can offer the opportunity to adapt to dynamic environmental changes and facilitate vehicle-to-vehicle communication.

The motivation behind this thesis stems from the recognition of the crucial role that federated fog computing plays in enabling autonomous driving and improving its architecture for federated learning applications. This thesis aims to contribute to the comprehensive and secure fog federation architecture necessary for enhancing the QoS of autonomous driving applications within an IoV framework while considering various strategies to compensate for the dynamicity of the environment. Motivated by our previous work where we defined the building blocks for fog federations architecture and its formation mechanism (Shamseddine *et al.*, 2020a), in this thesis we address the identified challenges and limitations. This research seeks to unlock the full potential of federated fog computing, enabling a future where autonomous vehicles navigate the roads in a safe, efficient, and precise manner.

0.2 Problem Statement

In this thesis, we focus on several challenges related to the concept of federated fog computing architecture within the context of IoT and IoV. We list below the main challenges related to the implementation of this concept:

- The potential instability of fog federations, caused by providers not dedicating the agreed-upon resources to their respective federation and deviating from it, can lead to a decrease in shared resources and computational capabilities. Such unstable fog federations negatively impact QoS and overall federation payoff.
- The existing fog federation-based formation mechanisms ignore the fast and large area displacements of the end-users in IoV settings that can lead to a degraded service quality due to the change of gateways, making these federations vulnerable to instabilities. Federations' stability is an important factor to maintain a stable performance. Otherwise, federation members might break from their federation causing a further reduction of the agreed-upon QoS.
- Federated learning is still suffering from many infrastructural complications in the IoV context due to its special requirements that are different from most other IoT applications. Thus, there is a need for studying the fog federation in terms of its architecture and formation to ensure adequate service quality and a suitable environment for Autonomous Driving applications.

- Finally, encountering untrustworthy fog providers in the federations will negatively affect the overall performance. An untrustworthy provider should be penalized by the federated fog community. To this extent, the current formation mechanisms of the fog federations ignore the trust and reputation metrics when initializing fog federations.

This thesis aims to address these challenges to overcome the limitations and improve the effectiveness of the federated fog computing architecture.

0.3 Research Objectives

Our main objective is a comprehensive and secure fog federation architecture that can enhance the QoS of the applications for IoT and IoV. To accomplish the aforementioned objective, we focus on the following 4 sub-objectives:

1. Developing a robust formation mechanism for the fog federations and reducing the chances of providers deviating from their federations. This involves designing incentives and disincentives to discourage providers from switching federations and ensuring their commitment to the chosen federation.

2. Extending the traditional fog federation formation process to support mobility in an IoV environment while considering the unoptimized distribution of services to improve the QoS.

3. Supporting learning applications within the context of IoV, namely vehicular federated learning, through studying their requirements and assigning roles for each entity accordingly.

4. Securing the fog federation formation process and considering the trust and reputation of each participant during the formation in order to secure the QoS.

0.4 Contributions and Novelty

In this section, we enumerate the contributions of this thesis to the federated fog computing concept while highlighting the novelty. It is worth noting that we previously addressed the problem with a comprehensive architecture (Shamseddine *et al.*, 2020a). We extend the previous federated fog architecture by adding and modifying modules/components summarized by the following list of contributions:

1. We propose a novel fog formation scheme embedding an evolutionary game theoretical model. Our approach offers to form stable fog federations in which no member has incentives to reallocate his resources elsewhere. We form the initial set of federations using the k-means clustering technique. Afterward, we extend the formation with a learning-based evolutionary game-theoretical model. Such a game studies the conflict and cooperation among fog providers in the presence of dynamic strategies. It encompasses a state (strategy) where no fog provider has incentives to change its current federation, i.e. evolutionary stable strategy. We also propose a latency-aware greedy service placement algorithm to cope with placing the services on the evolved generation in order to maintain relatively short network delays. The evolution from one generation to another is presented using a decentralized algorithm that can be executed by the providers separately for reaching stability. We use EUA Datasets, containing data collected from real IoT devices, to simulate and evaluate our approach while comparing it with a greedy algorithm. Experimental results explore that our proposed approach increases the total payoff for the federations and improves the QoS in terms of stability, response time, and resource availability.

   Novelty: We are the first to address the problem of stability in federated fog computing and analyze it using an evolutionary game theoretical model while proposing a solution to mitigate the federations' instabilities.

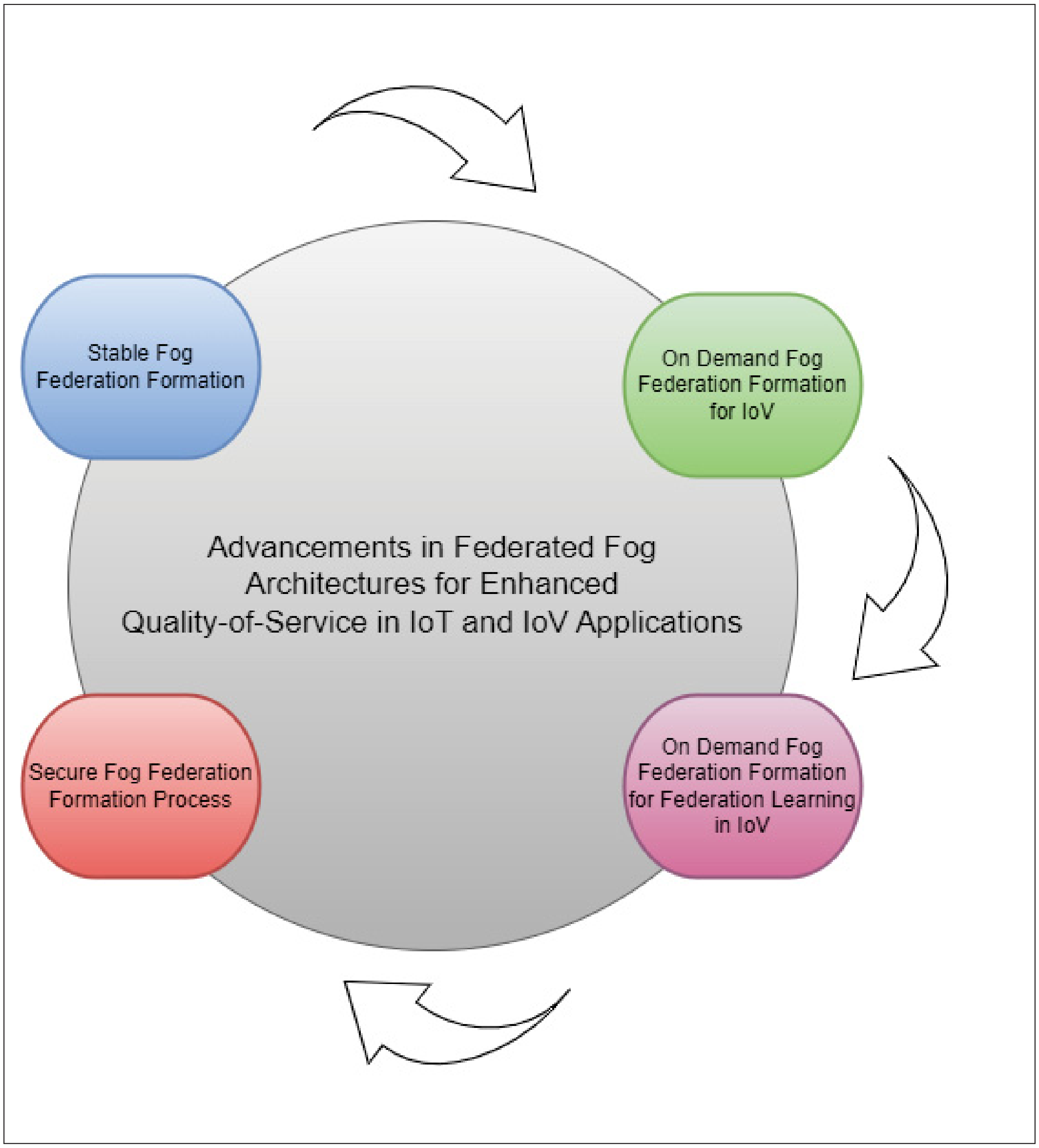


Figure 0.2 Thesis Novelty Graph

2. We devise a game-theoretic approach that relies on the Hedonic Coalition Formation mechanism in order to form fog federations that adapt to the mobile nature of the IoV. Our architecture is fully decentralized and thus does not rely on a central entity to operate. The problem formulation in terms of forming the federations and offloading requests among fog members is formulated as an integer program, then modeled as a Hedonic game. We adopt the Merge & Split as a formation technique, where the federations that

are not satisfied in terms of QoS merge with other federations that would enhance the service performance. The dynamic nature of IoV can frequently trigger the execution of the formation technique to adapt to the changes in this paradigm. In our approach, fog providers have full autonomy to split from the federation and find another federation to join in order to provide satisfactory QoS to its users. To evaluate our scheme, we use SUMO (Simulation of Urban Mobility) simulator that generates realistic vehicular trajectory data to serve in our experiments as moving users. Experimental analysis shows that our approach results in a higher user satisfaction rate in terms of QoS, stability architecture, and a lower execution time when compared to other approaches presented in the literature.
Novelty: We are the first effort to address the limitation of the literature where we consider dynamic coalition formation to maximize QoS in the context of IoV while taking into account the mobility of vehicles. Our fog federation maintenance characteristic grants a recovery stage for the provided services whenever the QoS falls below the threshold.

3. We advance a horizontal federated learning architecture for IoV applications empowered by fog federations. We rely on a Hedonic-game theoretical model for reinforcing the fog federations, i.e. the IT infrastructure, to maintain adequate service quality through migrating services among federation nodes according to the federated learning needs. In our proposal, we consider metrics tailored to IoV settings which makes the previous formations inapplicable, due to the dynamic behavior of the participants. Hence, we demonstrate how to adapt the formation to dynamic federated learning settings. In contrast to the resource-based solutions in the literature, we consider multiple learning applications simultaneously to fully utilize the infrastructure. Our proposed architecture ensures the engagement of more participants in the federated learning process than other approaches proposed in the literature. We evaluate our approach by simulating a process for training a level-1 federated autonomous driving application that can identify traffic signs on the road and alert the driver accordingly. We also compare our approach with other approaches

mentioned in the literature. Experimental evaluation reveals that our mechanism can achieve better model accuracy, lower model loss and response time, and handle more participants in the training process when compared with other approaches.

Novelty: We are the first to design a stable federated fog scheme to assist the federated learning processes within IoV for enabling the training of intelligent vehicular applications.

4. We leverage Ethereum Blockchain technology as an enabler to form reputable fog federations. Unlike other approaches, our proposed approach utilizes the inherent properties of Blockchain to enable decentralized decision-making and reputation management, while making reputation information available across the entire network. We employ a Hedonic game theoretical model to allow decentralized decision-making when establishing federations that are based on their preferences. In addition, we reinforce the formation with an intelligent feedback-based trust establishment mechanism that allows providers to rate the behavior of the other members in their federations through smart contracts to limit the impact of biased feedback. Furthermore, we penalize misbehaving providers by excluding them from the formation game when they fell below a certain reputation threshold. To prepare for our testbed, we rely on the EUA dataset for fog location placement in a certain area, along with vehicular traffic generated by SUMO (Simulation of Urban MObility) in that area. In addition, we use Solidity and Python to program our on-chain and off-chain operations, respectively. Results show that our approach is cost-effective and can yield an increased QoS and profit while reducing the number of misbehaving nodes in the environment when compared to other works in the literature.

   Novelty: Our novelty stems from the combination of both the on-chain (smart reputation contracts) and off-chain (hedonic game) processes, providing a holistic solution for secure federated formation in fog computing environments.

0.5 Author's Publication

0.5.1 Journal Publications

In terms of journals, the contribution to the state-of-the-art was a total of 4 journal articles. We list them below:

1. Hammoud, A., Otrok, H., Mourad, A., & Dziong, Z. (2021). Stable federated fog formation: An evolutionary game theoretical approach. Future Generation Computer Systems, 124, 21-32.

2. Hammoud, A., Kantardjian, M., Najjar, A., Mourad, A., Otrok, H., Dziong, Z., & Guizani, N. (2022). Dynamic fog federation scheme for internet of vehicles. IEEE Transactions on Network and Service Management.

3. Hammoud, A., Otrok, H., Mourad, A., & Dziong, Z. (2022). On demand fog federations for horizontal federated learning in IoV. IEEE Transactions on Network and Service Management, 19(3), 3062-3075.

4. Hammoud, A., Mizouni, R., Singh, S., Otrok, H., Mourad, A., & Dziong, Z. (2023). A Blockchain-based Hedonic Game Scheme for Reputable Fog Federations. IEEE Transactions on Network and Service Management, (Accepted).

0.5.2 Conference Publications

The thesis also resulted in 1 conference that falls under the same umbrella toward achieving an enhanced QoS in IoT and IoV applications:

1. Hammoud, A., Mourad, A., Otrok, H., & Dziong, Z. (2022, August). Data-driven federated autonomous driving. In Mobile Web and Intelligent Information Systems: 18th

International Conference, MobiWIS 2022, Rome, Italy, August 22–24, 2022, Proceedings (pp. 79-90). Cham: Springer International Publishing.

0.5.3 Collaborative Publications

We also contributed with other scholars to extend and ameliorate some modules that are relevant to certain components of our architecture. My primary role was focused on writing, supervising, and contributing to the conceptual aspects of these works. The result of these efforts is summarized in these articles (1 journal and 2 conferences, respectively):

1. Shamseddine, H., Nizam, J., Hammoud, A., Mourad, A., Otrok, H., Harmanani, H., & Dziong, Z. (2020). A novel federated fog architecture embedding intelligent formation. IEEE Network, 35(3), 198-204.

2. Arafeh, M., Hammoud, A., Otrok, H., Mourad, A., Talhi, C., & Dziong, Z. (2022, December). Independent and Identically Distributed (IID) Data Assessment in Federated Learning. In GLOBECOM 2022-2022 IEEE Global Communications Conference (pp. 293-298). IEEE.

3. Yasser, Z., Hammoud, A., Mourad, A., Otrok, H., Dziong, Z., & Guizani, M. Towards Stable Federated Fog Formation using Federated Learning and Evolutionary Game Theory. IEEE GLOBECOM 2023, (Accepted).

0.6 Thesis Outline

- Chapter 1 focuses on introducing some concepts that are relevant to this thesis.
- Chapter 2 presents the work of Journal 1. It addresses the problem of instability within fog federations when they are being formed by proposing a decentralized algorithm based on evolutionary game theory, which stabilizes the federations and enhances QoS for users.

- Chapter 3 presents the work of Journal 2. It introduces an adaptive fog federation formation scheme using game theory for IoV applications, allowing federations to adapt to environmental changes and improve QoS through merging and splitting mechanisms.

- Chapter 4 presents the work of Journal 3. It demonstrates a horizontal-based federated learning architecture empowered by fog federations and a Hedonic game-theoretical model to stabilize these federations.

- Chapter 5 presents the work of Journal 4. It extends the formation mechanism used for federated fog computing by introducing a Blockchain infrastructure to handle the federations' administrative tasks.

CHAPTER 1

PRELIMINARIES

1.1 Background

We discuss, in this section, some of the concepts and paradigms used throughout this thesis.

1.1.1 Smart Cities

Figure 1.1 A smart city

Day by day, people are moving to live in the cities, due to the existence of all facilities within. Such an increase might result in many problems, including traffic jams and resource drainage. In parallel, a smart city is a municipality that makes use of the information analyzed to increase the efficiency of the services provided. As shown in Fig 1.1, Smart cities depend on collecting information from all sources available within the city, such as data generated through the Internet of Things devices, communication networks, and software solutions. Mainly, the purpose of this concept is to improve the quality of life for the citizens. Many cities started integrating this concept, such as Amsterdam, Copenhagen, New York, etc...

Throughout the various networks, a smart city functions by:

1. Collecting the data from the sensors deployed within the city in real-time
2. Analyzing the data by turning them into useful information

3. Forwarding this information to the concerned sectors/parties

4. Generating optimized decisions accordingly

1.1.2 Internet of Things

Internet of Things, or IoT, is the main source of data that a smart city needs. Initially, various sensors are integrated within Things for generating frequent data that, once analyzed, can be used as information to further serve in efficiently managing the city's resources due to the merge of the physical and digital worlds. From a smart light bulb to smart home and a driverless vehicle, IoT devices are everywhere to help control the environment (Fig 3.4a). A statistical study that was recently published by Statista [1] depicts that the number of IoT devices will increase from 15.41 billion in the year 2015 to 75.44 billion devices in 2025, what encourages the investments in the computation resources to handle such received data.

Figure 1.2 IoT Adoption

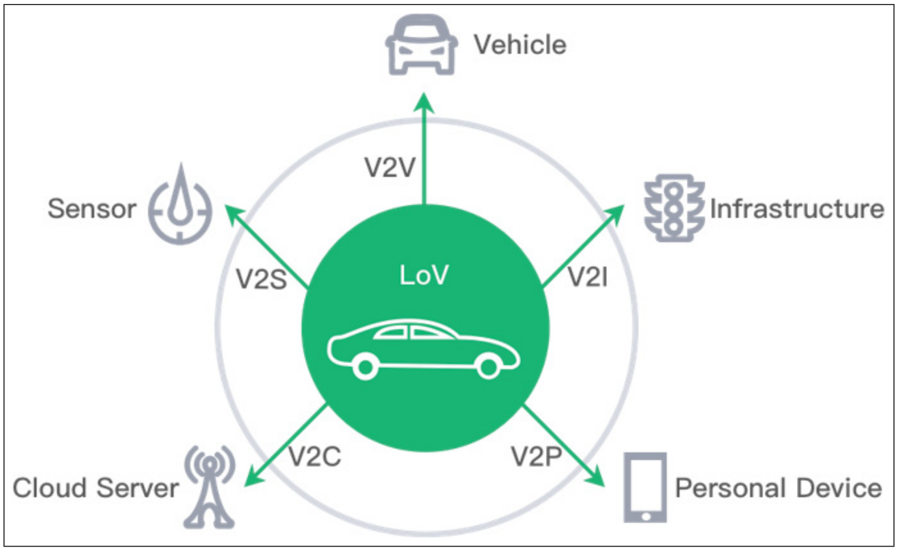


Figure 1.3 IoV Networks

1.1.3 Internet of Vehicles

Internet of Vehicles, or IoV, is an important part of IoT evolution. Mainly, it allows information exchange among vehicles and infrastructures through using a heterogeneous network (as depicted in Fig. 3.4b) such as:

- Vehicle to Vehicle (V2V)
- Vehicle to Infrastructure (V2I)
- Vehicle to Pedestrians (V2P)
- Vehicle to Clouds (V2C)
- Vehicle to Sensors (V2S)

These kinds of networks assist the drivers and the other engaged parties, enabling efficiency and road safety, leading to what is called the Intelligent Transportation System (ITS). Thereupon, the concept of Intelligent Vehicle can be enabled due to the incubating environment of nowadays.

[1] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Simply, Intelligent Vehicles manage to gather data (1) through the sensors embedded within and (2) collected from their surroundings through the networks. Then, vehicles analyze these data to output useful information that can be used to inform - and alert the driver in some cases - about things to consider during the journey.

1.1.4 Service Provider

Service providers are responsible for providing computation, storage, and network resources for the users, in exchange for getting paid for them. We distinguish among 2 different types of service providers: Cloud service providers, and Fog Service Providers.

- **Cloud service providers**: As defined by the National Institute of Standards and Technology (NIST), ‘cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction’ (Hogan, Liu, Sokol & Tong, 2011). In other words, it is the practice of delivering computation resources through the internet. This paradigm provides many advantages to the clients. For instance, it offers flexibility; the clients will not have to host their own Information Technology (IT) infrastructure since it will be managed by experts. The demand for it increased in the past few years for making IT easier for the users (Novet, 2018). Cloud Service Providers (CSP) offer three types of cloud services, listed below:
  - Infrastructure as a Service (IaaS) grants the clients with storage, network, and computation resources to manage them however they need. Amazon Web Services[2] is the leading IaaS provider.

[2] https://aws.amazon.com/

  - Platform as a Service (PaaS) offers a platform for the clients, on which they can deploy the software as they wish. An example of a PaaS for hosting web applications is Google App Engine[3].
  - Software as a Service (SaaS) offers software deployed on the cloud for the end-users. The famous Dropbox[4] is considered as a leading application of this type.

  With the help of Virtualization, several virtual machines can be hosted on the same physical machine, making it more efficient to manage the resources. Statistics have shown that some of the cloud providers are in a continuous increase in profit like Amazon Web Service, which is estimated to acquire 49% more profit than the last year (Networkers, 2018), (Evans, 2018).

- **Fog service providers**: Real-time IoT/IoV applications can be critical in a way they need to make an instant decision and cannot wait for their tasks to be offloaded to the cloud in order to get the response due to the high latency. Hence, the concept of Fog computing that extends the Clouds to the edge of the network. It was proposed by Cisco[5] to overcome the limitations of integrating IoT with the clouds. Things can now offload their tasks and data to the edge of the network, instead of forwarding them to the clouds. However, it becomes costly to deploy powerful Fog servers everywhere, therefore, overwhelming tasks are still being forwarded to the Clouds to be executed there. Table 1.1 summarizes the main differences between Cloud Computing and Fog Computing.

1.1.5 Federated Learning

Transferring data from the users to the Clouds for machine learning training purposes has raised many privacy concerns as it may result in the exposure of their private data either by a session hijacker or by the service provider itself. To address the aforementioned problem, Google developed Federated Learning, a highly privacy-preserving machine learning architecture serving in

[3] https://cloud.google.com/appengine/

[4] https://www.dropbox.com/

[5] https://www.cisco.com/

Table 1.1 Comparison between Cloud and Fog Computing

Parameter	Cloud Computing	Fog Computing
Latency	High	Low
Nodes Mobility	Static	Dynamic
Communication Mode	IP	Wireless communication
Bandwidth Cost	High	Low
Computation Capabilities	Low	High

protecting personal data from being exposed to other parties (Dhole, Thomas & Chandrasekaran, 2016). Such a mechanism consists of having the users train the model themselves independently, and then, forwarding the trained models only to the cloud, without the data, in order to be aggregated together to form one unified machine learning model.

CHAPTER 2

STABLE FOG FEDERATION FORMATION: AN EVOLUTIONARY GAME THEORETICAL APPROACH

Ahmad Hammoud[1] , Hadi Otrok[2] , Azzam Mourad[3] , Zbigniew Dziong[1]

[1] Ecole de Technologie Superieure (ETS), Montreal, Canada
[2] Department of EECS, Khalifa University, Abu Dhabi, United Arab Emirates
[3] Department of Computer science and Mathematics, Lebanese American University, Beirut, Lebanon

Paper published in Elsevier Future Generation Computer Systems on May 24, 2021. doi: 10.1016/j.future.2021.05.021

2.1 Abstract

Instability within fog federations is considered a serious problem that degrades the performance of the provided services. The latter may affect service availability due to fog providers withdrawing their resources. It may either lead to failures for some users' invocations or to an increase in the number of tasks inside the servers' processing queue. Such a critical problem strips the fog paradigm from its main characteristic, the low latency factor. To the best of our knowledge, no work in the literature has addressed the problem of encountering unstable fog federations. Their main concern were increasing the providers' payoff regardless of their behavior. To address the aforementioned limitation, this work studies the federations' stability by modeling the problem as an evolutionary game-theoretical model. Moreover, it devises a decentralized algorithm that implants the Replicator Dynamics model within. We used EUA datasets to test our mechanism in real settings. Experiments explore that the algorithm leads to an evolutionarily stable strategy over time, which stabilizes the federations and improves the Quality-of-Service for the users.

2.2 Introduction

The current revolutionary period we are living in was science fiction a few decades ago. Technology redefined the way people interact with their surroundings. In particular, Internet-of-Things (IoT) applications have become a necessary part of our lifestyle. From a smart light bulb, to a smart home and a driverless vehicle, IoT devices are everywhere to improve our quality of life. A statistical study that was recently published by Statista[1] depicts that the number of IoT devices will increase from 15.41 billion in the year 2015 to 75.44 billion devices in 2025. Such an increase encourages investors and stakeholders for investing more in the computational resources to satisfy the huge demand required by IoT devices.

In parallel, cloud providers cannot meet the Quality-of-Service (QoS) requested by the IoT applications due to the high latency between the devices and the cloud servers. Such network delays constitute a barrier for some applications such as health-care and autonomous driving where even small delays are costly (Shih, Chung, Pang, Chiu & Wei, 2016). To address this issue, Cisco[2] proposed a new concept called fog computing, which extends clouds to the edge of the network in order to massively reduce the network delays (Chiang & Zhang, 2016). IoT devices can now request resources from available nearby fog nodes instead of communicating with the relatively far-away cloud servers. Nevertheless, fog servers entail high deployment cost leading to limitations in available resources compared to the clouds (Stantchev, Barnawi, Ghulam, Schubert & Tamm, 2015; Li, Anh, Nooh, Ra & Jo, 2018). Hence, alternative solutions must be explored to satisfy the huge demand for resources by the Application Service Providers (ASPs). Many scholars recently addressed the resource limitation problem by trying to optimally schedule the tasks invoked by the IoT devices (Oueis, Strinati & Barbarossa, 2015; Sun, Dang & Zhou, 2016), whereas others considered overcoming such an issue through placing on-demand fog (Sami & Mourad, 2020). However, such alternatives are not feasible nor efficient when the fog provider, i.e. the party providing fog nodes, runs out of available resources in the geographical location having high demands for computing resources. Thus, federating fog providers would be considered as a convenient solution to overcoming all the aforementioned limitations.

Simultaneously, the concept of fog federations refers to various participants making use of their unallocated resources, instead of keeping them idle. By reaching an agreement, the collaborators will be able to handle more tasks than anyone can handle on its own. The advantages of such collaboration are twofold. On one hand, fog federations allow offloading tasks among servers

[1] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] https://www.cisco.com/

(i.e., fog nodes) belonging to different fog providers for the sake of processing the user request as quickly as possible, thus improving the QoS for the requests. On the other hand, it allows the fog providers to rent out their unused resources and expand their geographical footprints without the need of new points for presence. Hence, through fog federations, services can be deployed on more geographically distributed servers whenever there are spikes in the demands for serving such requests with acceptable QoS. In our experiments, we show that the federations can boost the service quality by improving many factors, such as the response time. In order to figuratively demonstrate the effectiveness of the fog federations, Fig. 2.1 illustrates how the ASPs are renting resources from fog providers to deploy their services. From the other side, users are trying to access these services by sending requests to the servers running the desired applications. Internally, federation members may offload requests to other members within the same federation in order to shorten the waiting delay for the requests. Thus, as illustrated, the latencies for the users are reduced, leading to a faster way of processing the requests when the providers are federating compared to the typical single fog providers.

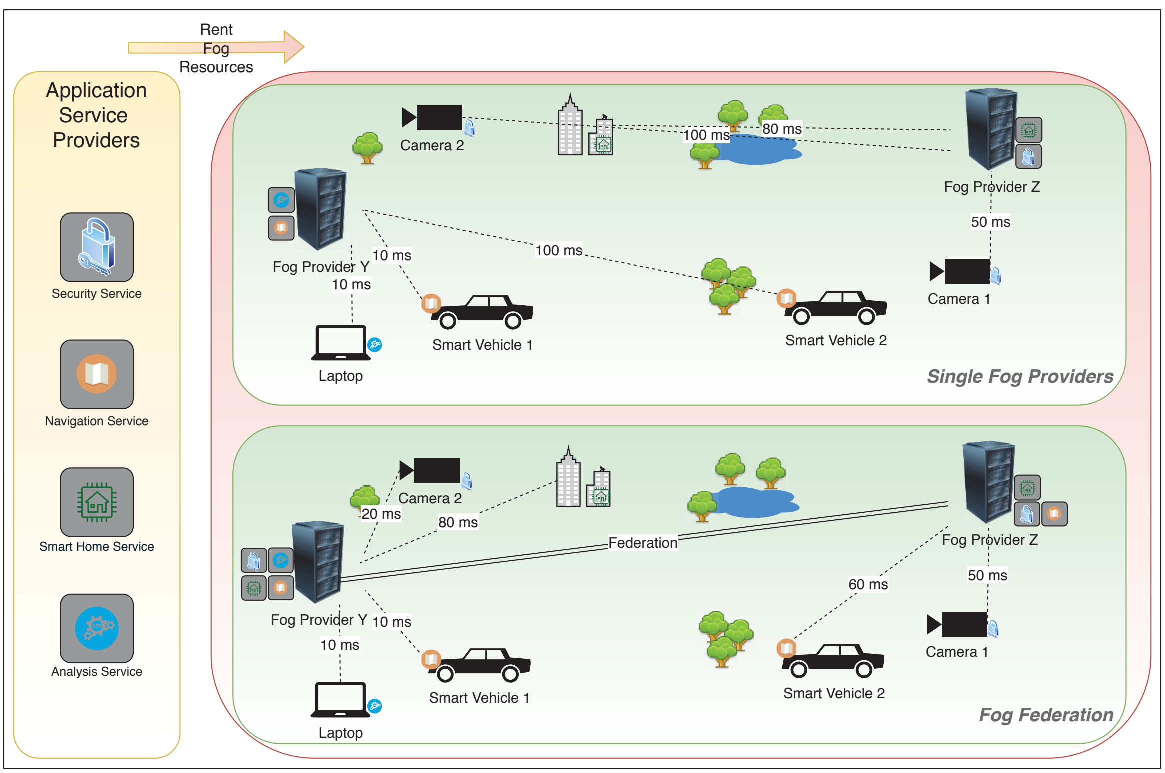


Figure 2.1 Fog Federations vs. Single For Providers

Problem Statement. To motivate the concept of federating fog providers, we show in Fig. 2.2 the response time of serving the requests, with and without federations using models presented in this paper. The X-axis represents the timeline (in terms of hours), whereas the Y-axis is the response time (in milliseconds). The blue and the orange lines are the response time the servers need to process the requests issued by the IoT device with and without federating. We notice that at any specific time, the federation was able to guarantee a satisfactory response time on average due to the cooperation among fog providers. The response time is reduced by almost 26% on average when the services are being handled by fog federations. On the other hand, if the providers show no cooperation, then in some situations the QoS requested by the services could not be reached, leading to penalties. For instance, the fog providers, as rational decision-makers, might feel urged to renege on their commitments and deviate from their federations for seeking better ones that can satisfy them. Such an act reflects negatively on the federations that are suffering from members loss, due to the decrease of the shared resource pool in terms of computational capabilities and points-of-presence. Such federations are referred to as *unstable fog federations*. Hence, how can such fog federations be efficiently

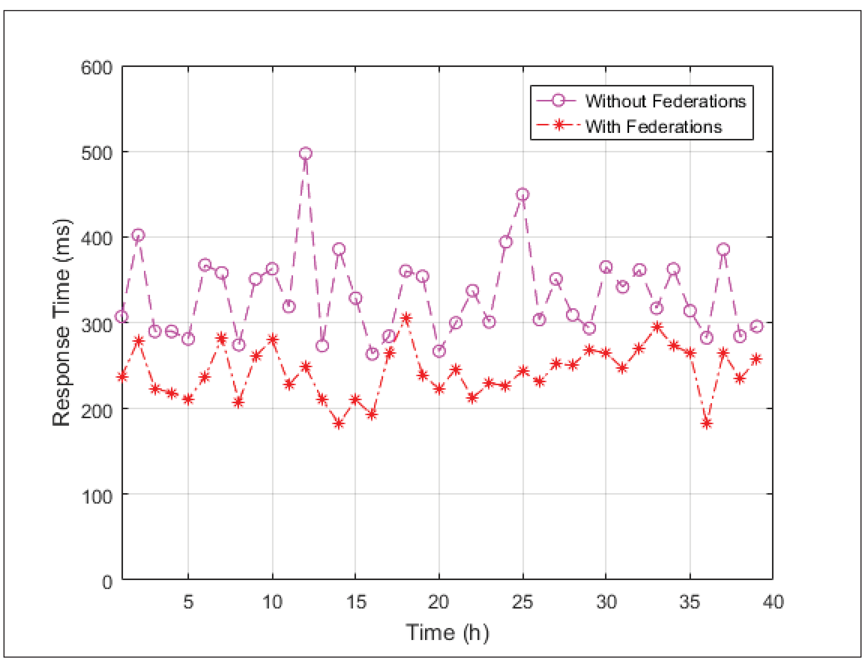


Figure 2.2 Response time of requests with vs. without federations

formed? It is a dilemma that encounters every fog provider due to the fact that the members of the federation directly affect the QoS (Rochwerger *et al.*, 2009). It becomes challenging for fog providers to remain stable, i.e. choose a federation and keep committed to it instead of changing to another one. To the best of our knowledge, none of the proposed solutions have yet tackled the aforementioned problem.

Contributions. In this paper, we address the raised problems by proposing a novel fog formation scheme embedding evolutionary game theoretical model. Our approach offers forming stable fog federations in which no member has incentives to reallocate his resources somewhere else. We form the initial set of federations using the k-means clustering technique. It is an unsupervised learning model that forms clusters based on the similarities among nodes. Afterwards, we extend the formation with a learning-based evolutionary game-theoretical model. Such a game studies the conflict and cooperation among fog providers in the presence of dynamic strategies. It encompasses a state (strategy) where no fog provider has incentives to change its current federation, i.e. evolutionary stable strategy. We also propose a greedy service placement algorithm to cope with placing the services on the evolved generation in order to maintain relatively short network delays. The evolvement from a generation to another is presented using a decentralized algorithm that can be executed by the providers separately for reaching stability. We use EUA Datasets (Lai *et al.*, 2018), containing data collected from real IoT devices, to simulate and evaluate our approach while comparing it with a greedy algorithm. Experimental results explore that our proposed approach increases the total payoff for the federations and improves the QoS in terms of stability, response time, and availability. The main contributions of this work are summarized as follows:

- Adopting an evolutionary game mechanism that simulates the dynamicity of the fog providers, in terms of rational and irrational decision making. To the best of our knowledge, no previous work has ever addressed the dynamic strategies that encounter such a paradigm.

- Forming the initial set of fog federations using the k-means clustering technique. Such a technique allow forming federations based on providers' similarities. In our algorithm, we use the location of the providers to join neighboring fog providers altogether.
- Advancing a latency-aware greedy approach for placing services on the available fog nodes within the federation.
- Devising a decentralized algorithm for fog providers that leads to stabilizing the federations through reaching the evolutionarily stable strategy.

Outline of the paper. The rest of the paper is organized as follows. In Section II, we overview the literature and compare the solutions in the literature with respect to the proposed approach. In Section III, we formulate the federation formation problem. In section IV, we propose our algorithm for solving the problem by employing an initial k-means clustering to form the initial set of federations, and then, studying the dynamicity of the fog providers through advancing an evolutionary game theoretical approach. We provide a numerical example in Section V. After that, we discuss the results of running our algorithm to form the stable fog federations in Section VI. Finally, we give a conclusion in Section VII.

2.3 Related Work

In this section, we give an overview of the literature and highlight on what is needed for advancing a quality fog federation formation mechanism.

2.3.1 Cloud Federation Formation Approaches

Due to the wide range of techniques used for forming cloud federations, we select and discuss the most recent ones in this subsection. In (Hammoud, Mourad, Otrok, Wahab & Harmanani, 2020a), the authors advanced an approach based on genetic algorithms and evolutionary game theory in order to study the problem of forming highly profitable federated clouds, while maintaining stability among the members in the presence of dynamic strategies. In (Hammoud, Otrok, Mourad, Wahab & Bentahar, 2018), the authors addressed the problem of having passive malicious cloud service providers allocating their resources in the cloud federations. They proposed a Maximin game-theoretical model that assists the broker to maximize the detection of the malicious providers. They were able to maximize the detection of malicious providers and improve the profit and QoS of the federations. In (Goiri, Guitart & Torres, 2010), the authors focused on enhancing the profit of cloud providers. They assisted the providers by making optimal decisions on where and when to allocate their computing resources. A linear optimization program was derived in (Rebai, Hadji & Zeghlache, 2015) for helping the providers in a certain federation to regulate their hosting and cooperation decisions on the basis of the encountered workload and the available pool of resources. In (Mashayekhy, Nejad & Grosu, 2015), a formation mechanism was proposed to build a near-optimal federation. Mainly, their algorithm consists of merging and splitting clusters of providers together until reaching the best solution possible. Authors in (Dhole *et al.*, 2016) addressed the formation problem by using trust as a measurement among providers. They claimed to reach stability, profit maximization, and fairness through their formation mechanism. In (Anastasi, Carlini, Coppola & Dazzi, 2017), the authors proposed a genetic approach for cloud brokering. Their approach consists of forming the federations according to the QoS requested by the applications. However, none of the aforementioned works have considered the latency factor in its mechanism, where it is an essential component in forming fog federations for real-time applications. Thus, they cannot be applied on the fog level. Hence, a need for dedicated fog federation formation techniques rises in order to maintain an adequate QoS.

2.3.2 Fog Service Deployment and Task-Scheduling Approaches

Initially, some works considered increasing the QoS by decreasing the latency between the fog and IoT devices. In (Mahmud, Ramamohanarao & Buyya, 2018), the authors presented a latency-aware application module management policy that increases the QoS and optimizes resource usage. Their policy can identify which applications should be deployed on the lower fog nodes (near the devices), and which shall be shifted to the upper fog nodes. In (Oueis *et al.*, 2015), the authors addressed the problem of forming fog clusters to locally process the set of offloaded requests by multiple users. The proposed approach covers both the task scheduling problem and cluster formation. The authors of (Sun *et al.*, 2016) covered the same problem, however, they modeled the formation process as a coalitional game, where each player (fog node) joins its preferred cluster. In (Li *et al.*, 2018), the authors highlighted the problem of deploying fog servers, and how costly it can be. They proposed a dynamic mobile cloudlet cluster policy for fog computing by using cloudlets as a supplement for the fog server for offloading. The problem of allocating a set of docker containers to a set of volunteering devices to provide services on the fly was studied in (Sami & Mourad, 2020). Their main aim was to provide efficiently enough resources for real-time IoT applications requiring computation processing. They used a Multi-Objective Memetic algorithm to solve that problem. However, all of these efforts are not convenient in case of the absence of available resources for scheduling the tasks. In addition, most of these works lack a business-driven model that motivates the participants to show cooperation when deploying services.

2.3.3 Fog Federation-Based Solutions

The concept of federating fog providers is still in its early phases. To the best of our knowledge, there are few published works tackling specifically federations in fog that can directly, or indirectly, enhance the QoS. In (Anglano, Canonico, Castagno, Guazzone & Sereno, 2018b), the authors tackled the concept of federating fog providers for the sake of improving the latter's payoff. They modeled the problem as a Hedonic game with transferable utility where players are the fog providers seeking to maximize their own payoff. The authors in (Veillon, Denninnart & Salehi, 2019b) provided a solution to reduce the latency of streaming video through federations. In particular, their approach was based on evaluating whether it was more convenient to fetch cached video data from neighboring nodes or to process them independently. In (Sharmin, Malik, Rahman & Noor, 2020a), authors proposed a micro-level resource management mechanism for fog federations, where they implemented a price-based workload balancing technique to limit offloading among units relative to other consortium members. However, to the best of our knowledge, the main limitation of the fog federated-based solutions in the literature is the absence of a study conducting the dynamicity of the fog providers. The abandonment of a critical provider on a particular federation may lead to reducing the offered QoS, resulting in a state of dissatisfaction among the ASPs.

2.3.4 Analysis

Table 2.1 highlights the main features of the related work compared to our proposed mechanism. Clearly, none of the aforementioned works in the literature has considered all of the latency factor, dynamicity, independent decision making (or decentralized mechanism), and stability when forming the federations. Such four factors altogether can enhance the quality of the formed fog federation.

2.4 System Model and Problem Formulation

Let us consider a set of fog providers $P = p_1, p_2, ..., p_n$, each of which has a number of servers $S_{p_i} = s_1, s_2, ..., s_m$ located at a particular geographical locations. Such servers are characterized by their processing power, measured in million instructions per second (MIPS). $F = f_1, f_2, ..., f_h$ is the set of federations under which the fog providers unite to form coalitions. We refer to the providers allocated in federation f_i at time t by P_{f_i}. At the same time, ASPs need to offer their services to the users in such a way that the offered QoS should meet the required minimum, otherwise the applications function poorly and ASPs lose some of their users. The federations handle sets of services by deploying them on the providers' servers (fog nodes). The set of applications allocated to federation f_i is represented by set $A_{f_i} = a_{1,f_i}, a_{2,f_i}, ..., a_{o,f_i}$. Likewise, each user is located at a specific location and is enrolled in a set of applications that sends out requests to the servers hosting these applications in order to process at a certain rate. Let the set $Usr_{a_j} = u_{1,a_j}, u_{2,a_j}, ..., u_{q,a_j}$ represent the users requesting service a_j.

The accrued cost C_{p_i} for a certain fog provider p_i is represented by the sum of the operational cost which includes CPU usage cost, memory and storage allocated, and energy usage of all of

Table 2.1 Comparison among related work

	Latency-Aware	Dynamic Providers Behaviours	Decentralized Mechanism	Stable Solution
(Anglano *et al.*, 2018b)	✓	x	✓	✓
(Sun *et al.*, 2016)	✓	x	✓	x
(Hammoud *et al.*, 2020a)	x	✓	x	✓
(Goiri *et al.*, 2010)	x	x	✓	x
(Mashayekhy *et al.*, 2015; Dhole *et al.*, 2016)	x	x	x	✓
(Hammoud *et al.*, 2018; Anastasi *et al.*, 2017; Rebai *et al.*, 2015)	x	x	x	x
(Mahmud *et al.*, 2018; Oueis *et al.*, 2015; Li *et al.*, 2018; Sami & Mourad, 2020; Veillon *et al.*, 2019b; Sharmin *et al.*, 2020a)	✓	x	x	x
Our Solution	✓	✓	✓	✓

its servers $OC(s_j)$, in addition to their traffic cost in terms of allocated bandwidth $TC(s_j)$ as in the following equation:

$$C_{p_i} = \sum_{s_j \epsilon S_{p_i}} \left(OC(s_j) + TC(s_j)\right) \tag{2.1}$$

For a fair monetary distribution to the federation members, every fog provider p_i receives a percentage of federation f_i's total payoff. We consider the utility to be the cost of the servers subtracted from the payoff, divided by the computation power of the servers (i.e. total MIPS

Table 2.2 Definitions

Symbol	Description
P	set of all fog providers
p_i	fog provider i
S_{p_i}	set of all fog nodes (servers) belonging to p_i
s_i	fog node i
F	set of all fog federations
f_i	fog federation i
P_{f_i}	set of all fog providers allocated within f_i
A	set of all applications (services)
A_{f_i}	set of all applications (services) that belong to f_i
a_i	application i
Usr	set of all users
Usr_{a_i}	set of all users of application i
$\sigma_{a_{k,f_i}}$	discount factor of application a_k for federation f_i
α_{a_k}	deduction rate for not meeting a_k's requested QoS
$OC(s_i)$	operational cost of s_i
$TC(s_i)$	traffic cost of s_i
$C(p_i)$	total cost of p_i
$Payment(a_{k,f_i})$	the payment from a_k's ASP to f_i
$Pow(s_i)$	s_i's computation value in terms of MIPS
$U(f_i)$	utility of f_i
$\bar{U}$	average utility of F
$R_{p_{j,f_i}}$	payoff of provider p_j from federation f_i
ρ	the number of fog federations
x	the vector of distribution of available strategies
x_i	the percentage of the population adopting strategy i
$f_i(x)$	fitness function for strategy i
$\nu(x)$	the average fitness by the population

within the federation). Such utility can be expressed as follows:

$$U(f_i) = \frac{\left(\sum_{a_{k,f_i} \epsilon A_{f_i}} Payment(a_{k,f_i}) \times \sigma_{a_{k,f_i}} - \sum_{p_k \epsilon f_i} C_{p_k}\right)}{\sum_{p_k \epsilon P_{f_i}} \sum_{s_l \epsilon S_{p_k}} Pow(s_l)} \tag{2.2}$$

where $Pow(s_i)$ represents the value of server s_i in terms of computing power (i.e. MIPS) and $\sigma_{a_{k,f_i}}$ is considered to be the discount factor that alters the regular payment issued by the application a_k content provider if the federation f_i is not able to meet the minimum requirements and it can be expressed as:

$$\sigma_{a_{k},f_i} = \begin{cases} 1 & \text{if QoS is met,} \\ \alpha_{a_k} & \text{otherwise.} \end{cases} \tag{2.3}$$

where α_{a_k} is the deduction rate due to not meeting the QoS, e.g. the average response time is above the agreed threshold.

Hence, the payoff of a provider p_{j,f_i} can be calculated as the following:

$$R_{p_{j,f_i}} = U(f_i) \times \left(\sum_{s_k \epsilon S_{p_{j,f_i}}} Pow(s_k) \right) \tag{2.4}$$

To stabilize the set of federations, we need to reduce the variability of the payments per shares. The least the difference among the latter, the more satisfied the fog providers would be, leading to fewer deviations from the federations. Such stability can be represented by the equation below:

$$minimize \sum_{f_i \epsilon F} (\bar{U} - U_{f_i})^2 \tag{2.5}$$

where $\bar{U}$ is the average utility which is calculated using the following:

$$\bar{U} = \sum_{f_i \epsilon F} U_{f_i} \times \frac{1}{\rho} \tag{2.6}$$

where ρ is the number of fog federations.

In the next section, we will discuss the formation and stability mechanism used to overcoming the problem of unstable federations.

2.5 Evolutionary Federated Fog Formation

Our proposed scheme is based on defining the formation process as an evolutionary game where each player has a preference function that leads the whole set of federations into its stable state.

2.5.1 Background

Game theory is the science of the optimal decision-making of independent and competing players in a strategic environment. Evolutionary game theory is used in settings where players are not obliged to be reasonable in their decisions (Smith & Price, 1973). Such a game focuses on the dynamics of strategy change and on which among these strategies can persist in these settings. The success of a strategy is directly related to the other players' selected strategies. Hence, a strategy is evaluated by comparing it with the other strategies within the same population. A strategy that shows success will be replicated by other players as well. Once the evolutionarily stable strategy is adopted by a certain population, no player has intentions to deviate from it. Such a strategy can survive invasions of relatively small invaders trying to sabotage it. Hence, it leads to stabilizing the population. In other words, let Pop denote the population adopting an evolutionarily stable strategy A. Pop will not deviate from A if a small number of invaders, adopting strategy B, joined the population. In contrast, the invaders will be forced to switch to A. Suppose that $O(A, B)$ represents the outcome of an individual choosing strategy A facing another one with strategy B. A is stable if it represents a strict Nash-Equilibrium ($[O(A, A)] > [O(A, B)]$), or if $O(A, A) = O(B, A)$ and $O(A, B) > O(B, B)$. If any of these two applies, then no individual has the incentive to deviate from their current strategy, even if the population gets invaded by a few mutants. Fig. 2.3 depicts the evolutionary mechanism. It shows that the population will keep on changing until reaching the state where all players are inheriting satisfactory strategy.

2.5.2 Game Characteristics

We present in this subsection the characteristics of the evolutionary game model to reach the evolutionarily stable strategy. The objective is to proceed from the initialization step to reach the 'End' state. The main components of such a game are (1) the players, (2) strategy, and (3) utility. Below, we break down each of these components and map them to our settings.

- **Players:** The players are the fog providers. Clearly, they are the decision-makers.

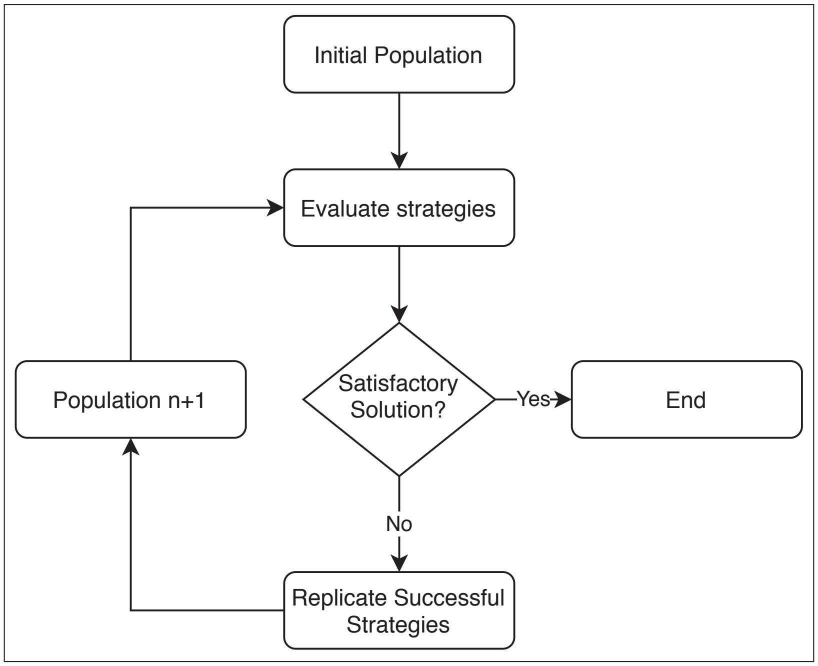


Figure 2.3   Evolutionary Game Theory Flowchart

- **Strategy:** A strategy is represented by a particular fog federation. In particular, a player adopting strategy i can be interpreted as the player allocating its resources in federation i.
- **Utility:** The utility is the player's monetary payoff per 1 unit of allocated resources in a certain federation.

Hence, the problem becomes finding the best formation of fog federation that keeps all the fog providers satisfied with their coalition. To reach such a formation, the fog providers will deviate from their federations if they find a better payoff by joining others. Successful federations are most likely to be joined at time $t + 1$ by unfortunate fog providers who are not satisfied with their selected federation at time t. The term evolution refers to this specific stage, i.e. the change that occurs from a state to another, making the population n to evolve into $n + 1$ where some of the players change their strategy. To represent such an evolution, we employ the replicator dynamics model that expresses the evolutionary dynamics (Schuster & Sigmund, 1983). In particular, we assume that the set $x = x_1, x_2, ..., x_m$ serves as the vector of distribution of available strategies within the targeted population. Intuitively, since all strategies are included in the set x, we can conclude the equation below:

$$\sum_{i=1}^{m} x_i = 1 \tag{2.7}$$

The replicator dynamic's general form is represented by $\dot{x}_i$ and is calculated as:

$$\dot{x}_i = x_i \left[f_i(x) - \nu(x) \right] \tag{2.8}$$

where $f_i(x)$ represents strategy i fitness function and $\nu(x)$ is the average fitness by the population, which can be calculated from:

$$\nu(x) = \sum_{j=1}^{m} x_j f_j(x) \tag{2.9}$$

Mapped to our problem, the fitness function is the payoff of the provider per unit of resources obtained from Eq. 2.2. The replicator dynamics' defined by Eq. 8 shows the percentage of payoff increase for the individuals adopting a successful strategy. Once $\dot{x}_i = 0$ is obtained, the evolutionarily stable strategy is reached.

2.5.3 Stable Fog Federation Formation

To demonstrate our approach, we divide the process into 3 different stages: Initialization, Player Strategy, and Stability.

2.5.3.1 Initialization

To initialize the population, we employ the K-means clustering technique. Such a technique follows the Expectation-Maximization approach. It consists of assigning data points to their nearest cluster (i.e. Expectation). After that, the process of recomputing the centroid for each cluster takes place (i.e. Maximization). Algorithm 1 shows how federations are initialized. The algorithm takes the set of fog providers, represented by P, and desired number of federations $\mathcal{K}$, and outputs the federations with their members. In Lines 1-3, we define and initialize the variables. In Lines 4-7, we set K initial centroids with random values. Then, we iterate on the fog providers and assign the providers according to their nearest centroid. Afterward, we recalculate the centroids. The recalculation function takes into consideration all of the clustered providers to calculate the new midpoint. Such steps are repeated until no more providers change centroids (Lines 8-17). Finally, we initialize a federation for each centroid and allocate the providers inside of them with respect to their centroid (Lines 18-23).

Algorithm 2.1 Initial Clustering

```
   Input: 𝒦, Pop
   Output: F
2  F ← ∅;
4  Centroids ← ∅;
6  terminate ← 0;
8  while K > 0 do
10     C ← RandomPoint;
12     Centroids ← Centroids ∪ C;
14     K ← K - 1;
15 end while
17 while terminate ≠ 1 do
19     terminate ← 1;
21     forall p ∈ P do
23         C ← nearest_Centroid(p);
25         if p.centroid ≠ C then
27             p.centroid ← C;
29             terminate ← 0;
30         end if
31     end forall
33     if terminate ≠ 1 then
35         forall C ∈ Centroids do
37             C ← recalculate(C);
38         end forall
39     end if
40 end while
42 forall C ∈ Centroids do
44     f ← ∅;
46     forall p ∈ P | p.centroid = C do
48         f ← f ∪ p;
49     end forall
51     F ← F ∪ f;
52 end forall
54 return F;
```

Once clustering is done, each federation executes Algorithm 2 to assign services to the provider's servers, with respect to the profit obtained, using a greedy allocation approach. The algorithm takes the set of services and the set of the providers allocated within federation f_i (A_{fi} and P_{fi} respectively) and outputs an allocation list $\mathcal{M}$ which has references to which services shall be deployed on which servers. After initializing the variables (Lines 1 and 2), we evaluate the performance of each server assigned to each service and store them inside the list (Lines 3-7). We order the list by the profit of each assignment in descending order (Line 8). Then, using a greedy technique, we pick the best available server for each service by keeping the best fit in terms of value (Lines 9 and 10).

Algorithm 2.2 Services Deployment

Input: A_{fi}, P_{fi}
Output: $\mathcal{M}$
2 $\mathcal{M} \leftarrow \emptyset$;
4 $value \leftarrow 0$;
6 **forall** $a_l \in A_{fi}$ **do**
8   **forall** $p_j \in P_{fi}$ **do**
10   **forall** $s_k \in s_{k,p_j}$ **do**
12             $value \leftarrow P(a_l) \times \sigma_{a_l,f_i} - C_{s_k}$;
14             $\mathcal{M} \leftarrow \mathcal{M} \cup [a : a_l, s : s_k, v : value]$;
15   **end forall**
16   **end forall**
18   Order $\mathcal{M}$ by v descending;
20   **forall** $m_k \in \mathcal{M}$ **do**
22   removeAll m_l from $\mathcal{M} | m_l.s = m_k.s, m_l.a \neq m_k.a$;
23   **end forall**
24 **end forall**
26 return $\mathcal{M}$;

2.5.3.2 Player Strategy

A player may reflect on its current strategy (i.e. federation) and decide that it might be better for him to switch into another. To imitate such an act, we devise a decentralized algorithm, that can be executed by the fog provider, to decide on which federation to join according to their preferences. Algorithm 3 shows how a fog provider may interact according to the evaluation of the available federations. The algorithm takes as arguments the fog provider's current federation f_i, the set of federations F, and the current vector of distribution of available strategies x. The output of the algorithm is the provider's preferred federation at the current time. Lines 1-6 consists of initializing the parameters. At Lines 7 and 8, the provider calculates the average utility by summing up all federations utilities with regards to the percentage of the population adopting them. At Line 9, if the player notices that they are not getting at least the average utility in terms of value, compared to all other strategies being played, then they consider switching into a more profitable federation. At Line 10, the player filters the federations, such that only the more profitable are being kept. After that, these federations get stored inside F' after being sorted in descending order according to the player's preferences. Lines 11-17 presents how the player sets his next strategy. Finally, the player selects his preferred strategy, according to how preferable a strategy with respect to the others is. It is worth mentioning that we imitate the player's behavior in terms of preferences and with the presence of a slight randomness.

2.5.3.3 Discussion

Evolutionary games are time-aware in the sense that the population is studied and evaluated over time. After setting the initial formation at time t, providers will start acting as rational beings for seeking better federations. To further imitate the dynamicity of such a non-cooperative game and the irrationality of the providers, players are allowed to change strategies at any particular time repetitively until they are satisfied, i.e. they do not have incentives anymore to break from their current federation. Having all players executing the decentralized algorithm over time will result in solving $\dot{x_i} = 0$ for all $x_i \in x$. In other words, it will lead to a state where all the utilities are equal or similar to the average. Thus, any deviation attempt from that state will lead back to it again, as it represents the evolutionarily stable strategy.

2.6 Numerical Example

In this section, we evaluate the proposed scheme in terms of forming stable fog federations. We consider a set of 10 fog providers with different locations and number of participating nodes as

Algorithm 2.3 Player Preference

Input: f_i, F, x
Output: f
$\alpha \leftarrow 0;$
$r \leftarrow random(0, 1);$
$f \leftarrow f_i;$
$\nu(x) \leftarrow 0;$
$F' \leftarrow \emptyset;$
$x' \leftarrow \emptyset;$
forall $f_j \in F$ **do**
    $\nu(x) \leftarrow \nu(x) + x_j \times utility(f_j|F);$
end forall
if $\nu(x) > utility(f_i|F)$ **then**
    $F' \leftarrow sort(F|f_j \in F, utility(f_j|F) > \nu(x);$
    forall $f_j \in F'$ **do**
        $x' \leftarrow x' \cup x_j(\nu - utility(f_j|F));$
    end forall
    forall $f_j \in F'$ **do**
        $\alpha \leftarrow \frac{x_j}{\sum x_k \in x'(x_k)};$
        if $r < \alpha$ **then**
            $f \leftarrow f_j;$
            break;
        end if
    end forall
end if
return $f;$

presented in Table 2.3. We assume that all fog nodes are equal in terms of computing power and total cost (5000 MIPS and 0.5$/h respectively). By applying the initial clustering technique, defined via Algorithm 2.1, we group up neighboring fog provider together. By setting K to 3, we get the federations given in Table 2.4.

Our solution grouped up providers A, B, C, and D into the first federation, providers E, F, I, and J into the second federation, and the remaining providers (G and H) are grouped into the third federation. Table 2.5 represents the ASPs and the federations they have chosen to request computing resources from.

Table 2.3 Available Fog Provider

Fog Provider	Number of Fog Nodes	Latitude	Longitude
A	3	10	10
B	4	12	11
C	2	13	9
D	4	11	13
E	1	1	3
F	2	3	3
G	4	7	13
H	6	6	12
I	2	2	3
J	2	2	4

Table 2.4 Fog Federations Using K-means

Federation	Fog Provider
f_1	A B C D
f_2	E F I J
f_3	G H

Table 2.5 Application Service Providers

ASP #	Agreed Price	Chosen Federation
1	15 $/h	f_1
1	5 $/h	f_1
2	20 $/h	f_2
3	30 $/h	f_2
4	10 $/h	f_3
5	20 $/h	f_3

Then the payoff is distributed based on to Equations 2.2 and 4.2. We set σ is equal to 1. Hence, the utility of federation f_1 is computed as follows:

$$U(f_1) = \frac{1}{13 \times 5000} \times ((15 + 5) \times 1) - (3 \times 0.5 + 4 \times 0.5 + 2 \times 0.5 + 4 \times 0.5) = 0.0002$$

whereas the payoff of the fog providers in exchange to their fog nodes allocated in federation f_1 would be given as:

$$R(A) = 3 \times 5000 \times U(f_1) = 3.12\$$$

$$R(B) = 4 \times 5000 \times U(f_1) = 4.155\$$$

$$R(C) = 2 \times 5000 \times U(f_1) = 2.07\$$$

$$R(D) = 4 \times 5000 \times U(f_1) = 4.155\$$$

The utility of federation f_2 is computed the same way:

$$U(f_2) = \frac{1}{7 \times 5000} \times ((20 + 30) \times 1) - (1 \times 0.5 + 2 \times 0.5 + 2 \times 0.5 + 2 \times 0.5) = 0.0013$$

and the payoff of federation f_2's members are:

$$R(E) = 1 \times 5000 \times U(f_2) = 6.642\$$$

$$R(F) = 2 \times 5000 \times U(f_2) = 13.286\$$$

$$R(I) = 2 \times 5000 \times U(f_2) = 13.286\$$$

$$R(J) = 2 \times 5000 \times U(f_2) = 13.286\$$$

Likewise, federation f_3's utility and its members payoff are calculated as:

$$U(f_3) = \frac{1}{10 \times 5000} \times ((10 + 20) \times 1) - (4 \times 0.5 + 6 \times 0.5) = 0.0005$$

$$R(G) = 4 \times 5000 \times U(f_3) = 10\$$$

$$R(H) = 6 \times 5000 \times U(f_3) = 15\$$$

Table 2.6 summarizes the aforementioned calculations. We notice that some of the providers would not be satisfied, thus starting to deviate from their current federations. For instance,

Table 2.6 Fog Providers' Utility and Payoff

Fog Provider	Federation	Utility	Payoff ($/h)
A	f_1	0.0002	8.5
B	f_1	0.0002	11.33
C	f_1	0.0002	5.67
D	f_1	0.0002	11.33
E	f_2	0.0013	2.83
F	f_2	0.0013	5.67
G	f_3	0.0005	11.33
H	f_3	0.0005	17
I	f_2	0.0013	5.67
J	f_2	0.0013	5.67

providers B and G are both having the same number of fog nodes and specs. However, due to G's allocation in f_3, it is getting an hourly payoff more than 240% of what B is acquiring from f_1. Hence, provider B might get tempted to break from f_1 and join another federation for the sake of improving its payoff. Algorithm 2.3 reflects such behaviour by solving the replicator dynamic's $\dot{x}_i = 0$ in order to obtain a satisfactory solution (i.e. fog federations formation) for all fog providers. Since the algorithm is time aware and executed in a decentralized manner, provider B may execute the algorithm to select the preferred federation at time t by B calculating first $v(x)$:

$$v(x) = 0.0002 \times \frac{13}{30} + 0.0013 \times \frac{7}{30} + 0.0005 \times \frac{10}{30} = 0.00055$$

Afterwards, it compares its utility with $v(x)$. If it does not meet the average utility, then it starts seeking other federations having a utility higher than $v(x)$. In this example, the only available federation that meets such a condition is f_2. So provider B should consider f_2 as the next strategy to adapt at time $t + 1$. After that, all the utilities for the federations affected by such a move (i.e. having B switching from f_1 to f_2) are recalculated as the formation becomes different from what it was at time t. The same process repeats until the algorithm returns the same federation which is represented in Table 2.7. This distribution of resources among federations would remain stable and cannot be sabotaged by invaders, since the algorithm will lead back to the same (or to a similar) distribution. Thus, the QoS will remain stable for the clients.

Table 2.7 Fog Providers' Utility and Payoff After Convergence

Fog Provider	Federation	Utility	Payoff ($/h)
A	f_2	0.00056	3.12
B	f_1	0.00056	4.155
C	f_1	0.00056	2.07
D	f_2	0.00056	4.155
E	f_3	0.00056	6.642
F	f_2	0.00056	13.286
G	f_3	0.00056	10
H	f_2	0.00056	15
I	f_3	0.00056	13.286
J	f_3	0.00056	13.286

2.7 Experimental Evaluation

2.7.1 Experimental Setup

The simulation has been conducted using *Matlab 2016* on a *Windows 10* equipped with Intel Core i7-8750H and 16 GB of RAMs. We used EUA Datasets[3], which have data collected from IoT and Edge devices. We assigned random transmission delays on the links and generated 40 services. The minimum demanded response time by the services varies from 250 to 350 milliseconds. Each IoT device u_i has a set of various services as mentioned in Section III, and a request rate per second ($0 \leq rr_{d_i}^{s_j} \leq 1$) for them. Each request needs processing of [800-1200] million instructions to acquire a result. We limit the number of fog provider to 100 and IoT devices to 600. Each provider has [1-3] available servers, each with a processing power of [4000-6000] MIPS. Finally, we consider $\mathcal{K} = 10$ after applying the Elbow method on evaluating the fittest number of federations, according to the provider's distribution.

2.7.2 Results and Discussion

In this section, we evaluate our evolutionary approach and proposed Algorithms against a greedy approach in terms of stability, response time, availability, player utility and federation payoff. Due

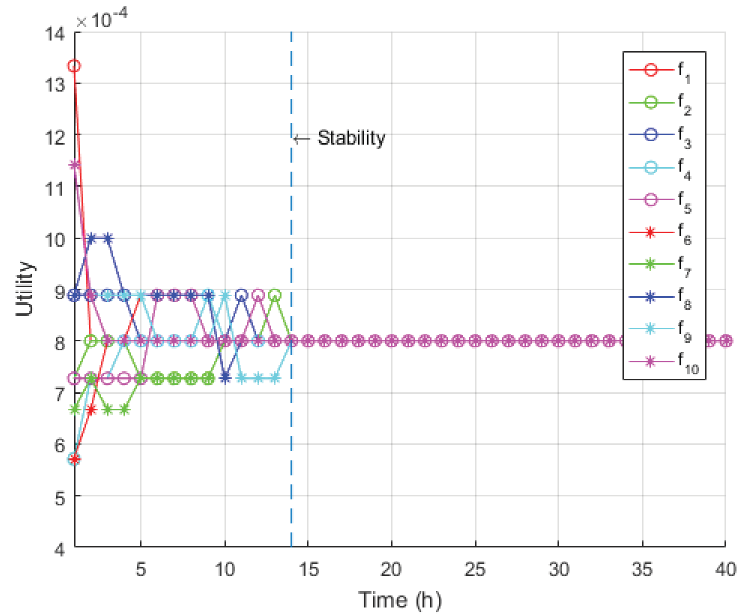
[3] https://github.com/swinedge/eua-dataset

Algorithm 2.4 Greedy Player Preference

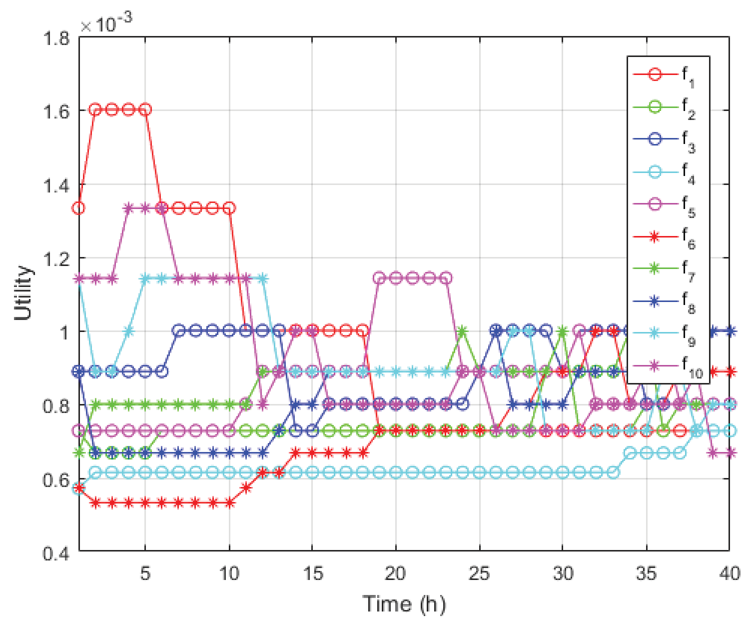
Input: f_i, F
Output: f
$\alpha \leftarrow 0$;
$r \leftarrow random(0, 1)$;
$f \leftarrow f_i$;
$f' \leftarrow f_i$;
$v \leftarrow -\infty$;
forall $f_j \in F$ **do**
    if $v < utility(f_j|F)$ **then**
        $v \leftarrow utility(f_j|F)$;
        $f' \leftarrow f_j$
    end if
end forall
if $r < \alpha$ **then**
    $f \leftarrow f'$;
    break;
end if
return f;

to the absence of a time-aware approach in the literature, we compare our evolutionary game with a greedy approach represented in Algorithm 2.4. Such an algorithm is similar to Algorithm 2.3 in a way that it is also executed by each player separately. However, the fog providers are seeking to reallocate their resources to the best available federation, i.e., the federation with the highest utility.

Fig. 2.4 depicts the utilities of federations. The X-axis represents the timeline and the Y-axis represents the utility of the strategy. We notice that when $x < 14$, the utilities of the federations were not stable at all. However, the federations converge at $x = 14$ on average. This is due to the stability mechanism implanted in Algorithm 3 where the population realizes the evolutionarily stable strategy. On the other hand, in the greedy reallocation approach case, the population could not stabilize at all and the variance of the utilities remained high for the first 40 hours.
The total payoff of all the federations is presented in Fig 2.5 where the Y-axis represents the payoff in terms of USD. According to the simulation, the evolutionary approach is able to always outperform the greedy approach and maintain a higher payoff that stabilizes at $x = 14$, whereas



a) Evolutionary Game Approach



b) Greedy Approach

Figure 2.4 Stability

the greedy approach is suffering from a lack of resources in some federations, which leads to having some unallocated services and the reduction in payoff.

Finally, we compared the evolutionary and greedy approaches together in terms of services' response time and availability in Fig. 2.6 and Fig. 5.5, respectively. As usual, the X-axis is the time-line for both figures, whereas the Y-axis is the response time in milliseconds for Fig. 2.6, and the percentage of availability for Fig. 5.5. Both approaches were able to decrease the response time and increase the availability from the initial formation. However, the evolutionary game outperformed the greedy approach and stabilized the services at full availability and a lower response time due to the stability mechanism reached at $x = 14$, whereas the greedy approach still suffered from a lack of a satisfactory strategy that pleases the participants and reduces their obligations to deviate from their federations.

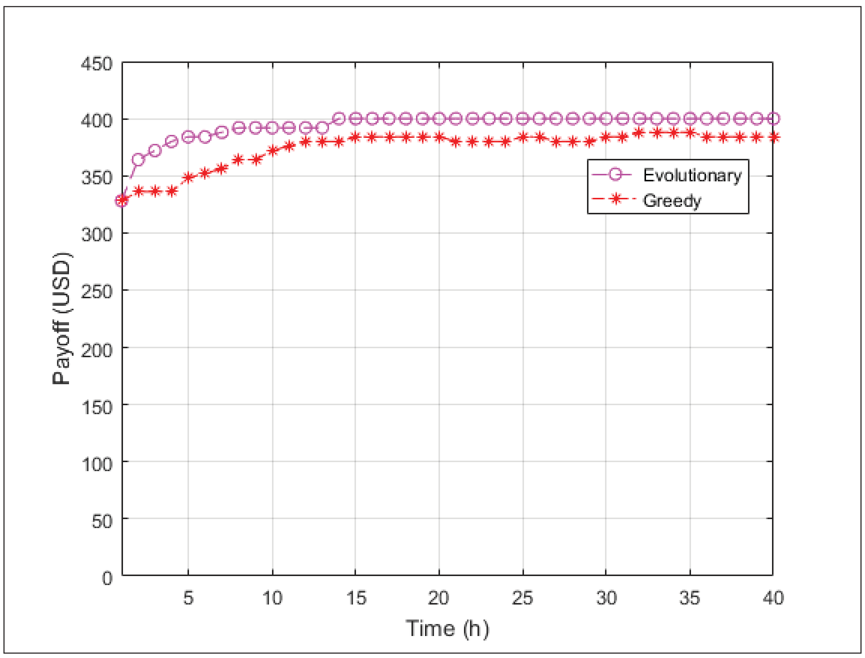


Figure 2.5 Total Federations' Payoff

2.8 Conclusion

Fog federation is a concept worth exploring since it helps to increase the computational capabilities of the fog providers and can provide improved QoS for real-time applications. On the other hand, federations may suffer from instabilities due to the providers' dynamicity that may lead some providers to leave their coalitions and join others that are more profitable. In this paper,

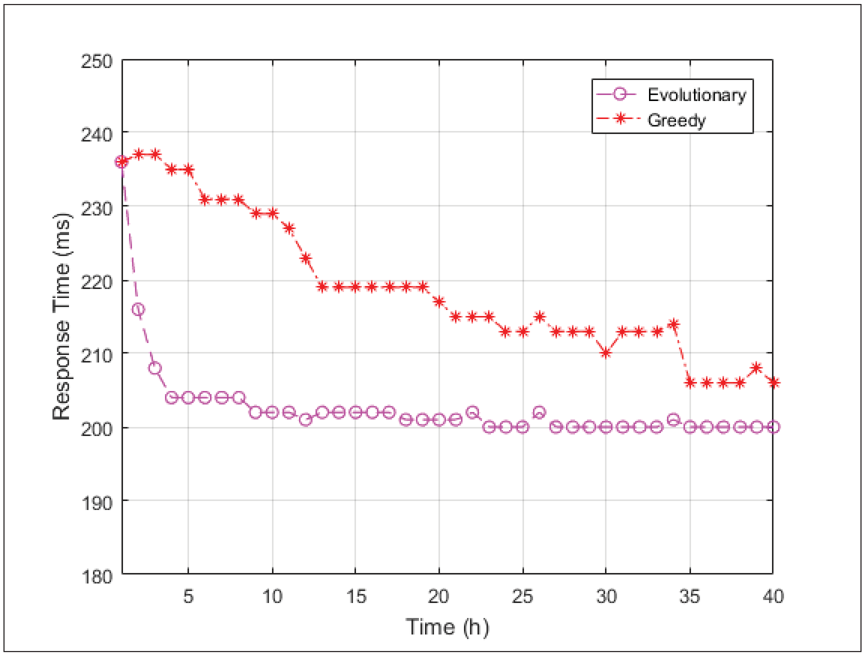


Figure 2.6 Response time of requests

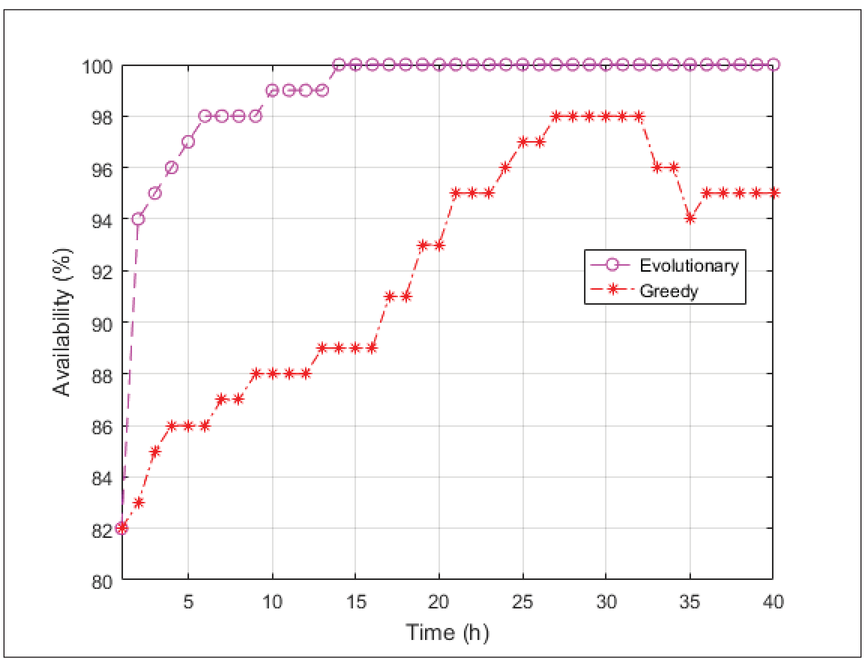


Figure 2.7 Availability of requests

we devised an evolutionary model to stabilize the federations. We modeled the non-cooperative scheme as an evolutionary game and advanced a decentralized model that inherits the settings of the replicator dynamics in order to reach an evolutionary stable strategy. The numerical results show how the formation process converges to a stable state which improves the payoff and QoS in terms of services' availability and response-time.

CHAPTER 3

DYNAMIC FOG FEDERATION SCHEME FOR INTERNET OF VEHICLES

Ahmad Hammoud[1] , Maria Kantardjian [2] , Amir Najjar[2] , Azzam Mourad[2] , Hadi Otrok[3] , Zbigniew Dziong[1] , Nadra Guizani[4]

[1] Ecole de Technologie Superieure (ETS), Montreal, Canada
[2] Department of Computer science and Mathematics, Lebanese American University, Beirut, Lebanon
[3] Department of EECS, Center of Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, UAE
[4] Department of Computer Science and Engineering, University of Texas, Arlington, United States

Paper published in IEEE Transactions on Network and Service Management on October 28, 2022. doi: 10.1109/TNSM.2022.3217972

3.1 Abstract

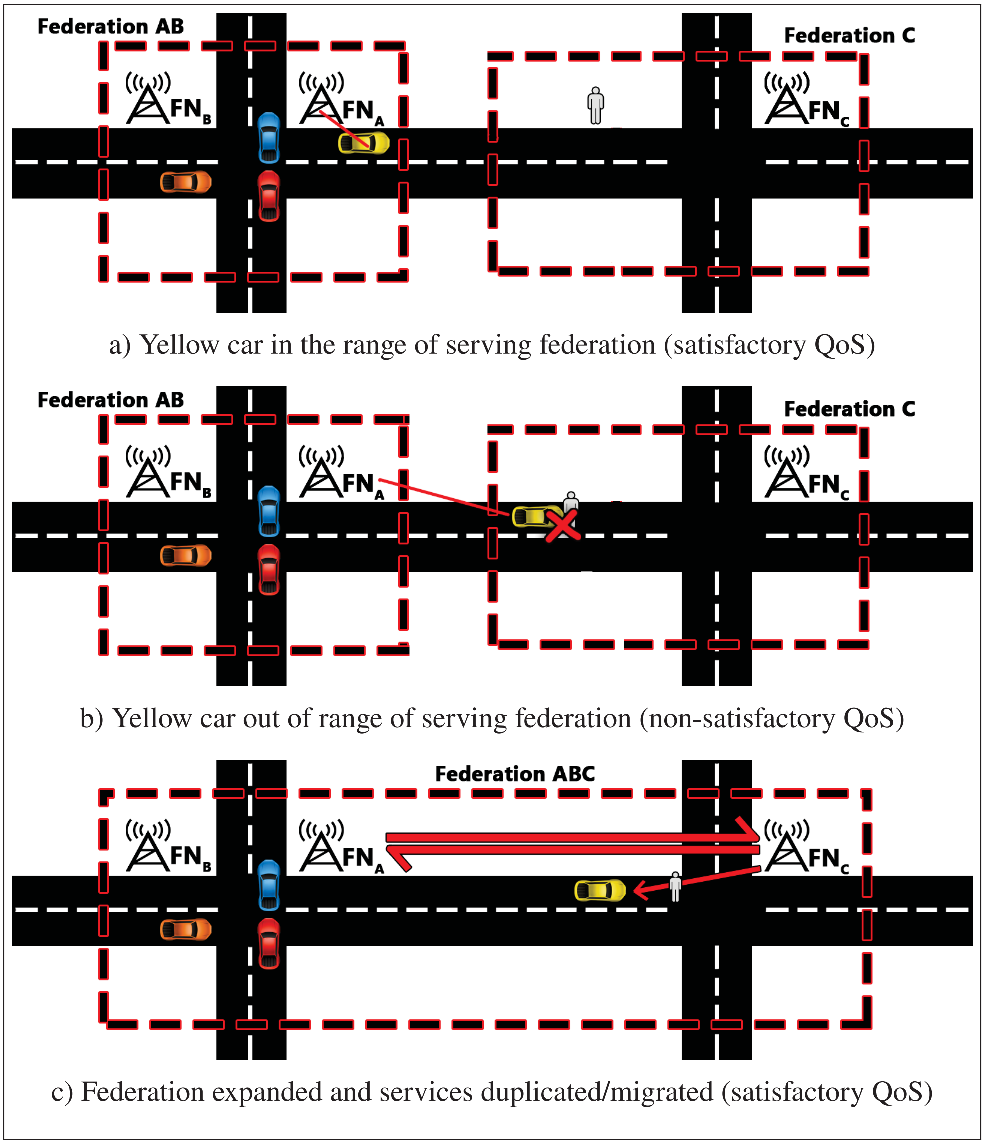
Federated fog computing is an answer for horizontally upscaling fog resources to improve the Quality of Service (QoS) of Internet of Things (IoT) applications. However, the dynamic nature of some IoT's crucial components, such as the ones of Internet of Vehicles (IoV), may hinder the QoS improvement and result in its deterioration instead. Specifically, delays can occur due to the unoptimized distribution of services and unbalanced network traffic loads on the fog nodes. The current federated fog architectures ignore the mobility of users during the formation of fog federations. In this work, we present an adaptive and efficient fog federation formation scheme using game theory according to the service requirements. The problem formulation in terms of forming the federations and offloading requests among fog members is formulated as an integer program, then modeled as a Hedonic game. We adopt the Merge & Split as a formation technique, where the federations that are not satisfied in terms of QoS merge with other federations that would enhance the service performance. Our adaptive fog federation formation mechanism is designed to cope with the environmental changes in the IoV paradigm. Experimental evaluation shows that our framework can acquire better QoS and lower time to form the federations compared to the literature.

3.2 Introduction

The emerging Internet of Things (IoT) devices communicate and exchange data with each other to provide better services and quality of life for the end users (Khabbaz, Assi & Sharafeddine, 2020; Chamra & Harmanani, 2020; Shurrab, Singh, Mizouni & Otrok, 2022). IoT provides many services in various applications such as home automation, health, social life, agriculture, etc (Rahman, Tout, Talhi & Mourad, 2020; Abououf, Singh, Otrok, Mizouni & Damiani, 2021). For instance, in Barcelona, IoT-enabled urban services have dramatically reduced traffic jams and pollution, in addition to reducing the consumption of light, water, and energy (Madakam & Ramachandran, 2015). The large number of data exchange in IoT and the support of real-time IoT applications have motivated researchers to investigate latency-free solutions (Islambouli & Sharafeddine, 2019; Xue *et al.*, 2018). One of the recent efforts is the emergence of *fog computing* (Shih, Chung, Pang, Chiu & Wei, 2017; Sorkhoh, Ebrahimi, Assi, Sharafeddine & Khabbaz, 2020). Fog can be dispatched closer to the user, extending the cloud computing paradigm to the edge of the network (Sami, Mourad, Otrok & Bentahar, 2020; Xue *et al.*, 2019). This results in reducing the communication delays between the users and the servers, leading to an enhanced Quality of Service (QoS) (Al-Fuqaha, Guizani, Mohammadi, Aledhari & Ayyash, 2015; Yang *et al.*, 2022). Internet of Vehicles (IoV) integrates IoT devices into vehicles to support various services such as Intelligent Transportation and Autonomous Driving through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications(Fawaz, 2018; Feng *et al.*, 2020). Nowadays, the concept of IoV is turning into an important developing point of discussion in both research and industrial fields due to its wide set of applications and its ability to provide many benefits such as road safety(Maheswaran, Yang & Memon, 2019; Mourad, Tout, Wahab, Otrok & Dbouk, 2020).

Nevertheless, fog nodes (i.e. fog servers) are costly to deploy profusely(Hammoud, Otrok, Mourad & Dziong, 2021). Additionally, fog providers might not be able to handle unexpected network loads caused by congestion in some geographical areas. This can lead to degraded quality of service. To avoid the cost of purchasing, operating, and managing a large number of fog nodes, the concept of fog federations was founded. Fog federation consists of merging resources from several fog service providers in order to serve the huge stream of demand emitted by the users. Rather than keeping their unallocated resources idle, fog providers in a single federation share their available resources among themselves to increase the service reliability and profit (Hammoud *et al.*, 2020b), (Al-Hilo, Samir, Assi, Sharafeddine & Ebrahimi, 2020). However, due to the mobile nature of the vehicular paradigms, vehicles may move beyond the coverage area of the federation which can deteriorate the service quality of crucial IoV applications. In addition, the stability of the formed federation is one important factor to maintain a stable performance. Otherwise, federation members might break from their federation causing a further reduction of the agreed-upon QoS. Furthermore, the formation of the grand federation, i.e. a big coalition formed by all providers, has been addressed in the literature and proven that it is not optimal in most cases (Guazzone, Anglano & Sereno, 2014; Mashayekhy, Nejad & Grosu, 2014; Hammoud, Otrok, Mourad & Dziong, 2022b).

Our main objective is to develop a comprehensive fog federation architecture, including related models and algorithms, that can overcome the limitations of the existing solutions in terms of adaptivity and stability, resulting in enhanced QoS of the applications and improved profitability of the fog providers under various scenarios. Fig. 3.1 illustrates the need for fog providers to form adaptive federations that prevent QoS from diminishing. In Fig. 3.1a, Fog Providers FN_A and FN_B cooperate as one federation AB in order to serve the incoming requests from the vehicles, meanwhile fog provider FN_C is idle as the vehicles have a better connection to FN_A and FN_B. We assume that the vehicular service requested is deployed by provider A, and can be replicated/migrated to other providers within the same federation. The federation's coverage is represented by the dashed rectangle; the vehicles are served with deteriorated QoS once they move out of the fog range. To show the dynamicity of this environment, we illustrate the yellow vehicle to be moving away from the coverage area of the federation. The red line connecting the vehicle and the fog provider represents the optimal connection for the former to access the desired service. Fig. 3.1b showcases a possible outcome, whereby the QoS is diminished, resulting in increased latency that could be harmful to real-time IoV applications and could lead to accidents. Fig. 3.1c shows the effectiveness of an adaptive fog federation architecture. As the



a) Yellow car in the range of serving federation (satisfactory QoS)

b) Yellow car out of range of serving federation (non-satisfactory QoS)

c) Federation expanded and services duplicated/migrated (satisfactory QoS)

Figure 3.1 Motivation for a Dynamic Fog Federation Formation

yellow vehicle approaches FN_C, the fog provider in federation AB handling its requests offloads them to FN_C, to which the yellow vehicle has a better connection, thus forming a federation ABC that cooperate by sharing resources, managing requests, and offloading tasks among each other to avoid the deterioration of QoS.

To satisfy the requirements of IoV, this work proposes a game-theoretic approach that relies on the Hedonic Coalition Formation mechanism in order to form fog federations that adapt to the mobile nature of the vehicles. It is worth mentioning that game theoretical frameworks have become prevalent in many engineering fields, including communications (Saad, Han, Debbah, Hjorungnes & Basar, 2009; Hammoud *et al.*, 2018). Our devised architecture is fully decentralized and thus does not rely on a central entity to operate. The problem formulation in terms of forming the federations and offloading requests among fog members is formulated as an integer program, then modeled as a Hedonic game. We adopt the Merge & Split as a formation technique, where the federations that are not satisfied in terms of QoS merge with other federations that would enhance the service performance. The dynamic nature of IoV (e.g. QoS diminishing as a result of vehicles moving away from the federation's coverage area) can frequently trigger the execution of the formation technique to adapt to the changes in this paradigm. In our approach, fog providers have full autonomy to split from the federation and find another federation to join in order to provide satisfactory QoS to its users. To evaluate our scheme, we use SUMO (Simulation of Urban Mobility) simulator that generates realistic vehicular trajectory data to serve in our experiments as moving users. Experimental analysis shows that our approach results in a higher user satisfaction rate in terms of QoS, stability architecture, and a lower execution time when compared to other approaches presented in the literature. The main contributions of this work are summarized as follows:

- Devising an adaptive federated fog architecture that considers environmental changes to support IoV.
- Modeling the fog federations formation and offloading problems as a Hedonic game-theoretic model.
- Proposing a decentralized Merge & Split algorithm that maximizes QoS when forming or restructuring the set of fog federations.
- Analyzing the effectiveness of our proposed approach by comparing it with other benchmark models.

The rest of the paper is organized as the following. Section II discusses the related work existing in the literature. In section III, we illustrate the proposed fog federation architecture. In section IV, we formulate the problem as an integer program. Section V demonstrates our dynamic fog federation formation technique. In section VI, we present and analyze the experiments carried out to show the effectiveness of the proposed solution. Finally, we conclude the paper in section VII.

3.3 Related Work

The idea of forming federations has already been investigated in literature in different scenarios. Many researchers have tried to propose federation formation mechanisms in order to make better use of the resources of the participating entities. We address below the recent efforts of cloud-based and fog-based solutions. The cloud-based solutions are included due to their similarities to the fog-based solutions in terms of resource management and task offloading mechanisms.

3.3.1 Cloud-based Solutions

Some examples in the cloud computing paradigm (Guazzone *et al.*, 2014) and cellular networks (Anglano, Guazzone & Sereno, 2014) suggested forming federations using game theoretical approaches for profit maximization. In (Guazzone *et al.*, 2014), Guazzone et al. devised a distributed algorithm that allows Cloud Providers to cooperate if it would result in an increase in individual profit and a reduction in energy costs. While the proposed algorithm always converges to a final partition (i.e., federation formation) in a static and predictable environment, it is a non-preferred solution in a dynamic setting where adapting to environmental changes strongly affects the performance of the partition. Additionally, the QoS was not addressed in their approach. Cloud federations were formed in (Hammoud *et al.*, 2020a) using an evolutionary game theoretical model to reach stability within the formed federations based on the profitability of the cloud providers. Nevertheless, the dynamicity that IoV imposes was not considered as the formation was only tailored to cloud computing. In (Anglano *et al.*, 2014), Anglano et al.

proposed a distributed algorithm that allows Network Operators to cooperate if the resulting federation would increase the individual profit of the entities and meet QoS requirements set by the users. However, similar to (Guazzone *et al.*, 2014), the proposed solution always converges to a final partition, which is inconvenient in our settings. Furthermore, they do not take the QoS into consideration.

3.3.2 Fog-based Solutions

The concept of forming fog federations to support IoT applications is very recent, and the number of published works focusing on enhancing QoS by federating fog providers is limited. In (Sharmin, Malik, Ur Rahman & MD Noor, 2020b), Sharmin et al. suggested a micro-level fog unit deployment in applications that are sensitive to delays, such as in IoV. In the proposed framework, fog federations act as a consortium whereby resources that are underutilized are shared with other fog providers. Additionally, they implemented a price-based workload balancing algorithm that would limit fog units from offloading to other consortium members. In (Veillon, Denninnart & Salehi, 2019a), Veillon et al. proposed an approach that aims at minimizing latency in video streaming applications, specifically for users who are in distant geolocations through fog delivery networks federations (called F-FDN). F-FDN works by pre-processing video streams that are popular in a certain region. Specifically, since FDNs are limited in resources, they suggested pre-processing only the popular parts of the video and the remaining parts to be processed on-demand. Furthermore, to reduce the on-demand processing, FDNs reuse preprocessed video data on the neighboring provider, thus forming a federation. In (Anglano, Canonico, Castagno, Guazzone & Sereno, 2018a), Anglano et Al. have proposed a distributed game-theoretic approach to form coalitions as fog federations where FIPs (Fog Infrastructure Providers) that share the same co-location facility may join/leave a coalition autonomously according to their own preference, without any permission requirements, in order to maximize their individual monetary profit resulting from the formed federations, whereby the FIPs share resources and workloads. Their approach resulted in stable and profitable coalitions. However, the main target was to maximize profitability rather than maximize QoS, which is a critical metric in IoV applications, especially for road safety applications. Additionally, similar to (Guazzone *et al.*, 2014) and (Anglano *et al.*, 2014), the coalitions converge to final partitions, which determine the resulting coalition structure. In (Ennya, Hadi & Abouaomar, 2018), Ennya et al. investigated the distribution and offloading of tasks in fog computing. They modeled the problem using coalition formation in game theory, whereby a user's requests are handled based on its proximity to an idle fog provider or federation. However, as in the other scenarios, the coalitions converge to a final partition and the coalitional structure is determined. The authors of (Hammoud *et al.*, 2021) devised an architecture to reduce the dynamic behavior of the fog providers when deciding on which federation they want to join. They proposed a decentralized algorithm based on the replicator dynamics equation in order to reach a global consensus for the formed federations. Nevertheless, their main concern was to analyze the behavior of the fog providers when forming federations without taking the dynamic users' requirements into consideration. In (Sharaf & El-Ghazawi, 2019), Sharaf and El-Ghazawi suggested a Markov Chain Monte Carlo (MCMC) algorithm for forming coalitions between fog providers. They introduced some constraints on the coalitions, which is that the formed coalitions must be of semi-equal computational powers. The coalitions are also based on the preferences of the fog nodes. In (Shamseddine *et al.*, 2020a) Shamseddine et al. proposed a fog federation formation mechanism using a genetic algorithm approach to form the federations and learning to predict perceived QoS by the users. The proposed approach is managed by a central entity called a Broker who is responsible for forming and maintaining the federations.

Forming vehicular federations was also proposed. Manoochehri and Wenkstern proposed a dynamic coalition formation, where autonomous vehicles form coalitions, and in each coalition, a leader vehicle is elected which manages the coalition and decides whether other member vehicles of the coalition may join/leave the coalition or form a new coalition. Their approach does not have a central broker that has global information, but rather the leader and member vehicles acquire information about the surrounding by V2V (Vehicle to Vehicle) communications (Manoochehri & Wenkstern, 2017).

To the best of our knowledge no research in literature considers dynamic coalition formation to maximize QoS in the context of the Internet of Vehicles while taking into account the mobility of vehicles which causes a change in the environment requiring the formation of different coalitions over time. It is worth to mention that other efforts still exist in the literature to manage resources and enhance communications for vehicular applications such as resource deployment and allocation, code index solutions, etc... (Salahuddin, Al-Fuqaha, Guizani & Cherkaoui, 2014; Tran, Kaddoum & Truong, 2018; Kaddoum, Nijsure & Tran, 2015; Balasubramanian, Otoum, Aloqaily, Al Ridhawi & Jararweh, 2020). However, in this work, we focus on the federation-based efforts.

3.4 Proposed Architecture

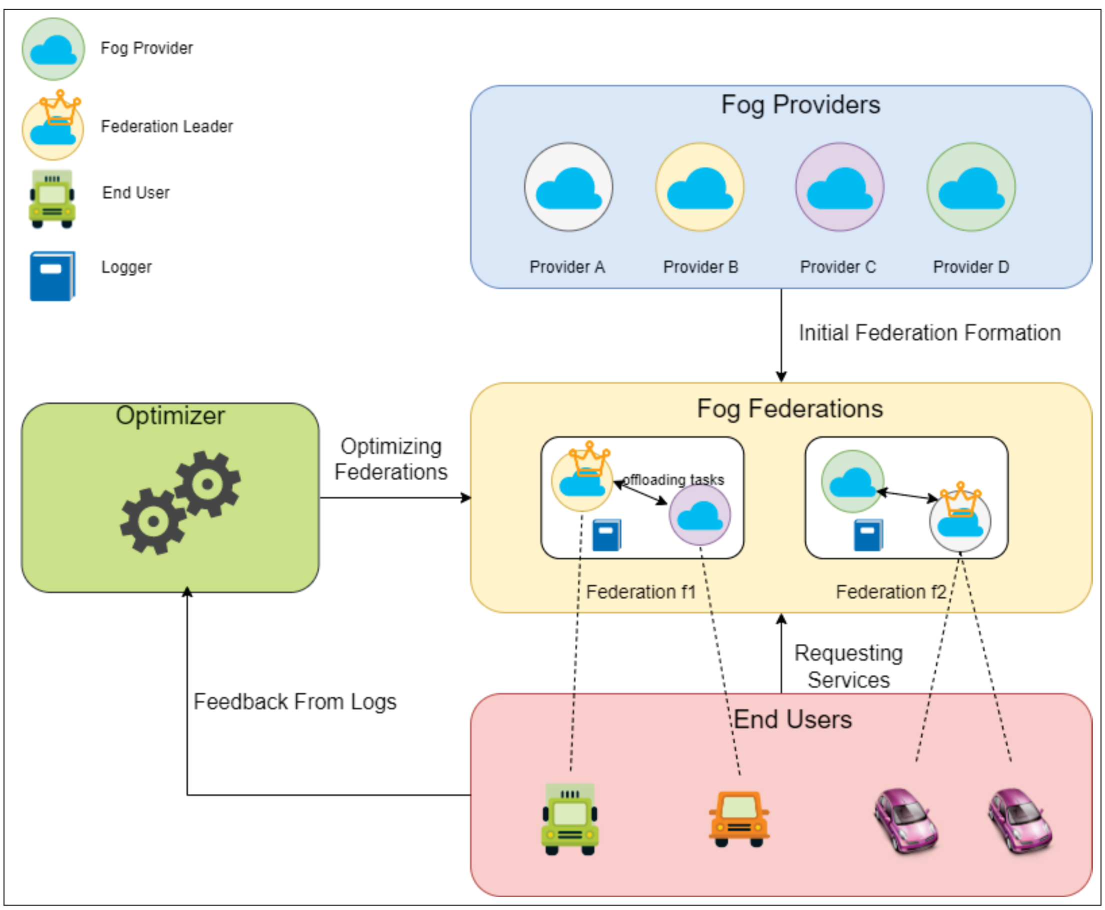


Figure 3.2 Dynamic Federation Formation Architecture

Fig. 3.2 illustrates our proposed architecture. Initially, every fog provider is assigned as a singleton fog federation and declares the services it offers. Afterward, the vehicles, i.e. users, request services from the corresponding fog federation with the expectation to receive responses to their requests with minimal delay to satisfy their needs and ensure road safety. A federation keeps a log of the users' requests, which is used to study the dynamic behavior of the users and modify the fog federations according to the federation optimizer. Maintaining a satisfactory QoS over time may require the optimizer to execute frequently in order to adapt to the changes in IoV. We list below the main components of our proposed architecture

- **Fog Providers:** are the ones who supply the resources to the end-users. They are distributed in certain geographical locations near the end-users. Each fog provider owns at least one fog node and offers a set of services for the end users.

- **Fog Federations:** each federation is a set of fog providers that share resources with other members in order to achieve higher QoS. A fog provider can offload one or more of its services to other providers within its federation. We follow the general assumption in the literature that a fog node can only be part of a single federation at a given time.

- **Federation Leader:** is a fog provider that gets elected to handle the managerial decisions on behalf of the federations, such as whether or not to merge with another federation. The decision is based on the logged requests from the logger component.

- **Logger:** every federation leader has access to a logger that belongs to the same federation. This component collects the data from all of the fog providers of the belonging federation. The data is the service quality of the served requests. The optimization of the logger is out of this work's scoop, and is left for future work (Tello, Gianini, Mizouni & Damiani, 2019).

- **End Users:** the users, i.e. vehicles, are assumed to be in a dynamic environment, whereby they are mobile and their position changes over time. These vehicles request services that are deployed to the fog nodes and expect to receive a response with low latency and high throughput.

- **Optimizer:** also referred to as *Merge & Split-based Fog Federation Formation*. The optimizer allows the federations to adapt to the changes and prevent the QoS from deteriorating. It helps to continuously reinforce the federation structure, according to the needs of the vehicles as they change their parameters. The details of the optimizer are provided in sections IV and V.

Our devised architecture makes use of all of its components by assigning them the designated roles to serve in the optimization process of the computing infrastructure. By allowing the federation leaders to have access to their loggers, significant decisions can be taken to adjust the formation of the federations and the offloading of requests accordingly. If the members are not satisfied with the changes in their current formation -established by the leaders- they are able to split from their federation and/or join a different one. The next section provides technical details on the dynamics of the addressed environment and mathematically formulates the fog federation formation problem.

3.5 Problem Formulation

The research aims to study a dynamic environment that has users' requirements changing over time. The goal is to devise an infrastructure that maintains a satisfactory service for the end-users at any given time. We detail below the modeling of our devised architecture. The system model is comprised of n vehicles and m fog providers. Each fog provider i has its fog nodes deployed at a geographical location characterized by X_i and Y_i coordinates. The set of vehicles and fog providers are defined as $N = \{V_1, V_2, V_3, \ldots, V_n\}$ and $M = \{FN_1, FN_2, FN_3, \ldots, FN_m\}$, respectively. The state of every component is associated with time $t \in T$.

Table 3.1 Summary of Notations

Notation	Description
n	total number of vehicles
m	total number of fog providers
N	set of vehicles
M	set of fog providers
V_i	vehicle i
FN_j	fog provider j
q_k	QoS of request k
x_{lt}	x coordinate of vehicle l
y_{lt}	y coordinate of vehicle l
f_i	fog federation i
C_{f_f}	set of fog providers located in federation f
F	the set of all fog federations
$a_{ij,t}$	connection between vehicle i and fog j at time t
$b_{ijk,t}$	offloading decision of connection of the vehicle i from fog j to fog k at time t
$c_{jf,t}$	the membership of fog j within federation f at time t
$d_{ij,t}$	request vehicle i to fog j is satisfactory

3.5.1 Task Invocation

For every time $t \in T$, a vehicle may request one or more services from various fog providers. The QoS values of these requests at time t are represented by $R_t = \{q_1, q_2, q_3, \ldots, q_n\}$ such that

$$\forall q_i \in R_t, 0 \leq q_i \leq 1 \tag{3.1}$$

where $q_i = 1$ means that the request was processed successfully by the server.

3.5.2 Vehicle Trajectory

The physical location of the service requester (i.e. the vehicle) plays an important role due to the limited wireless coverage of the fog nodes. We define two trajectory sets X and Y as follows.

$$UX = \begin{Bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1t} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2t} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3t} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nt} \end{Bmatrix}$$

$$UY = \begin{Bmatrix} y_{11} & y_{12} & y_{13} & \dots & y_{1t} \\ y_{21} & y_{22} & y_{23} & \dots & y_{2t} \\ y_{31} & y_{32} & y_{33} & \dots & y_{3t} \\ \dots & \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & y_{n3} & \dots & y_{nt} \end{Bmatrix}$$

where x_{it} and y_{it} are the coordinates of vehicle i at time t.

3.5.3 Fog Federations

Given the fog nodes provided by the different fog providers and the services each one deploys, federations are formed based on the combination of providers that maximizes their own QoS. A federation f_i has a set of fog providers $C_{f_f} \subset M$. The set of all fog federations is defined as $F = \{f_1, f_2, f_3, \dots, f_n\}$. We introduce a variable $a_{ij,t}$ to model the connections between the vehicles and the fog provider at time t as per the following:

$$a_{ij,t} = \begin{cases} 1, & \text{if vehicle } i \text{ requesting from } FN_j \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

We also define another variable $b_{ijk,t}$ to model the offloading patterns between the fog providers at time t:

$$b_{ijk,t} = \begin{cases} 1, & \text{if } a_{ij,t} = 1 \text{ and} \\ & \text{service } i \text{ runs on } FN_k \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

meaning that if a vehicle i is requesting a service from FN_j, then $b_{ijk,t} = 1$ when FN_j is offloading this service to FN_k. If FN_j is not offloading its service and is running it locally, then $b_{ijj,t} = 1$. We model next the placement of the fog providers in the federations. We introduce variable $c_{ij,t}$ that specifies whether fog provider i belongs to federation j at time slice t:

$$c_{ij,t} = \begin{cases} 1, & \text{if } FN_i \text{ belongs to } F_j \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Finally, as each deployed service has different QoS requirements, we define the satisfaction threshold for service c, $service\_thresh_c$, to be a value between 0 and 1. If the QoS value of a certain request is greater than or equal to $service\_thresh_c$, the request is considered to be satisfactory. We introduce the variable $d_{ij,t}$:

$$d_{ij,t} = \begin{cases} 1, & \text{if } a_{ij,t} = 1 \text{ and} \\ & QoS_{ij,t} \leq service\_thresh_c \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

where x is a service deployed on FN_j. Hence, we can formulate the problem as an integer program, which is a mathematical optimization program in which the variables are restricted to be integers, for every time t as per the following:

$$\text{Maximize} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij,t}$$

Subject to

$$\sum_{j} c_{ij,t} = 1 \quad \forall i \in [1, m] \tag{3.6}$$

$$\sum_{k=1}^{n} b_{ijk,t} \leq 1 \quad \forall i \in [1, n], j \in [1, m] \tag{3.7}$$

$$b_{ijk,t}.\sum_{v} c_{jv,t}.c_{kv,t} = 1 \quad \forall i \in [1, n], j, k \in [1, m] \tag{3.8}$$

At time slice t:

- Constraint (3.6) ensures that each fog provider belongs to only one federation.
- Constraint (3.7) ensures that a service invocation if it exists, is either run locally or offloaded to only one fog provider.
- Constraint (3.8) ensures that an offloading, if it exists, happens between two fog providers that belong to the same federation.

The problem at hand can be divided into two sub-problems. The first one is the assignment of fog providers to the corresponding federation. The second sub-problem is the assignment of service invocations to be run either locally, i.e. on the server that received the request, or offloaded to a different server within the same federation. Both of these problems can be reduced to the assignment problem which is known to be NP-hard (Krumke & Thielen, 2013). Therefore, we model the formation as a Hedonic game and solve it using the Merge and Split method in a reasonable amount of time, while preserving the preferences of the fog providers over their desired federations. We detail this mechanism in the next section.

3.6 Merge & Split-based Fog Federation Formation

In this section, we first define the preceding terminology related to the game. Then, we describe the approach and algorithm in detail.

3.6.1 Preliminary Terminology

Definition 1 (Dynamic Coalition Formation): In static coalition formation games, the coalition structure is imposed by an external factor and the aim is to study this structure. Whereas in dynamic coalition formation games, the aim is to study how the players are interacting in order to form the coalition, as well as how the coalitions are adapting to external and environmental variations (Saad *et al.*, 2009). In this application, one external factor that may affect the coalition structure is the variation of the service quality from one request to another.

Definition 2 (Hedonic Game): A Hedonic game is a game applicable to both static and dynamic coalition formation games. It allows the formation of coalitions based on the preferences of the players. The players have full autonomy on whether to stay in a certain coalition or leave. Thus, the coalitions are a result of the preferences of the fog providers over all the possible coalitions (Anglano *et al.*, 2018a). A coalition formation game is considered to be Hedonic if it submits to the following conditions: 1) the utility of a player depends only on the other players within the same coalition. 2) The players hold preferences to which coalitions they would like to join, and the coalitions are formed based on these preferences(Hammoud *et al.*, 2022b). Hedonic games have shown to have great potential in wireless and communication networks (Saad *et al.*, 2009). A player's preferences in our problem are the coalitions that allow it to maximize the satisfaction rate (i.e. utility) of its service requests. And the modeled game submits to the aforementioned conditions, thus, it is considered as a Hedonic game.

Definition 3 (Satisfaction Rate): The satisfaction rate of a federation or a fog provider is defined as the rate of service requests that meet their QoS threshold over the total number of service requests from the members of the federation or the fog providers respectively.

Definition 4 (Satisfaction Threshold): The satisfaction threshold α_p of a fog provider p is the minimum satisfaction rate at which a fog provider p can be considered satisfied and does not have an incentive to leave its current federation. This threshold can be set according to the nature of the environment.

Definition 5 (Stability): A coalition is stable when none of its members have an incentive to leave the current coalition in order to achieve a better outcome. In our case, a fog federation is stable when all of its members meet the satisfaction threshold α.

3.6.2 Formation Mechanism

The solution is modeled as a fog coalition formation game (M,v), whereby each fog provider in the set M is referred to as a player in the game, and v is the characteristic function, which is the satisfaction rate of the users obtained when a fog provider in a federation A cooperate as a coalition to maximize QoS.

We rely on the merge and split algorithm to form our federations (Apt & Radzik, 2006). It provides a stable federation structure where none of the fog providers is tempted to change the federation afterwards. Fig. 3.3 summarizes the adaptive coalition formation algorithm, combining all properties mentioned as definitions in the previous subsection. We explain below the two phases, i.e. Merge and Split.

Merge Phase: During the merge phase, each coalition selects a leader who is responsible for taking the final decision to merge with another coalition. This allows the merge between coalitions to be faster, as it is a coalition-to-coalition merge and not a provider-to-coalition merge. The leader can be selected according to its qualifications or through a voting mechanism. In order to study the requirements of the environment, all the members participate in logging the data on the federation logger component. Leaders have access to these logs and they base their decision of merging federations according to what maximizes the QoS. After deciding on merging with another federation, all the members of both federations join together into a single federation, and a leader selection process should take place again.

Coalition $A \subset F$ merges with coalitions $B \subset F$ to form a new coalition $C \subset F$ consisting of all members of A and B:

$$C = \{x : x \in A \cup x \in B\}$$

Split Phase: Following the merge phase, all coalitions enter the split phase. Each coalition undergoes a 'Fog Provider Scan' stage, whereby all fog providers' satisfaction threshold α is checked. If any of the fog providers' satisfaction threshold in the resulting coalition is not met, it leaves the coalition in order to potentially find a better coalition to join in future stages throughout the game. This satisfies the Hedonic property. If all fog providers have a satisfaction threshold above α, then the coalition structure is maintained and no fog provider leaves the coalition. This satisfies the stability property of coalitions. However, before a fog provider actually leaves the coalition, there are a few cases to handle.

1. If a fog provider wants to leave the coalition and it had offloaded tasks to another fog provider, then it leaves the coalition and the offloaded tasks remain being serviced at the other fog provider.
2. If a fog provider wants to leave the coalition and another fog provider had offloaded tasks to it, then it services those requests and then leaves the coalition.

3. If a fog provider wants to leave the coalition and it had neither offloaded tasks to another fog provider nor was handling tasks from another fog provider, then it leaves the coalition.

A fog provider y in coalition $A \subset F$ splits from A. The result is new coalitions $B, C \subset F$ as follows:

$$B = \{x : x \in A | x \neq y\}$$
$$C = \{y\}$$

As illustrated in Fig. 3.3, the game begins with each fog provider initialized to be a federation of its own. Thus, initially $|F| = m$. At time t, all federations in F enter the merge phase, whereby the leader of every coalition gathers information from the members in its coalition about their preferences by relying on the logger component where the meta requests are logged. It then chooses to merge resources with the federation that maximizes the satisfaction rate for the overall coalition. The coalitions are modified according to the merges that occurred during the merge phase and the new set of coalitions is referred to as F'. Once all coalitions finish the merge phase, all coalitions in F' enter the split phase. Since the game is Hedonic, in the split phase, any fog provider that does not meet the satisfaction threshold leaves the current coalition and forms a federation by itself in order to seek a different strategy that benefits it as a rational player. As the current round ends and all the players have made a move, a new round $t = t + 1$ takes place where the federations can once again make decisions about who to merge with and whether or not a provider needs to break from its federation. It is worth mentioning that if a player $p1$ splits from a coalition $c1$, it can still merge with a different coalition $c2$ in the next merging phase after it establishes its own coalition $c3$ in the current phase.

The process repeats as dynamic changes in the environment occur, allowing the coalition formation to adapt to the users' requirements. One aspect to note is that merges happen at the

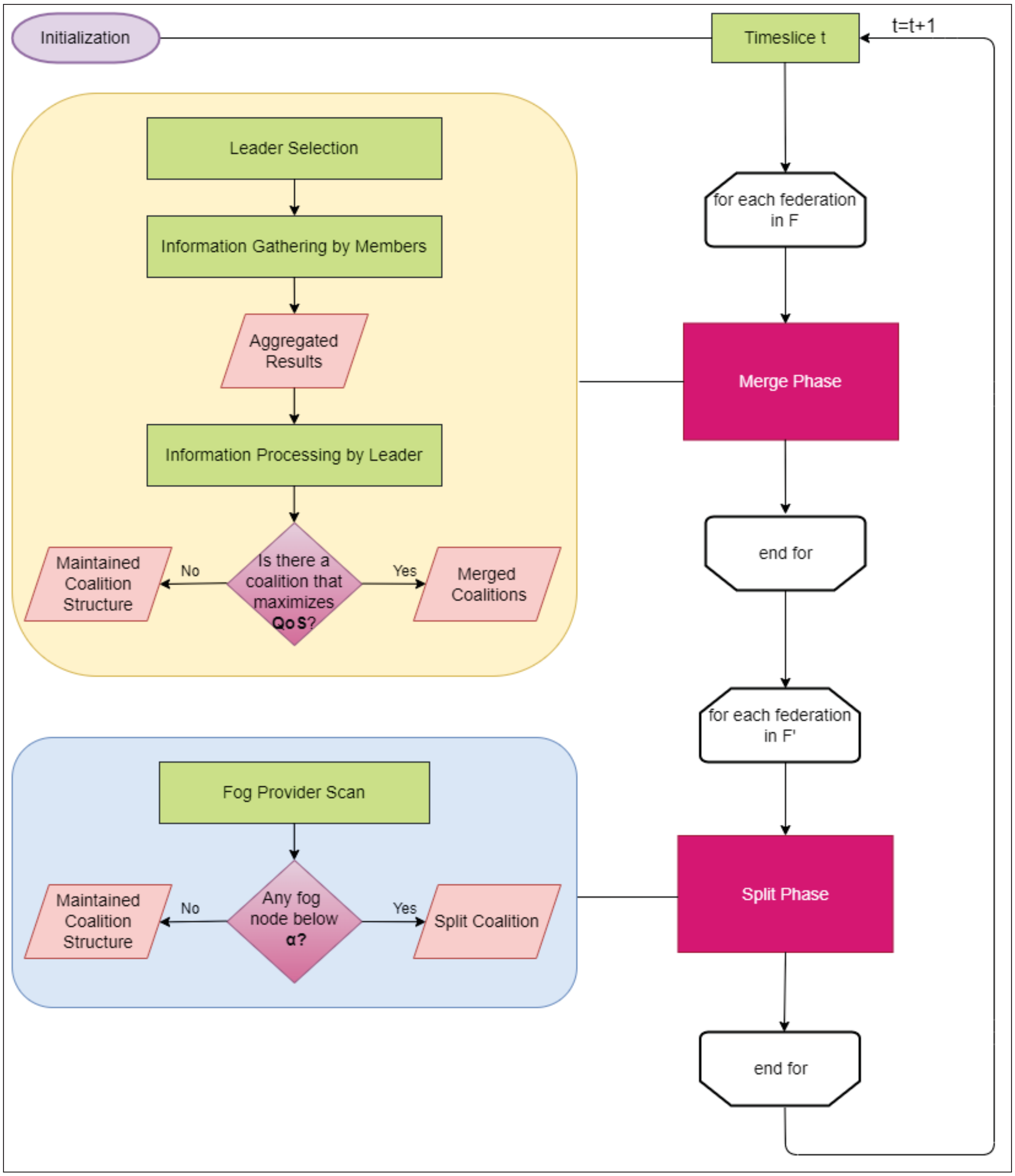


Figure 3.3 Adaptive Coalition Formation Algorithm

coalition level since it is faster than each fog provider leaving its current coalition and joining the new coalition, while splits happen at the provider level. Should any fog provider be unsatisfied with the resulting merged coalition due to the fact that its leader decided to merge with another coalition, it can split in the subsequent split phase.

3.7 Experimental Evaluation

3.7.1 Experimental Setup

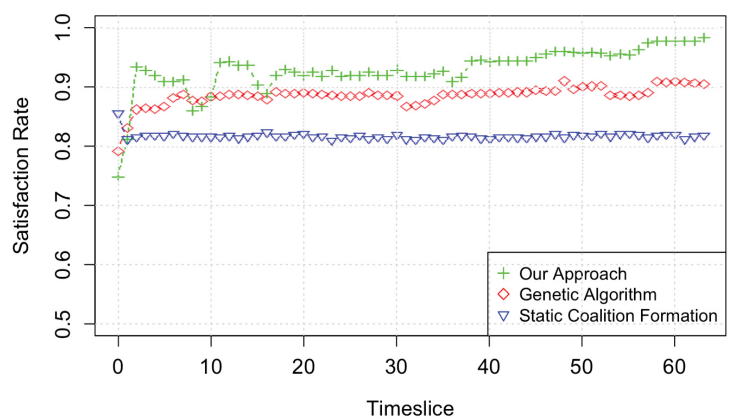
In order to evaluate the effectiveness of our adaptive fog federation architecture, we relied on a vehicular trajectory dataset using SUMO (Simulation for Urban Mobility). The dataset consists of 142 users (vehicles) and 500 fog providers each providing a unique service. The road simulated on SUMO comprises a grid of 12 intersections connected through 17 streets of 500 meters length.We also assign a satisfaction threshold between 85% and 95% for a provider to maintain its spot in the federation, or leave it. $service\_thresh_c$ is the rate of satisfaction that a service requires in order to function normally. The parameters used are specified in Table 3.2. We compare our framework with the Genetic approach presented in (Shamseddine *et al.*, 2020a) and to the Static Coalition Formation in (Anglano *et al.*, 2018a). The metrics we used to compare these approaches are the satisfaction rates of the end-users (i.e vehicles), stability of the formed coalitions, and execution time.

Parameter	**Value**
n	142 users
m	500 fog providers
GA's mutation rate	0.5
GA's crossover	one point crossover
GA's population	16
α_x	[85-95]%
$service\_thresh_c$	[60-100]%

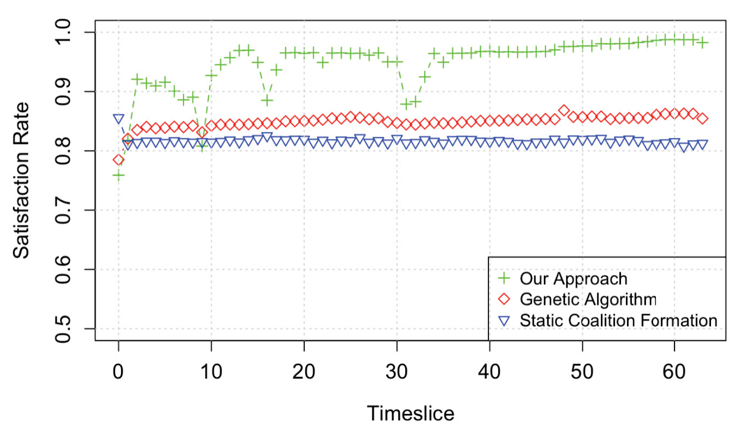
Table 3.2 Parameters used in simulation

3.7.2 Simulation Results

In this subsection, we discuss the experiments performed and the results obtained by implementing the approach presented in this paper. We run the experiments for each algorithm 10 times and average their results with various traffic conditions as a variable.



a) General IoT Services



b) Dedicated IoV Services

Figure 3.4 Satisfaction Rate of the end users

In Fig. 3.4, we study the satisfaction rate of the end-users resulting from the fog federation formations. As previously mentioned, the satisfaction rate is the rate of users receiving a response to their invocations in less time than the threshold with respect to the total number of invocations. This metric is representative of the QoS because users are only satisfied when they receive adequate QoS for their requests for services. Thus, a higher satisfaction rate implies a better QoS and vice versa. We evaluate the performance of the approaches in two different scenarios. The first scenario consists of random IoT requests where the users can either be moving or in a static location (Fig. 3.4a). The second one is where all the users are constantly changing locations (Fig. 3.4b). The first aspect to notice in this experiment is that generally the approach used in this paper resulted in a much better satisfaction rate as compared to the two other approaches in both of the scenarios. Our approach was able to converge to a satisfaction rate of 99%. Another aspect to notice is that in the former part of the curve, the satisfaction rate of users is varying noticeably with time as compared to the other approaches. From t=0 we observe that the satisfaction rate was the lowest. This is mainly due to the fact that as our game starts with each fog provider as its own federation, this causes an overload of incoming requests and the fog provider has no other node to offload some tasks. This leads to high queuing delays and as a result low user satisfaction rate. However, the fog providers then begin to look for offloading opportunities and form federations that would maximize QoS in the coming time slices. The algorithm quickly recovers after the first merge and split phase is finished and better federations are formed in the subsequent time slices, as can be observed at t=1 and on. Throughout the duration of the game, decreases in the satisfaction rate such as the decrease between t=7 and t=9 can be observed. These are due to the fast-changing environment of the vehicles; the federations may quickly become unsatisfactory, however, the algorithm takes this feedback from the vehicles and quickly adjusts the federation and recovers from the sudden drop in satisfaction rate. The satisfaction rate of end-users becomes more stable in the latter part of the curve due to coalition structures becoming more stable and better federations are formed with time. In parallel, the Genetic Algorithm slightly ameliorates the satisfaction rate with time due to its fitness function. On the other hand, the Static Coalition Formation starts with a good rate that is equal to 0.86 but decreases over time and converges to 0.81 because it is unable to recover from the dynamicity of the users. Thus, our proposed mechanism outperforms both approaches previously implemented in the literature in terms of the satisfaction rate of the end-users.

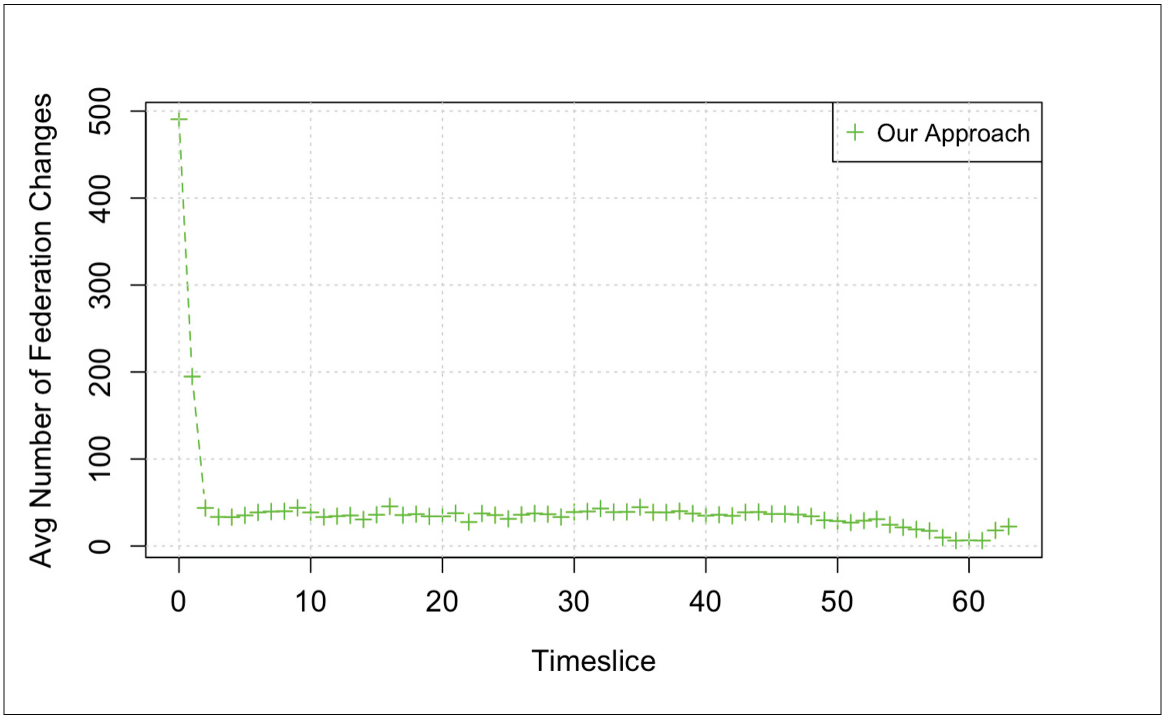


Figure 3.5 Stability of the formed federations

In Fig. 3.5, we study the stability of the formed federations by studying the average number of federation changes for all fog providers, i.e. the average number of times fog providers in a certain time slice change their current federation, either through a merge or a split. As defined earlier, stability is the situation where fog providers who meet the satisfaction threshold do not have the incentive to leave the federation they are in for the purpose of finding a better one, i.e. a federation that would increase its satisfaction rate further. At the beginning of the game, since each fog node is a federation of its own and the federations are constantly changing through merge & split in order to find offloading opportunities that would increase the satisfaction rate above the threshold, the average number of federation changes is very high. However, with time the fog nodes generally have a satisfaction rate above the threshold for their invocations, hence not leaving the coalition, resulting in the coalition structure being more stable. For instance, at t=15, Fig. 3.4 shows that the satisfaction rate has dropped from the previous time slice. In parallel, at t=15, Fig. 3.5 shows a spike in the average number of federation changes. This observation implies that at t=15, when the satisfaction rate has started to drop, the fog federation structure has started to change through merge and split, in order to recover from the drop in satisfaction rate. Another example is at t=60, where the satisfaction rate is very high and in turn, the average number of federation changes has decreased. However, due to the rationality of the players, one strategic change might not be convenient to compensate on the lack of resources without a collaborative decision making mechanism among providers. We notice that at times t=9, t=16, and t=31, the algorithm took a few rounds to increase the QoS after it had dropped. We do not compare the average federation changes to the approach in (Shamseddine *et al.*, 2020a) using the Genetic Algorithm because the fog federation formation was carried out by a central entity, called a Broker, and fog nodes do not have the autonomy to leave a federation. As for the approach in (Anglano *et al.*, 2018a) using Static Coalition Formation, federations do not change over time, the formed federations are static.

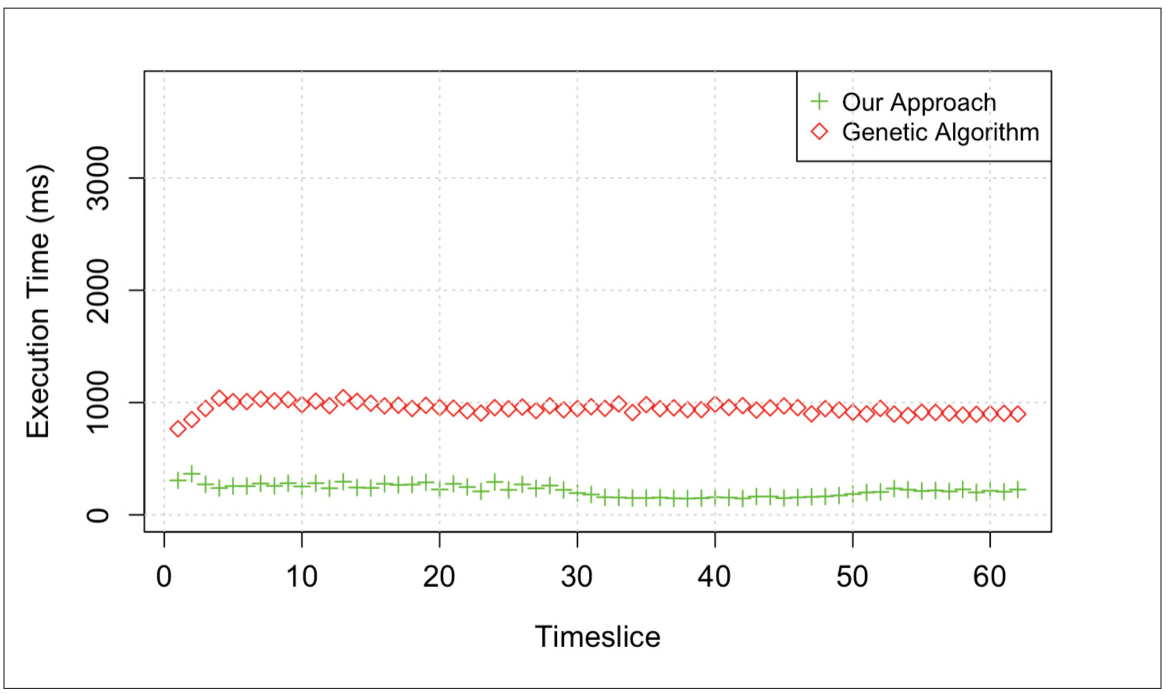


Figure 3.6 Execution time of forming the federations

In Fig. 3.6, we study the execution time of our proposed mechanism. This metric is very crucial in the setting of IoV and autonomous driving because they are delay-sensitive and a small fraction of delay can cause casualties. As seen in the figure, our approach forms the fog federations in less than half the time taken by the Genetic Algorithm in (Shamseddine *et al.*, 2020a). It is worthy

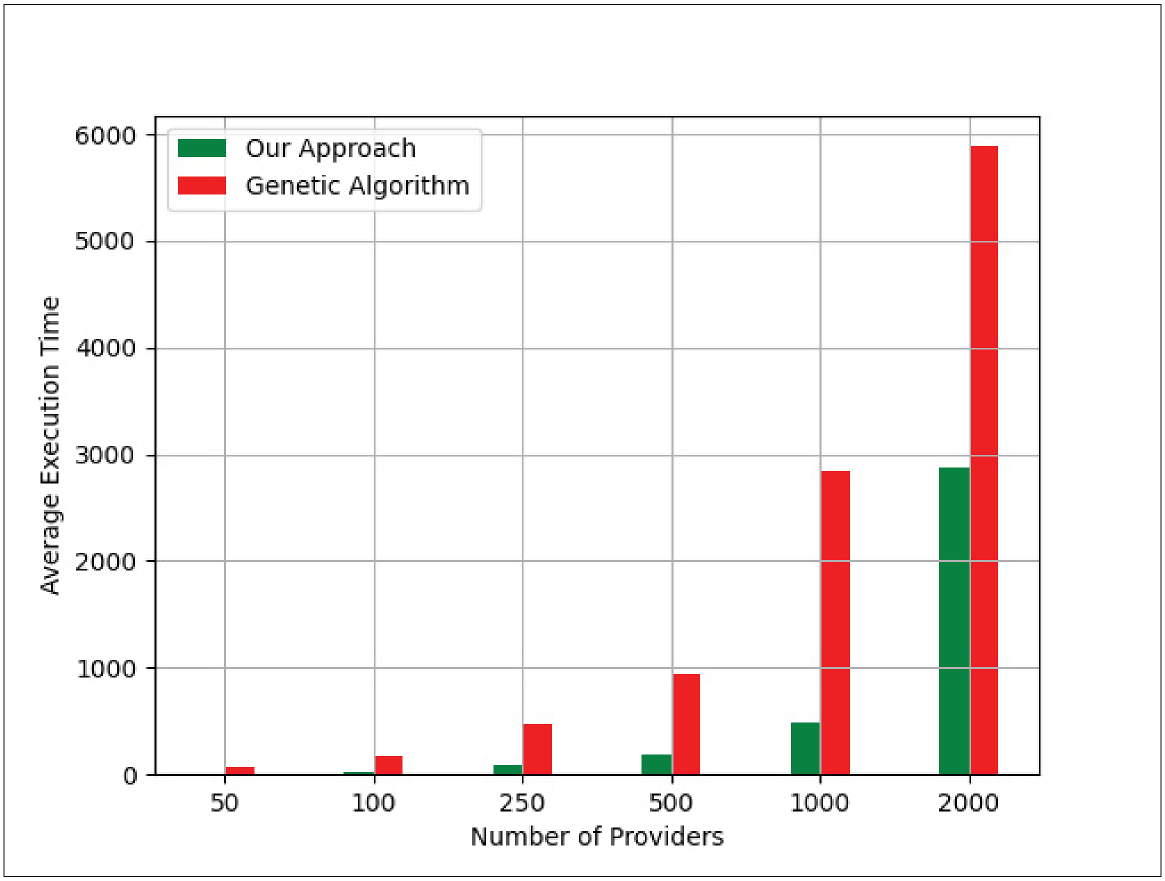


Figure 3.7 Average execution time of forming the federations versus the number of providers

to note that this experiment was run on one device and not multiple devices as a decentralized approach proposes. Thus, we expect the execution time of our approach to be even lower when running in a decentralized manner. The algorithm has a low execution time because unlike the Genetic Algorithm in (Shamseddine *et al.*, 2020a), the system is not controlled by a central entity or broker. Additionally, all members in a federation gather information about possible offloading opportunities, leading to a decrease in the time needed to search for a federation to merge with. In addition, to evaluate the scalability of our scheme, we measure the average execution time when having a different number of fog providers in Fig. 3.7 and compare it with the Genetic work. As observed in Fig. 3.7, we were able to achieve a relatively low execution time especially when the number of providers is low. The Genetic approach on the other hand suffers from having to explore a huge number of solutions in the search space in order to obtain the sub-optimal formation, thus taking a longer time than our approach to obtain a valid solution. We do not compare the execution time to the approach in (Anglano *et al.*, 2018a) using Static Coalition Formation since the federations do not change over time, the formed federations are static. We conclude from these figures that our scheme is scalable and it can achieve greater results than other approaches in the literature.

3.8 Conclusion

Fog computing enhances the intensive computation needed by autonomous vehicles. However, fog providers occasionally get overloaded with requests, resulting in queues and delayed responses. As a solution, forming fog federations was suggested by researchers to enhance the service quality and lower the costs. Nevertheless, this gives rise to another problem; fog federations degrade in terms of QoS when they do not adapt to changes caused by the mobility of vehicles in the IoV paradigm. This paper presents an adaptive fog federation formation mechanism that enhances QoS by forming mobility-aware federations adapting to the environmental changes of the vehicles and responding to location changes in real-time. Specifically, we rely on the Merge & Split method in Dynamic Coalition Formation in Game Theory. On one hand, our proposed scheme leads to intelligent usage of the available resources since idle fog providers do not waste their resources but rather tasks are offloaded to them, and on the other hand it ensures fog providers aren't overloaded in congested areas. Through the presented results, we showed the effectiveness of the proposed mechanism through having a higher user satisfaction rate, obtaining stable coalitions, and low execution time. Future enhancements to this research include integrating Machine Learning to predict changes in the environment in order to respond to these changes ahead of time and introducing Blockchain for added security and preventing malicious attacks.

CHAPTER 4

ON DEMAND FOG FEDERATIONS FOR HORIZONTAL FEDERATED LEARNING IN IOV

Ahmad Hammoud[1] , Hadi Otrok[2] , Azzam Mourad[3] , Zbigniew Dziong[1]

[1] Ecole de Technologie Superieure (ETS), Montreal, Canada
[2] Department of EECS, Khalifa University, Abu Dhabi, United Arab Emirates
[3] Department of Computer science and Mathematics, Lebanese American University, Beirut, Lebanon

Paper published in IEEE Transactions on Network and Service Management on May 3, 2022.
doi: 10.1109/TNSM.2022.3172370

4.1 Abstract

Federated learning using fog computing can suffer from the dynamic behavior of some of the participants in its training process, especially in Internet-of-Vehicles where vehicles are the targeted participants. For instance, the fog might not be able to cope with the vehicles' demands in some areas due to resource shortages when the vehicles gather for events, or due to traffic congestion. Moreover, the vehicles are exposed to unintentionally leaving the fog coverage area which can result in the task being dropped as the communications between the server and the vehicles weaken. The aforementioned limitations can affect the federated learning model accuracy for critical applications, such as autonomous driving, where the model inference could influence road safety. Recent works in the literature have addressed some of these problems through active sampling techniques, however, they suffer from many complications in terms of stability, scalability, and efficiency of managing the available resources. To address these limitations, we propose a horizontal-based federated learning architecture, empowered by fog federations, devised for the mobile environment. In our architecture, fog computing providers form stable fog federations using a Hedonic game-theoretical model to expand their geographical footprints. Hence, providers belonging to the same federations can migrate services upon demand in order to cope with the federated learning requirements in an adaptive fashion. We conduct the experiments using a road traffic signs dataset modeled with intermodal traffic

systems. The simulation results show that the proposed model can achieve better accuracy and quality of service than other models presented in the literature.

4.2 Introduction

Internet of Things (IoT) allows data collection from the sensors integrated within Things that, once analyzed, can be used to efficiently manage the related resources. Correspondingly, the Internet of Vehicles (IoV) is an important part of IoT evolution (Mourad *et al.*, 2020). It allows data exchange among vehicles and infrastructures by using heterogeneous networks that are based on Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Such networks assist the drivers and other engaged parties, enabling road safety and leading to what is called the Intelligent Transportation System (ITS) that is based on intelligent vehicles. The intelligent vehicles gather data (1) intercepted by the sensors embedded within and (2) collected from their surroundings via the networks. Then, they analyze the data to generate useful information that can be used to inform and alert the drivers about issues to be considered during the journey.

4.2.1 Machine Learning in IoV

Artificial Intelligence (AI) has become a key component of the IoV paradigm that allows to develop complex services (Hammoud *et al.*, 2020c). For instance, Autonomous Driving systems, such as Alphabet's Waymo[1] and Tesla's FSD[2], are examples of the widely studied IoV applications that are critical for improving road safety in the future. The safety of intelligent vehicles' trips strongly relies on how well-trained the AI systems are. Particularly, the vehicle scans its surroundings using various sets of sensors, including cameras, beams of radar, lidar, ultrasound, GPS navigation, etc... Then, it passes the sensed data to the AI system which, in turn, analyzes it and makes the best decision under given circumstances (e.g., speed up, stop, turn left, etc...). In order for such a complex model to be ready for deployment, a huge data is required,

---

[1] Alphabet, the parent company of Google. https://waymo.com/waymo-driver/

[2] Tesla's Full Self-Driving system. https://www.tesla.com/en_CA/support/full-self-driving-computer

and a machine-learning procedure must be carried out to discover statistically significant patterns in such data. The data may consist of an enormous number of vehicular trips, which are stored and broken down into sets of intercepted sensors' values and their corresponding interactions made by the driver. Training such a huge stream of data requires capable computing servers such as the ones deployed in Cloud Computing. By having the cloud collecting the data from a significant number of vehicles, it is able to perform intensive computations to extract useful patterns and combine these patterns into one single optimized machine learning model that could be forwarded to the vehicles for deployment.

4.2.2 Federated Learning

Transferring data from the users to the cloud has raised many privacy concerns as it may result in the exposure of their private data either by a session hijacker or by the service provider itself. To address this problem, Google developed federated learning, a privacy-preserving machine learning architecture that protects personal data from being exposed to other parties (Dhole *et al.*, 2016). Such a mechanism consists of having the users train the model independently, and then forwarding only the trained models to the cloud, without the data, to be aggregated together to form one unified machine learning model. Federated learning has proved its efficiency in various fields such as IoT (AbdulRahman, Tout, Mourad & Talhi, 2020), IoV (Pokhrel & Choi, 2020), Healthcare (Xu *et al.*, 2021), keyboard word prediction (Konečnỳ *et al.*, 2016), and many others.

4.2.3 Fog Federations

The more complex the AI application gets, the more computing and storage resources it requires from both parties, i.e. the trainers and IT infrastructure. The output model that needs to be transferred to the cloud can have a size at hundreds of megabytes level (Xia, Ye, Tao, Wu & Li, 2021). Such a huge size would drain the cloud infrastructure and increase costs and delays for the participants and the servers. Scholars tackled this particular limitation by proposing the integration of fog computing in order to assist the process of federated learning within

IoT environments (Stojmenovic & Wen, 2014). Notably, fog computing possesses one critical advantage over cloud computing: the low latency factor. Fog servers are located near the end-users making the communication delay negligible between both parties. Nonetheless, covering a wide area or multiple areas with satisfactory service quality would require deploying many fog servers to avoid QoS deterioration. Due to the fact that IoV is a very dynamic environment, fog providers can not guarantee optimal performance because of their limited coverage area. In contrast, utilizing the fog computing infrastructure to its full potential through federating providers and enabling services migration between them on demand could lead to an efficient and low-cost solution. Fog federations consist of multiple fog providers allocating parts of their resources to be shared in order to handle tasks that cannot be executed otherwise while maintaining an adequate quality of service. Therefore, fog federations could enable the execution of the federated learning tasks even when unpredicted road conditions occur. We illustrate a basic scenario for fog federation formation in Fig. 4.1 where provider *A* has a service of type *s1* (represented by the green circle) deployed on its fog node in *area 1*, and provider *B* has the service *s2* deployed in *area 2*. Users located in *area 1* are requesting service *s1* with a good QoS. Nevertheless, users in *area 2* that need to request the same service, i.e., *s1*, are being served with a degraded QoS due to network limitations. For this reason, providers *A* and *B* can federate according to a service level agreement, and then replicate service *s1* in *area 2* to enhance the QoS of the users in that area.

4.2.4 Problem Statement and Objectives

The concept of federated fog computing and the federation formation came to light in the very recent past (Shamseddine *et al.*, 2020b). With the advent of IoT technology and the drastic increase in the number of IoT devices, scholars sought the opportunity to enhance the computing infrastructure and amplify the Quality-of-Service as a further step aiming toward smart cities. Nevertheless, all of the fog federation-based efforts ignore the fast and large area displacements of the end-users in IoV settings that can lead to a degraded service quality due to the change of gateways, making these federations vulnerable to instabilities. For instance, in such a mobile

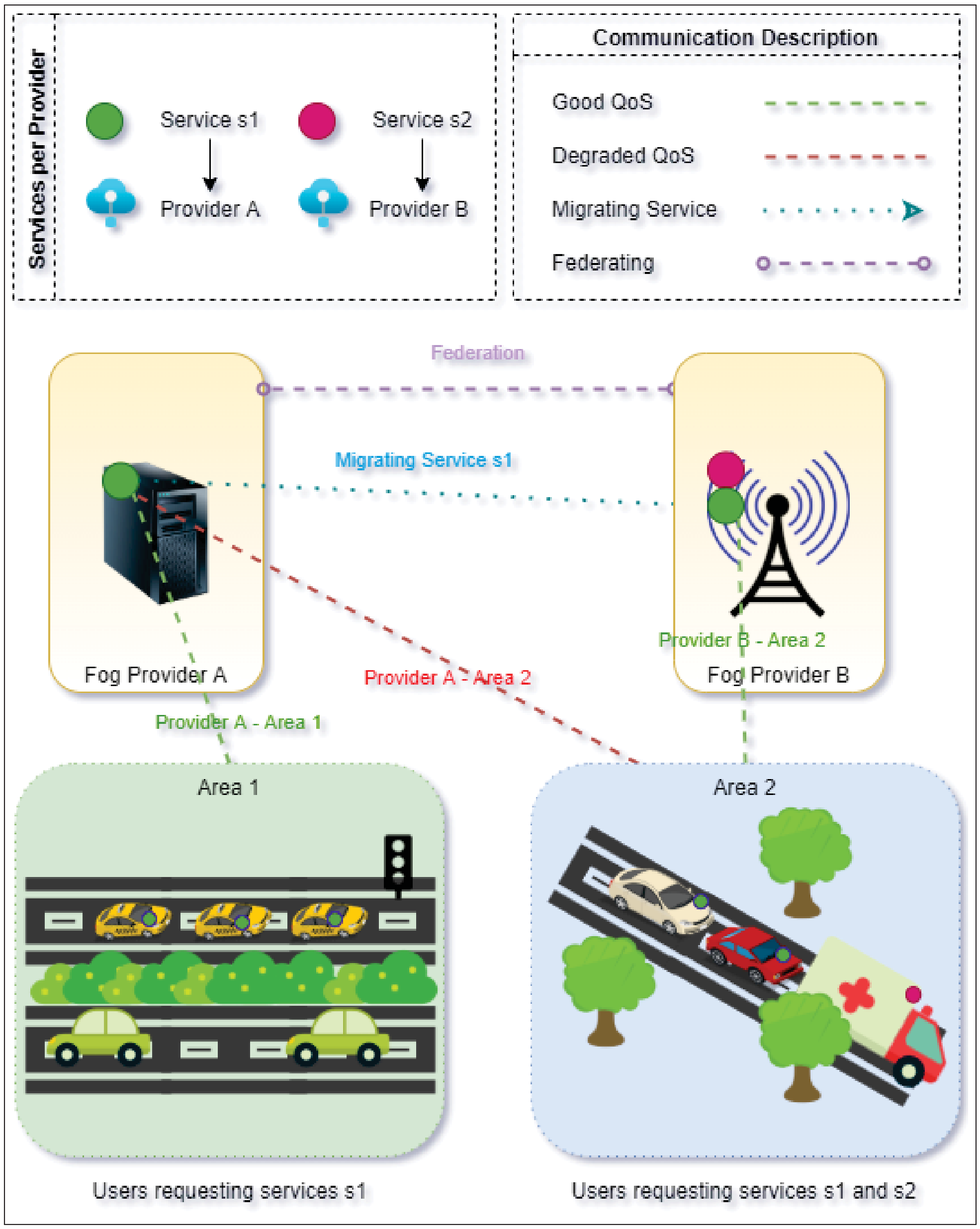


Figure 4.1 Migrating services on-demand among federation members

environment, a fog service provider might decide to break from the federation once it may appear more convenient to join another federation if it is more profitable, or to work independently. This causes the rest of the providers within the abandoned federation to have more load of tasks to process. Furthermore, federated learning is still suffering from many infrastructural complications due to its special requirements that are different from the ones of other IoT applications. Thus, there is a need for studying the fog federation, in terms of its architecture and formation to ensure adequate service quality and a suitable environment for Autonomous Driving applications.

4.2.5 Contributions

This work aims to enable scalable federated learning in the highly dynamic IoV environment. We propose a horizontal federated learning architecture for IoV applications empowered by fog federations. We rely on a Hedonic-game theoretical model for reinforcing the fog federations, i.e. the IT infrastructure, to maintain adequate service quality through migrating services among federation nodes according to the federated learning needs. It is worth mentioning that Hedonic games were used to form cloud and fog federations ((Ray, Saha & Roy, 2018; Anglano *et al.*, 2018b)) as these games perform well in a strategic environment where entities compete and cooperate based on their preferences. Nevertheless, in our proposal, we consider metrics tailored to IoV settings which makes the previous formations inapplicable, due to the dynamic behavior of the participants. Hence, we demonstrate how to adapt the formation to the dynamic federated learning settings. In contrast to the resource-based solutions in the literature, we consider multiple learning applications simultaneously to fully utilize the infrastructure. Our proposed architecture ensures the engagement of more participants in the federated learning process than other approaches proposed in the literature. We evaluate our approach by simulating a process for training a level-1 federated autonomous driving application[3] that can identify traffic signs on the road and alert the driver accordingly. It is worth mentioning that the levels of driving automation are ranged from level 0, where there is no automation, to level 6, where the steering wheel inside the vehicle is optional (Herrmann, Brenner & Stadler, 2018). The dataset is publicly available on Kaggle[4]. We also compare our approach with other approaches mentioned in the literature. Experimental evaluation reveals that our mechanism can achieve better model accuracy, lower model loss and response time, and handle more participants in the

---

[3] Level 1 autonomous driving indicates that the system and the driver have shared control of the vehicle.

[4] https://www.kaggle.com/valentynsichkar/traffic-signs-preprocessed

training process when compared with other approaches. Our contributions can be summarized as follows:

- Devising a dynamic Horizontal Federated Learning architecture for IoV empowered by fog federations that can cope well with the dynamic IoV environment. To the best of our knowledge, we are the first to integrate the fog federation layer to assist in executing the federated learning procedure.
- Adopting a Hedonic game-theoretical model for establishing stability within the federations of fog providers.
- Distributing fairly the workload among the fog providers in the same federation to reduce costs and execution time, and to allow for multiple learning applications to run simultaneously.
- Evaluating the performance of the proposed approach by training driving assistant models and comparing them with models trained by other architectures mentioned in the literature.

Outline of the paper The rest of the paper is organized as follows. In Section 4.3, we discuss the literature and compare the existing solutions relevant to the proposed approach. In Section 4.4, we demonstrate our horizontal federated learning architecture. We formulate and solve the Hedonic game-theoretical model in Section 4.5. Afterward, in Section 4.6, we discuss the results obtained by utilizing our proposed architecture for training a traffic sign recognition model against other approaches. Finally, a conclusion and a list of some aspects of our architecture that can be further studied and enhanced in the future is given in Section 4.7.

4.3 Related Work

In this section, we overview and discuss the related literature efforts that apply to the federated learning concept and the infrastructure reinforcement solutions for enhancing the service quality.

4.3.1 Enabling Federated Learning

Traditional machine learning architectures rely on central entities to receive and process the data collected from users. Such architectures have raised many privacy concerns in terms of exposing private and sensitive data to the party collecting the data from one side, and to possibly man-in-the-middle attacks from another side, where external parties may eavesdrop on the communication between the user and the server. Federated learning has emerged to solve the aforementioned problem as described in (Dhole *et al.*, 2016). Their proposed architecture relies on having a central server and several participants who want to engage in the training process. In federated learning, a participant is an entity that has data but does not want to share it with an external party, thus, it trains the data itself. Before the training begins, the server sends an initial model to all participants to retrain the model according to their own data. After the participants finish local training, they forward the models to the server where they are aggregated and unified into one global model. A communication round comprises the aforementioned actions, i.e. initial model, local training, and aggregation. Depending on the learning application, the federated learning might require several communication rounds to converge, where the consecutive round starts from the aggregated model obtained in the previous round.

There exist 3 categories of federated learning in the literature: (1) Horizontal Federated Learning, (2) Vertical Federated Learning, and (3) Federated Transfer Learning. In a horizontal-based federated learning scheme, the feature spaces are similar for all datasets whereas their samples are different. An example of this category could be having 2 hospitals with a set of different patients for each one, thus different records. But they share similar features for these records. The second category, i.e. vertical federated learning, has the same set of users in all datasets, but they differ in terms of features. The third category is when the majority of both the feature spaces and the samples are different but they have minor subsets of features and samples that are overlapping.

Most of the researchers were occupied with studying the effect of federated learning on other applications, such as Healthcare and malware detection-based applications (Xu *et al.*, 2021; Poirot *et al.*, 2019), ignoring the actual burden that federated learning may provoke in terms of required resources due to the large number of users integrating with the process. In parallel, some have noticed the need for reinforcing the framework to further improve it and make it more reliable. We group some of these recent efforts and categorize them below.

4.3.1.1 Preserving Privacy Approaches

The federated learning concept preserves the privacy of the participants in terms of not revealing their private data to the central server as each participant trains its data locally. Some scholars took further actions in securing the architecture to not disclose the exchanged parameters. For example in (Truex *et al.*, 2019), the authors devised a mechanism that relies on secure multiparty computation (SMC) and differential privacy for reducing the growth of noise injection while preserving privacy for the participants. Their approach was able to protect against inference threats. The authors of (Hao *et al.*, 2019) introduced an improved version of BGV homomorphic encryption scheme to defend the architecture counter privacy leakage of the uploaded models during aggregation and collusion attacks. In (Xu, Baracaldo, Zhou, Anwar & Ludwig, 2019), Xu et al. devised an efficient approach, called *HybridAlpha*, based on SMC to secure the parameters and prevent reverse-engineering them. Their model is resilient to participants dropping out and can reduce the learning time against other SMC-based solutions. In (Qu *et al.*, 2020), Qu et al. devised a Blockchain-based approach integrated within the federated learning framework for enabling decentralized privacy and preventing a single point of failure.

4.3.1.2 Active Sampling

In the typical federated learning settings, a subset of the users gets selected to participate in the training process in a random manner. Some scholars were interested in optimizing the selection process for such participants. For instance, the authors of (Nishio & Yonetani, 2019) proposed a selection procedure according to certain conditions for the sake of aggregating as many updates as possible within a predefined time window. In a similar fashion, the authors of (AbdulRahman *et al.*, 2020) proposed a multi-criteria-based approach for client selection, where they considered their resources and availability. Their framework improves the accuracy in a shorter number of communication rounds compared to others due to benefiting the most from the available participants. In (Huang *et al.*, 2020), a fair historical-aware selection mechanism was proposed that takes into consideration low-priority clients as well in order to guarantee data diversity. Active sampling in IoV was also recently addressed in the literature. For instance, the authors in (Liu, Yu, Deng & Wan, 2021) proposed a framework to reduce the costs and overheads of communication between the vehicles and infrastructure by introducing a flexible aggregation policy to constrain the upload time of the models and eliminate stragglers. In (Saputra *et al.*, 2021), the authors proposed a dynamic FL-based economic framework for the IoV network. They devised a dynamic selection method for the vehicles that should integrate with the learning process while taking both their location and quality-of-information into account. In (Lim *et al.*, 2020), the authors devised an incentive mechanism based on game theoretical models between workers and model owners, as well as among model owners in order to form the federations of trainers in IoV networks.

4.3.1.3 Resource-Aware Solutions

Aside from the works considering the resources of participants during the selection process, some scholars sought to ameliorate the supporting infrastructure in terms of utilizing its resources in an optimized manner by reducing costs, enhancing the communication, and introducing new techniques to further support the federated learning architecture. For instance, the authors in (Wainakh, Guinea, Grube & Mühlhäuser, 2020) explored the usage of a horizontal federated learning approach to reduce the centralization of power and control of the central authority, in addition to creating the possibility of employing a trust mechanism among users to reduce the threats. Moreover, the authors of (Caldas, Konečny, McMahan & Talwalkar, 2018) introduced the usage of lossy compression on the global model forwarded and a dropout algorithm to help the participants train a sub-model instead of a whole. They were able to reduce the size of the forwarded models and lower the local computations while maintaining a decent accuracy but at the cost of an increased number of rounds.

The integration of a fog computing tier within the federated learning architecture was studied as well in the literature. For instance, the authors of (Yao & Ansari, 2020) proposed a balancing technique for the trade-off between the wireless data transmission latency and Internet of Things energy consumption in a fog-aided IoT network where the models will be aggregated locally on the fog nodes. In (Zhao, Feng, Yang & Luo, 2020), the authors suggested adopting a hierarchical federated learning approach where the fog nodes apply local aggregation to the collected models and then forwarded to the cloud for global aggregation. The authors of (Saha, Misra & Deb, 2020) have also emphasized embedding fog servers within the hierarchy for local aggregation purposes due to its benefits of reducing the latency and consumed energy. They further devised a greedy algorithm to select the global aggregator fog node.

4.3.2 Quality-based Solutions

Cloud, Fog, and Edge computing were intensively used throughout the literature to support the federated learning procedure (Section 4.3.1.3). In this part, we review some of the recent literature efforts for enhancing the service quality through a variety of solutions such as service deployment strategies and cloud and fog federations.

4.3.2.1 Service Deployment Strategies

In (Wu *et al.*, 2019), Wu et al. proposed an optimal Cloud-Edge service deployment scheme in IoV based on the users' preferences. Their method consists of building preferences for the users to choose the deployment strategy of their services. Then, they use Genetic Algorithms to maximize their utility for satisfying the QoS requirements. The authors of (Li *et al.*, 2018) have addressed the problem of deploying fog servers into the fog computing infrastructure. Their proposal consists of designing a clustering policy for dynamic mobile cloudlets by using the latter as a supplement for the fog server for offloading purposes. Sami et al. studied the problem of allocating a set of docker containers to a set of volunteering devices to provide services on the fly. Their objective was to provide efficiently enough resources for real-time IoT applications requiring computation processing (Sami & Mourad, 2020). They relied on a Multi-Objective Memetic algorithm for solving their problem.

4.3.2.2 Cloud and Fog Federations

Cloud and Fog federations can offer more reliability and availability for the federated learning architecture in terms of resources. The authors of (Hammoud *et al.*, 2020a) advanced an approach based on genetic and evolutionary models to reach a cloud federations formation that is stable and highly profitable. In (Goiri *et al.*, 2010), the authors addressed the matter of increasing the revenue of cloud providers through a mechanism that determines the optimal decision on where and when to allocate their computing resources. In (Dhole *et al.*, 2016), the authors tackled the federation formation problem using trust as a measurement among providers. They claimed to reach stability, profit maximization, and fairness through their formation mechanism. In (Anglano *et al.*, 2018b), the authors addressed federating fog providers through a Hedonic game model in order to improve the payoff of the fog providers. The formation mechanism took place according to the resources and the profit of the players for the sake of better serving IoT devices. In (Sharmin *et al.*, 2020a), a micro-level resource management mechanism was proposed. Their framework establishes a federated fog acting as a consortium to share free resources among the consortium members. In addition, they proposed a price-based model for sharing the resources while limiting the offloading among units relative to other consortium members. The authors in (Veillon *et al.*, 2019b) devised a solution for improving the latency metric of video services. Specifically, their solution consists of reducing the latency of streaming video through federating fog parties based on caching and fetching video data from neighboring fog nodes. In (Shamseddine *et al.*, 2020b), the authors provided a novel federated fog architecture for serving IoV and modeled the federated fog formation problem by combining both genetic and machine learning models to optimize the overall service quality. In (Hammoud *et al.*, 2021), we proposed a stable federated fog formation mechanism using an Evolutionary Game Theoretical model. The objective was to strengthen the relationship between the fog provider and the federations by achieving an evolutionary stable strategy for the game.

4.3.3 Discussion

The literature presents potentially fine approaches through enhancing the components and processes within the federated learning architecture. Nevertheless, there are still some flaws in terms of applicability within an IoV environment. For instance, no work in the literature has considered the dynamic behavior of the participants and the changes that may occur to the topology when facing dynamic circumstances. In addition, training effective IoV applications may require a large amount of large-size models to be forwarded for aggregation which can cause degradation to the service quality by taking extra time and energy for uploading them. It is worth mentioning that the fog-based solutions have addressed a part of the aforementioned issues, nonetheless, they all assumed full cooperation and collaboration among the fog nodes without a business-driven solution for the stakeholders. Furthermore, the fog federation formation mechanisms presented in the literature are not fully stable in this paradigm when the rewards for the federation are based on the service quality. Besides, the majority of scholars were testing their frameworks by relying on MNIST[5] dataset from which it is relatively easy to discover statistically significant patterns. Thus, a stable and detailed infrastructure tailored to IoV is needed to enable the smooth execution of the federated learning applications. The infrastructure should prove its efficiency by managing the training of realistic IoV applications.

4.4 Dynamic Horizontal Federated Learning Approach

Fog Federations enriches the IoT infrastructure with the benefits of sharing fog resources and distributing the workload among the federation members. However, the federation concept was not studied in the literature while considering the dynamic behavior of users under IoV settings, which may cause a degradation in the services quality and deviations from the federations. Therefore, a solution must be devised to maintain a satisfactory quality for the mobile end users in such a dynamic environment. Moreover, federated learning requires a specific architecture by design, as it involves components that differ from any other applications. Hence, we present in

[5] http://yann.lecun.com/exdb/mnist/

this section an architecture tailored to supporting federated learning through the federated fog concept.

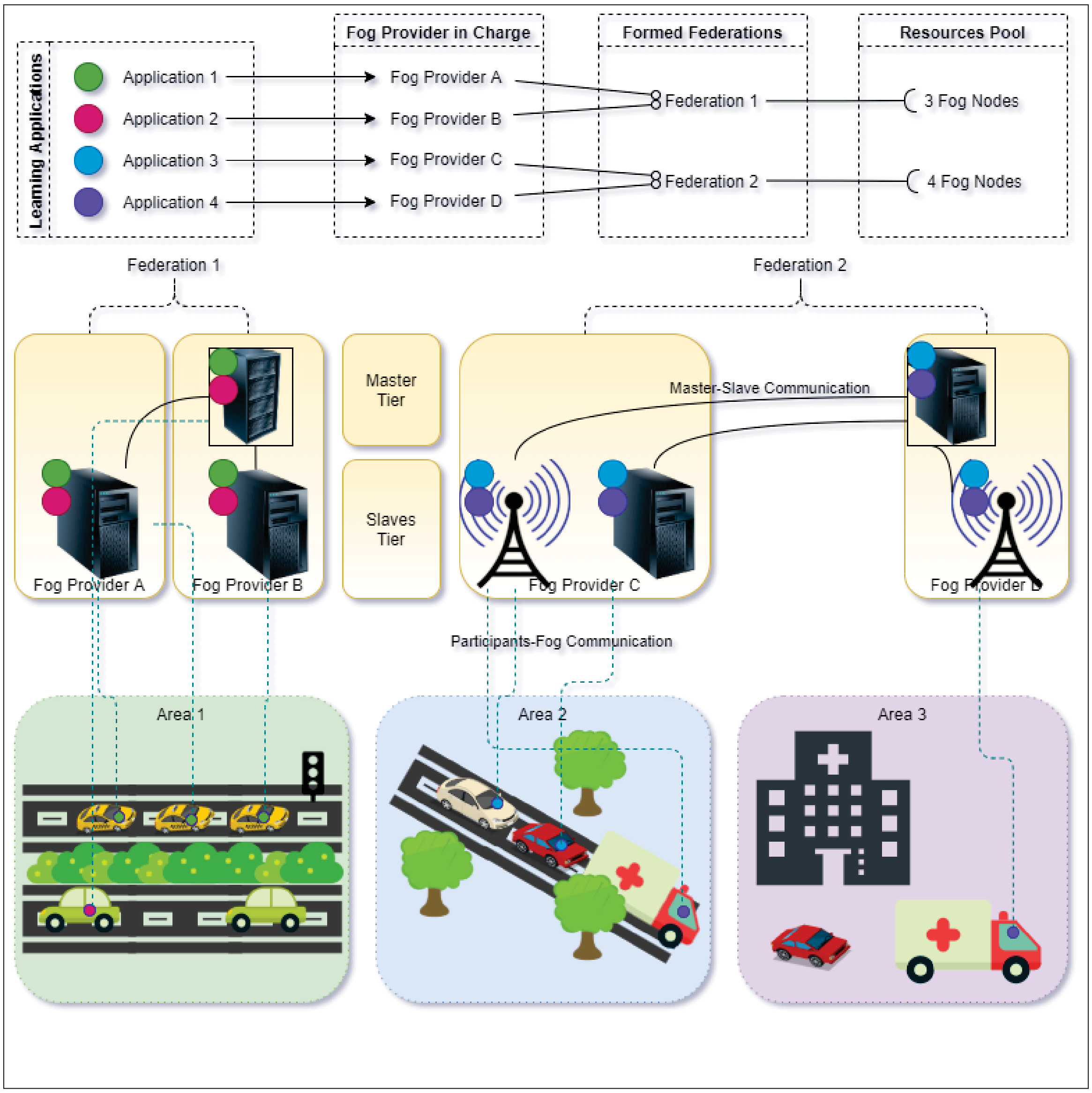


Figure 4.2 Dynamic Horizontal Federated Learning Architecture

Fig. 4.2 summarizes the proposed architecture. It is a Horizontal-based architecture as the car brands have different sets of drivers, nevertheless, they all want to train their intelligent road-safety applications, which means having similar features to a certain extent. The fundamentals of such an architecture can be split into two parts: architecture components and processes. The components are the actors in our architecture, and the processes represent their interaction for executing the federated learning procedure in this dynamic environment. Below, we describe their elements and their roles.

4.4.1 Architecture Components

- **Clients**: are the brands that want to train one or several IoV learning applications to enable road safety and full vehicular autonomy. They assign providers to operate their learning applications.

- **Fog Providers**: are the main source of resources in our architecture. They are independent parties that provide computing and networking infrastructure to the clients that require resources to operate their applications. A service agreement is made between the provider and the client to agree on the resources needed, the duration of the contract, the price which the client should pay, and other contract-based terms.

- **Fog Nodes**: are owned by the fog providers. They offer accessibility to the services which are deployed on them for the users. The nodes are split into two types: master nodes and slaves nodes. In general, a node is responsible for collecting the trained models from the participants, applying an aggregation technique to unify these updates, and then, selecting the participants for the next round and forwarding the latest model to them.

- **Fog Federations**: are coalitions of fog providers which are formed according to certain criteria. They enable a horizontal learning topology throughout their internal cooperation and aggregation techniques. Their main role is to enhance the QoS by offering an efficient infrastructure for federated learning in IoV. In our architecture, we will refer to the shared pool of nodes as Federations.

- **Master Node**: in a single federation, there exists one master node among the slaves, which is elected to have additional functionalities. In addition to its general duties, it simply collects and applies the second level of aggregation to the models aggregated by the fog nodes, and redistributes the output to its slaves to unify the model for the whole federation.

- **Participants**: are essentially the trainers of the basis of the architecture (i.e. vehicles), they are subscribed to certain federated learning-based applications and they engage in their learning process. In Fig. 4.2, the applications are illustrated by a colored circle that is drawn on top of the vehicle.

4.4.2 Architecture workflow

In this section, we detail the processes of our proposed approach, illustrated in Fig 4.3, as follows:

1. **Inquiry**: in the first step, multiple clients contact the providers to arrange a certain agreement on managing their IoV federated learning applications training phase.

2. **Initialization**: the providers receive these requests, set up a price to charge these requesting clients, and agree on the service quality provided. Each application is required to start learning from a certain model that can either be generated randomly or based on training initial data as a pre-processing phase.

3. **Resources identification**: each provider, owner of fog servers, identifies its available fog nodes that shall participate in the learning phase for the current round of training, and sets up the contemporary non-converged model of each training application it is responsible for.

4. **Offering training**: each application relies on many participants for enhancing its decision-making procedure. Thus, available participants for the ongoing round connect to the servers and offer their resources to update the application models they signed up for.

5. **Federations formation**: at this stage, federations will be formed by the providers for enriching the dynamic infrastructure. The federations are formed according to various metrics that include costs, bandwidth, participants, and other metrics. We will discuss the formation mechanism in-depth in the next section.

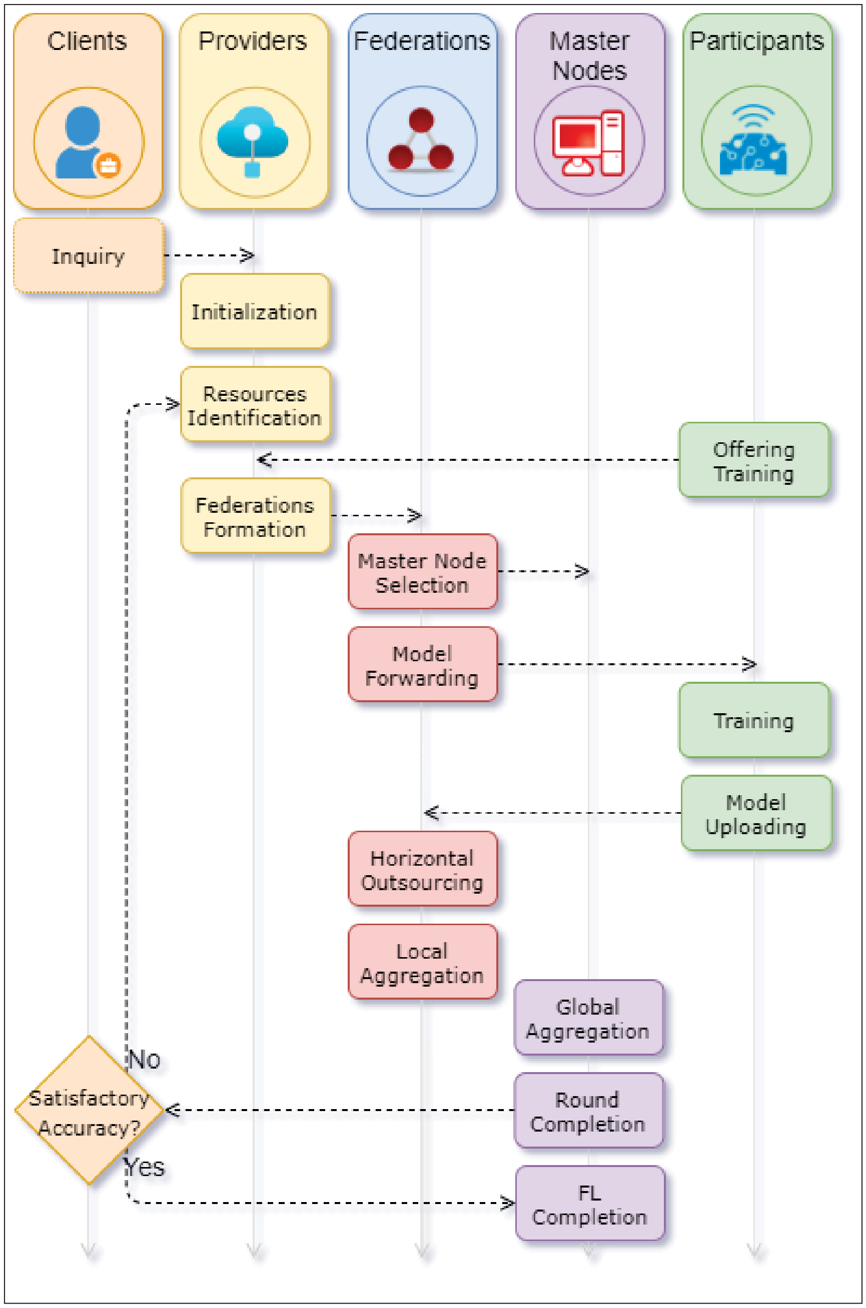


Figure 4.3 Framework Processes Timeline

6. **Master node selection**: once the federations are formed and the fog nodes are determined, the providers elect a fog server to become the master node for global aggregation.

7. **Model forwarding**: afterward, the federations nodes forward the ongoing models to the participants.

8. **Training**: the participants train the received models for a predetermined duration with the use of an optimizer, such as the stochastic gradient descent (SGD).

9. **Model uploading**: Once the training is finished, the participants forward the trained model to the federation.

10. **Horizontal outsourcing**: the members of a federation split the received tasks among themselves by distributing their workload for enhancing the service quality, i.e., processing the tasks faster and with lower cost.

11. **Local aggregation**: the nodes apply the local aggregation technique to the models to obtain a locally aggregated model, and then, forward the latter to the federation master node.

12. **Global aggregation**: the master node receives the aggregated models from its slaves and applies a global aggregation to obtain a global and unified model for the learning application

13. **Round completion**: after the completion of a round, each application is assessed through a testing mechanism to check whether it reached its maturity and convergence or should be trained further. We repeat the steps from 3 to 13 until the convergence of all models.

4.5 Hedonic Federations Formation

We focus in this section on the federations formation game and solution for stabilizing the service quality of the set of fog federations within the dynamic environment. The main reason to stabilize the QoS is to enable the smooth execution of the aforementioned processes in the proposed federated learning architecture workflow.

4.5.1 System Model

In the dynamic IoV environment, we follow the general assumption that the servers of the fog providers are statically located in various zones (i.e. immobile resources). Whereas the participants, i.e. vehicles, are mobile and can change zones during their journeys. We detail below the system components.

Table 4.1 Summary of Notations

Notation	Description
FP	set of all fog providers
fp_i	fog provider i
N_{fp_i}	set of all fog nodes (servers) belonging to fp_i
n_i	fog node i
$VM(n_i)$	virtual machines running on fog node i
F	set of all fog federations
f_i	fog federation i
FP_{f_i}	set of all fog providers allocated within f_i
A	set of all IoV Learning applications
A_{f_i}	set of all applications (services) that belong to f_i
V	set of all vehicles (users)
v_i	vehicle i
$A(v_i)$	set of all applications vehicle i is subscribed to
a_i	IoV Learning application i
$acc^t_{a_i}$	model accuracy of application i at time t
$loss^t_{a_i}$	loss function of application i at time t
$Cont(fp_j, f_i)$	fp_j's share of resources in f_i
$Rev(f_i)$	revenue f_i
R_{fp_j,f_i}	payoff of provider fp_j from federation f_i

4.5.1.1 IoV Learning Applications

Learning-based applications, such as Autonomous Driving, require intensive training before converging and becoming ready for deployment. We assume the existence of $\mathbf{A} = a_0, a_1, a_2, ..., a_m$ IoV learning applications that need to go through a federated learning phase to become relatively accurate in making decisions. a_i is the i^{th} application and is supported by one or more car brands. Each Learning application is characterized by its model accuracy $acc^t_{a_i}$ and loss function $loss^t_{a_i}$

at time t. acc_{a_i} and $loss_{a_i}$ represent the latest model accuracy and loss function for application a_i. Each application aims for the participation of β_{a_i} vehicles at each training round where β_{a_i} is a large number due to the fact that data is Non-Independent and Identically Distributed (Non-IID) as each vehicle differs in terms of data rows and ratio per label distribution.

4.5.1.2 Application Users

As implied by the term Internet-of-Vehicles, the users of the IoV applications are mainly the vehicles $\mathbf{V} = v_0, v_1, v_2, ..., v_n$ themselves. Vehicle v_i embeds IoV applications in accordance with (1) its brand and (2) the applications it is subscribed to, represented by the set $A(v_i)$. The users are assumed to cooperate when requested to train and contribute to these applications. The vehicles are equipped with limited computational resources that may prevent them from engaging in the training process when these resources are occupied by other tasks.

4.5.1.3 Fog Providers

Due to the previously mentioned network limitations, brands assign the management of their IoV learning applications to Fog Service Providers. A fog provider n, i.e. $fp_n \in \mathbf{FP}$, may own multiple fog servers (nodes) $N(fp_n) = n_0, n_1, n_2, ..., n_o$, located within specific geographical areas, relatively close to the end-users by whom the services need to be accessed. With the advent of virtualization, fog nodes now encompass multiple virtual machines running services independently. $VM(n_i)$ is the set of virtual machines running on node n_i. Such machines possess computational power, measured in terms of allocated cores, memory, and storage capacities.

4.5.1.4 Fog Federations

Fog Federations are formed and canceled with the consent of their members. Initially, two or more providers, $FP_m \subset FP$, form a federation f_i intending to improve their computational performance and offer a better service quality. $FP_n \subset FP \mid FP_m \cup FP_n = \varnothing$ may join f_i later on, or form their own federation, f_j. Similar to (Hammoud *et al.*, 2021), we assume a fair monetary distribution among the federation participants, such as:

$$R_{fp_{j,f_i}} = Rev(f_i) \times Cont(fp_{j,f_i}) \tag{4.1}$$

where $U(f_i)$ is the revenue of f_i and $Cont(fp_{j,f_i})$ is the contribution of resources made by provider j in federation i, which can be obtained by:

$$Cont(fp_{j,f_i}) = \alpha \frac{R^c(fp_{j,f_i})}{R^c(f_i)} + \beta \frac{R^m(fp_{j,f_i})}{R^m(f_i)} + \gamma \frac{R^s(fp_{j,f_i})}{R^s(f_i)} \tag{4.2}$$

where R^c, R^m, R^s represent the cores, memory, and storage resources. $R^c(f_i)$ are the cores allocated to federation i, and $R^c(fp_{j,f_i})$ are the cores contributed by fp_j into f_i. To fairly highlight the importance of each type of resource we assign the weights $\alpha, \beta,\ \&\ \gamma$, such that:

$$\alpha + \beta + \gamma = 1 \tag{4.3}$$

4.5.2 Hedonic Games

Game theory is a study of optimizing the outcome of the players, i.e. engaged parties, by determining their optimal strategy. A coalitional game is a game-theoretical model that evaluates the interaction of players when they split into groups. Such a game results in a set of coalitions formed by the players. In other words, a coalition $S_i \subset \mathbb{N}$ is the i^{th} coalition of players that agreed to form a union for sharing their resources, where $\mathbb{N}$ is the set of all players. Π is the set of all coalitions $[S_1, S_2, S_3, ..., S_m]$ where $\cup_{i=1}^{m} S_i = \Pi$. A coalitional game is considered Hedonic if (1) the utility of any player depends only on the players within the same coalition and (2) the coalitions are established according to the preferences of the players.

Preference Function: a preference relation $>_{p_i}$ indicates the decision of player p_i when facing two choices. $S_j >_{p_i} S_k$ denotes that p_i prefers joining coalition S_j over S_k. A preference function $\varrho_{p_i}(S_j)$ denotes the preference of p_i for joining coalition S_j.

$$S_j >_{p_i} S_k \iff \varrho_{p_i}(S_j) > \varrho_{p_i}(S_k) \tag{4.4}$$

Stability: the objective of the game is to devise a set of coalitions that is stable and resistant to deviations in the sense that no player is willing to leave its current federation and join a different one, assuming that none of the other coalitions can provide it any better utility. That is to say,

$$\forall p_i \in \mathbb{N}, S_j \geq_{p_i} S_k \mid p_i \in S_j \;\; \& \;\; k \neq j \;\; \& \;\; 1 \leq k \leq m \tag{4.5}$$

The usage of $\geq_{p_i}$ instead of $>_{p_i}$ is to emphasize that the preference of player i for joining S_j is *at least* as good as joining S_k

4.5.3 Game Formulation

Our proposed game is a coalition formation game resulting from the competition among some big brands looking for forming multiple independent federations rather than just forming a grand federation where all competitors collaborate. For the sake of simplifying the parameters used along the rest of the work, we replace the notation for the players set $\mathbb{N}$ by the set FP because the players in our game are the fog providers. Likewise, the set of coalitions S is replaced by the set of the fog federations F that we are forming.

Our game is a Hedonic game because it suffices both conditions mentioned in the previous part. The first condition is valid because the utility of the player is solely related to his contribution while considering the other players' within the same coalition. The second condition holds as well, as the preferences can be defined by the following equation:

$$\varrho_{fp_i}(f_j) = \frac{SU^{f_j}(fp_i)}{U(fp_i)} \times R_{fp_{j,f_i}} \tag{4.6}$$

where $SU^{f_j}(fp_i)$ is the number of participants that need to connect to player i but cannot reach any of its virtual machines within the predefined acceptable QoS threshold (while it can establish a good connection with one or more VMs allocated inside federation j) and $U(fp_i)$ is the total number of participants that should engage in any of fp_i's learning processes. The motivation behind such parameters is derived from the fact that our architecture is tailored to a dynamic environment with moving users, thus the heavy dependency of our preference function on the connection status between the provider and the participant. The consolidation of both (1) users and (2) payoff metrics within the preference function is to assure a sufficient number of participants with a satisfactory link to engage in the learning process and an acceptable payoff to the provider itself.

4.5.4 Federation Formation Algorithm

Following the structure of our architecture devised in Section 4.4, each round can be accompanied by a formation mechanism for the set of federations if needed[6]. In each round, a provider dedicates its available resources to only one federation. The formation of the federations is executed in a decentralized manner by the providers themselves, as they decide to join a federation by their own choice, according to their own preferences. The set of formed fog federations is based on the fusion of all decisions of the same round until stability occurs where no additional decision can benefit any of the federations' members, i.e. no player has incentives to switch to a different federation. The formation mechanism is presented in Algorithm 4.1. The input of this algorithm consists of the last set of federations, i.e. at Round t, and the output is the updated set for Round $t + 1$. We introduce three new variables, F^*, F^{t+1}, and L. The first variable is a temporary variable to hold the last formation. The second one is the final set of federations and we initialize it as empty at the beginning of the algorithm. The last one is a list that denotes the players who made a move in the current sub-round. The core of the algorithm starts with the 'while' statement at the fifth line. It indicates that the algorithm stops only when the new set of federations is identical to the temporary set. In other words, the algorithm stops only when there are no changes in the members of the federations as all of the players are satisfied with their current federation. In each loop, i.e. sub-round, we update the result by setting F^{t+1} to be F^*. Then, for each player in each federation, we calculate the preference value for joining other federations and, accordingly, it selects the most suitable federation for the current round. Then, the set of federations resulting from this algorithm engages in the remaining federated learning

[6] Federations may maintain the same formation in the next round if all the members can still be satisfied with their QoS

processes presented in our architecture. It is worth mentioning that sometimes F^t and F^{t+1} can be identical which means that the federations structure at time *t* is still suitable at time *t+1*.

Algorithm 4.1 Federation Formation Mechanism

```
Input: F^t
Output: F^{t+1}
F* ← F^t;
F^{t+1} ← Ø;
L ← Ø;
while F* ≠ F^{t+1} do
    F^{t+1} ← F*;
    foreach f_i ∈ F* do
        foreach fp_j ∈ f_i do
            if fp_j ∉ L then
                Curr ← ϱ_{fp_j}(f_i) ;
                foreach f_k ∈ F* do
                    New ← ϱ_{fp_j}(f_k) ;
                    if New > Curr then
                        f_i ← f_i \ {fp_j} ;
                        f_k ← f_k ∪ {fp_j} ;
                        Curr ← New ;
                    end if
                end foreach
                L ← {fp_j} ∪ L ;
            end if
        end foreach
    end foreach
    L ← Ø;
end while
```

4.6 Experimental Evaluation

4.6.1 Experimental Setup

Computer Characteristics: To evaluate the performance of our proposed approach, we run extensive computational jobs on Graham and Cedar clusters offered through ComputeCanada[7].

[7] https://www.computecanada.ca/

We equip each job with 32GB of RAMs, 1 CPU per task, and 1 GPU node for faster machine learning processing.

4.6.1.1 Dataset

The main purpose of our proposal is to maintain an efficient execution of federated learning tasks while maximizing the Quality of Service in IoV settings. Therefore, we rely on a combination of two datasets to perform the experiments due to the lack of a unified dataset that contains vehicular road paths with driving assistant data. The used datasets are the following:

Trajectories data: The first dataset consists of data extracted from SUMO[8], short to Simulation of Urban MObility, which is an open-source software that simulates realistic vehicular trajectories according to the provided simulation settings. We generated 300 vehicles, each with a unique trajectory starting at various times. We intended to rely on a moving set of cars to show the performance of our proposed architecture in a dynamic environment. Nevertheless, when a vehicle finishes its trajectory (at a random time), it reaches a parking place where it does not change its location any further.

Machine learning data: For the purpose of simulating an actual development of a Driving Assistant model, we relied on images of traffic signs imported from Kaggle[9], which, when processed, can equip the vehicles with a Traffic Sign Recognition (TSR) technology for enhancing safety on the road. We show a sample of this dataset in Fig. 4.4. One of the characteristics of this dataset is that most of the images are blurred and cannot be easily identified. We intended to choose this dataset for the purpose of our IoV application due to the fact that the cameras of the vehicles capture the traffic signs while driving when attempting to identify them. Therefore, the image quality is not guaranteed in real-time.

[8] https://www.eclipse.org/sumo/

[9] https://www.kaggle.com/valentynsichkar/traffic-signs-preprocessed

Figure 4.4 Traffic Signs Samples

4.6.1.2 Benchmarking models

We compare our approach with three other approaches. The first one is a standard federated learning approach supported by fog computing, such as the one presented in (Zhou *et al.*, 2020a), where no federations are formed to enhance the QoS. We refer to this approach as *Fog Computing Approach*. The second one is a static Hedonic formation approach for IoT paradigm presented in (Anglano *et al.*, 2018b). It is worth mentioning that the second approach is intended for supporting IoT applications in general - not federated learning. Therefore, we replace our formation technique with theirs and maintain the rest of the processes mentioned in our approach for enabling a quantitive comparison. We refer to the latter in the simulation as *Anglano Approach*. We also compare our approach with the centralized training method in the machine learning evaluation, referred to as *Centralized Approach*.

4.6.1.3 Parameters and Applied Methods

Each vehicle has a different velocity and random source and destination as simulated in SUMO. We simulated 2 scenarios where the first one consists of equipping each vehicle with a random set of collected data, between 1000 and 3000 traffic sign images (total number of images is 200,000) distributed in an IID manner label-wise, i.e. 42 labels each which is the total number of labels. The second scenario consists of randomly assigning the number labels to the vehicles. We assume having 12 fog providers spread across the map and they are in the process of managing a similar TSR model requested by car brands.

Machine learning model: We set up a Residual Neural Network (ResNet) with 56 layers, and we set the training epochs for each federated round to be 10. To make the comparison fair for the Centralized method, we also present each 10 epochs from the Centralized results in the figures as 1 round. The used optimizer is SGD. It is worth mentioning that we utilized a ResNet due to its proven efficiency on the Signs dataset according to the implemented methods in the dataset's page on Kaggle[10].

Aggregation technique: For the local and global aggregations, we adopted the Federated Average (FedAvg) function for averaging the sequences according to the number of trained data per each received model.

Initial training model For a fair comparison, we devise an initial model by training randomly selected 1000 traffic signs. The acquired accuracy was 6%. The three analyzed approaches are then boosted from the initial model at the beginning of their simulations.

4.6.2 Results and Discussion

We run our simulation 10 times for each experiment and present the averages of these 10 runs. Then, we split the results into two categories. In the first set of results, we evaluate the service quality provided to the vehicles. In the second set of results, we measure the performance of the trained model under different data distributions. We compare our approach to the two benchmark approaches defined in Section 4.6.1.2 (Anglano Approach, and Fog Computing Approach).

4.6.2.1 Resource Availability

The average latency of the invocations from the vehicles to the servers as a function of the timeline is presented in Fig. 4.5. The x-axis denotes the timeline where each value is a snapshot taken from the vehicles' paths that also represents the beginning of a new communication round. The y-axis is the average latency. We observe that our federation formation mechanism acquires better latency when invoking tasks and requests. In particular, our approach achieved an average

[10] https://www.kaggle.com/valentynsichkar/traffic-signs-preprocessed/code

of 18ms compared to 20ms and 23ms obtained by Anglano and Fog Computing approaches, respectively.

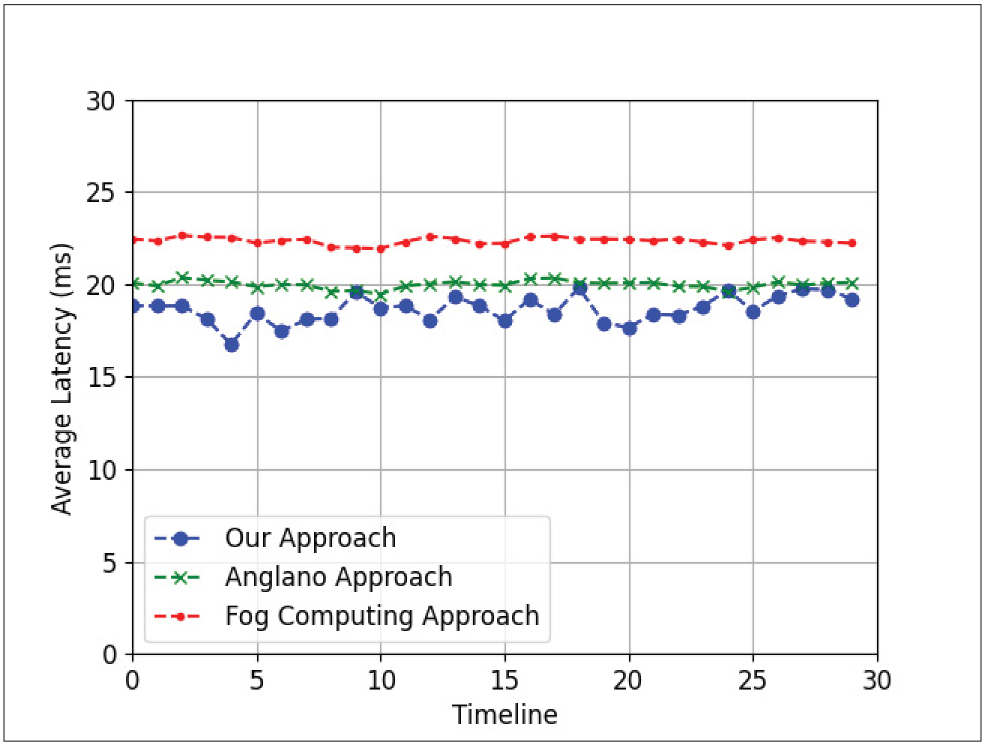


Figure 4.5 Invocation Delay

Furthermore, Fig. 4.6 shows the rate of satisfactory invocations as a function of the timeline. The x-axis indicates the timeline as usual, whereas the y-axis is the rate of the vehicles with satisfactory service quality. A satisfactory invocation implies that the vehicle can establish a satisfactory connection with the fog server, which qualifies the former to integrate with the current training round. It can be observed that the Fog Computing Approach, where federations were not applied, has the lowest rate of around 0.2. Anglano Approach does better due to having federations and it reaches 0.35 on average. Our approach outperforms both of the benchmark models and reaches 0.39 on average. Nevertheless, it is noticeable that sometimes Anglano's approach can acquire similar Invocation Satisfaction Rate and average latency values, e.g. at times 9 and 24. This is due to the fact that both approaches rely on federating resources, and Anglano's formation might reach relatively suitable states that intertwine with ours for certain snapshots of the simulation. We conclude from Fig. 4.5 and 4.6 that our approach can better utilize the resources due to the dynamic formation mechanism, thus establishing satisfactory connections with the vehicles due to its superior services migration among federations nodes.

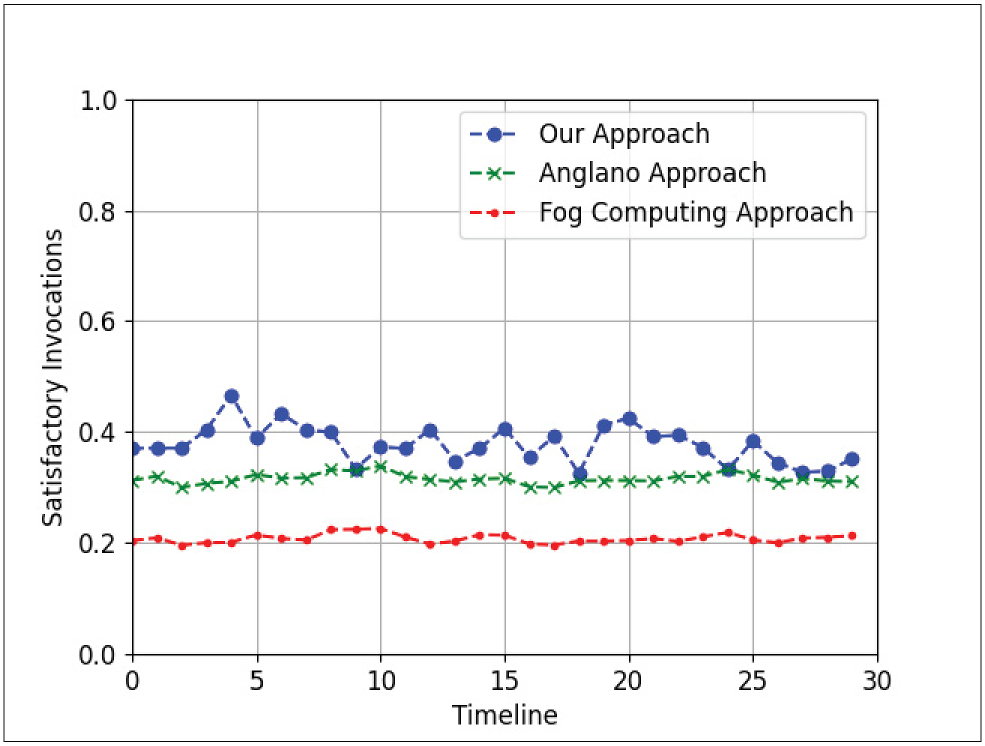


Figure 4.6 Invocation Satisfaction Rate

4.6.2.2 Federated Learning

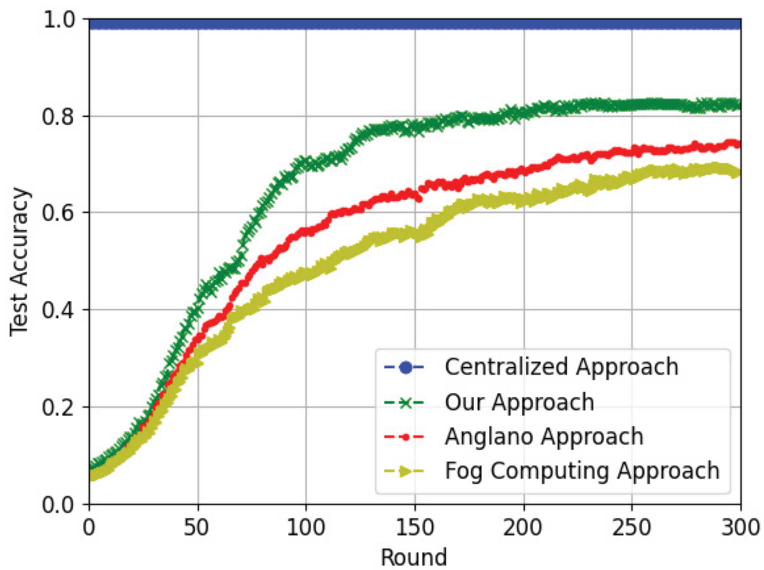
To test the performance of the federated learning models, we ran the four approaches (i.e., Our Approach, the Centralized Approach, Anglano Approach, and Fog Computing Approach) for 300 rounds each, starting from the same initial model, and visualized the models' accuracy and loss as they progress. Fig. 4.7 and 4.8 show the results for the algorithms under IID and non-IID data settings, respectively. In Fig. 4.7a, like any other model training, all of the approaches start with low accuracy and tend to seek convergence with time except for the centralized approach which achieved a 0.98 accuracy after the first round of training. The high accuracy perceived in the first round is due to the fact that all the data are available for the server to train and a round is comprised of 10 epochs. The fog computing approach kept on improving its models until reaching 0.68 accuracy by the end of the training period. Anglano Approach acquires better results, due to the cooperation among the providers, as it reaches 0.74. Our approach was able to outperform both of the non-centralized approaches by converging to 0.82 at Round 200, making our approach the most efficient among the considered federated learning methods. The case of IID distribution of data can be more suitable when the servers are in control of the data, such as in distributed learning settings (Wang *et al.*, 2021).

Fig. 4.7b presents the models' losses calculated from

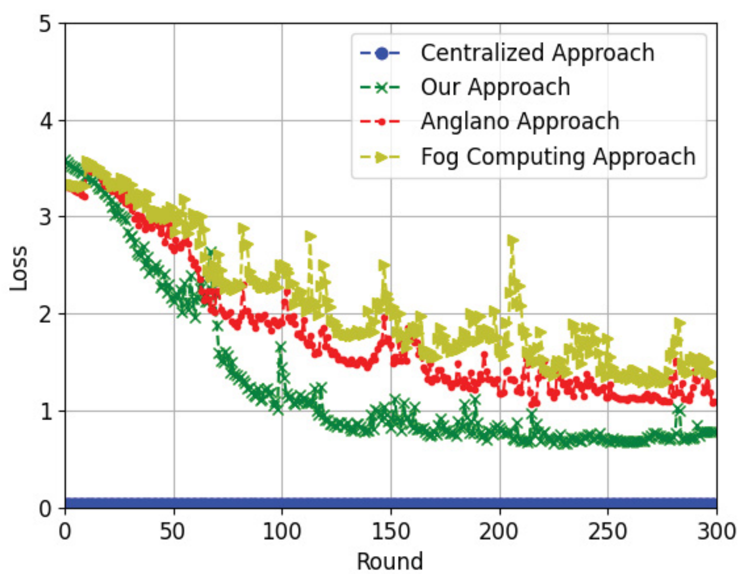
$$D(S, L) = -\sum_i L_i \times log(S_i) \tag{4.7}$$

where the set S is the set of probabilities of the classes the prediction belongs to, and L is the one hot encoded labels set that indicates the correct prediction. We also note that the loss of the centralized approach is stable at 0.04 as it can learn from all the data at once. Due to the wider inclusion of participants, our approach is able to reduce the loss to almost 0.7 by the end of the training rounds, whereas the other federated approaches are still suffering from high losses due to the delay in receiving new models' updates.

Fig. 4.8 shows that our model can also provide better results in terms of accuracy (Fig. 4.8a) and loss (Fig. 4.8b) than the other federated approaches for the case of non-IID settings. Nevertheless, the progress is slower due to the fact that the data is distributed in an uneven manner in terms of labels. It can be noticeable that the convergence occurs at Round 245 in the non-IID rather than Round 200 in the IID settings. We conclude from these figures that our approach can enhance the learning mechanism of the architecture by including better service quality for larger users participation, which leads to an earlier model maturity close to the optimal training in a short number of rounds. Although our approach achieved lower accuracy and higher losses than the Centralized Approach, we can still achieve near-optimal results in federated settings by increasing the number of fog nodes, participants, and/or communication rounds. Nevertheless, we limit ourselves to only selecting acceptable QoS participants as an active sampling method.



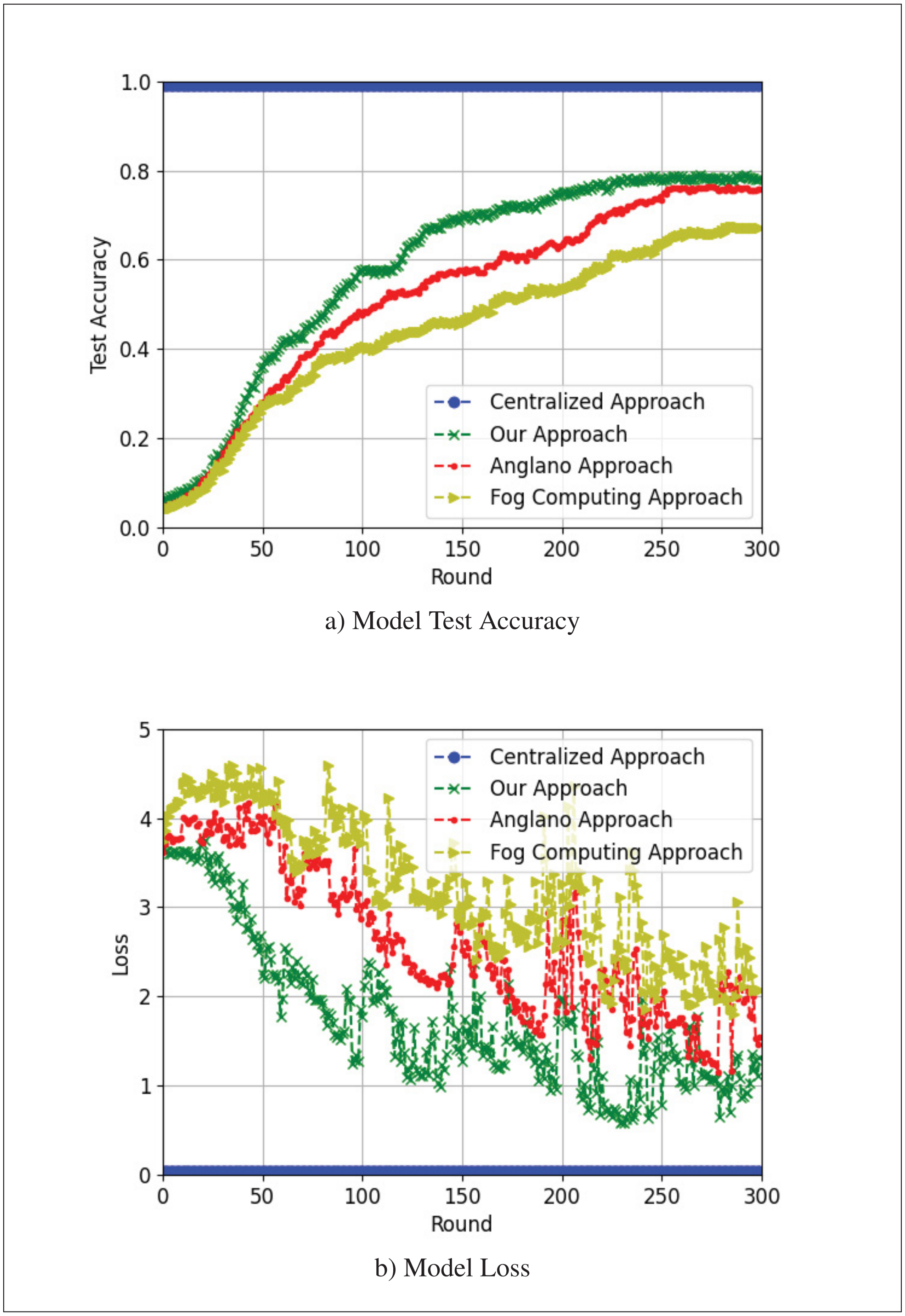
a) Model Test Accuracy



b) Model Loss

Figure 4.7 Federated Learning Simulation Results: IID settings

In addition, the simulation requires more intensive computational power for a larger number of communication rounds.



a) Model Test Accuracy

b) Model Loss

Figure 4.8 Federated Learning Simulation Results: non-IID settings

4.7 Conclusion and Future Work

The advancement of technology, especially in the Internet of Things development, has shifted the Vehicle Ad-hoc Networks into the Internet of Vehicles. IoV has attracted companies and car brands into investing in smart and autonomous vehicles which can make decisions on behalf of the drivers towards optimal road safety. Federated learning is a promising technique to train intelligent models from users' data while preserving privacy at the same time. We proposed in this research a federated learning architecture assisted by fog federations to enhance the accuracy and service quality of the IoV intelligent applications. We relied on the Hedonic coalitional game and maintained the stable set of federations by using the preferences of each fog provider independently. For the sake of testing our approach, we simulated an environment for detecting traffic signs on the road. The experimental evaluation revealed that our approach can achieve superior accuracy and quality of service of the learning procedure when compared to other approaches proposed in the literature. This is resulting from the fact that our approach relies on state-of-the-art methods used to optimize the federated learning mechanism.

For future work, there are several issues worth considering to further improve our framework in certain aspects. First, the federation formation process in our approach relies on a real-time assessment of the environment. It could be of interest to investigate a federation formation mechanism that can predict the optimal set of formed federations a priory with the help of Artificial Intelligence to prepare for services migration. Second, we assume that the collaboration among the fog providers in terms of model aggregation is given, especially when training homogeneous applications. While collaboration seems efficient and leads to better results, it can be optional for fog providers to rely on privacy-preserving techniques and strictly limit the visibility of their collected models to their own servers without allowing other providers to benefit from such knowledge. In addition, in the current architecture, the selection of the master fog node is executed by an election procedure without specifying its details. In this context, it could be of interest to rely on a trust-driven mechanism to decide which node gets to be the master in each federation. Furthermore, our architecture relies on a QoS-based sampling mechanism, thus it might be interesting to evaluate the architecture's performance under a different active sampling technique such as the ones presented in the literature. Finally, even though our approach is tailored to federated learning, it can still be extended to cover distributed learning applications in case the data is not critical and can be exposed to external parties. The realization of this feature can occur by having fog nodes engaging in the training procedure or by offloading training tasks from the top of the topology towards the bottom.

CHAPTER 5

A BLOCKCHAIN-BASED HEDONIC GAME SCHEME FOR REPUTABLE FOG FEDERATIONS

Ahmad Hammoud[1] , Rabeb Mizouni[2] , Hadi Otrok[2] , Shakti Singh[2] , Azzam Mourad[3] , Zbigniew Dziong[1]

[1] Ecole de Technologie Superieure (ETS), Montreal, Canada
[2] Department of EECS, Center on Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, UAE
[3] Department of Computer science and Mathematics, Lebanese American University, Beirut, Lebanon

Paper Submitted to IEEE Transactions on Services Computing on Mar 31, 2023.

5.1 Abstract

Fog computing empowers the Internet of Vehicles (IoV) paradigm by offering computational resources near the end users. In this dynamic paradigm, users tend to move in and out of the range of fog nodes which has implications for the quality of service of the vehicular applications. To cope with these limitations, scholars addressed forming federations of fog providers for task offloading purposes. Nonetheless, a few challenges remain a burden for the formation of the federations. The formation mechanisms used to structure the federations of providers are still not fully stable. This causes a problem because a structureless federation can lead to an underperforming infrastructure. Furthermore, most of the literature ignored the honesty metrics of the providers and how trustworthy they are in allocating the agreed-upon resources for processing the tasks. Moreover, adopting a central reputation mechanism is questionable in terms of reliability due to many complications including the lack of consensus. In this work, we develop a Blockchain-based reputation mechanism for assisting the formation of fog federations for IoV applications. Our mechanism comprises on-chain smart contracts for storing and manipulating the providers' reputations and an off-chain Hedonic-based formation process that considers the parameters extracted from the chain to build the federations. We develop smart contracts using Solidity and deploy them on the Ethereum Blockchain. We test our mechanism using the EUA dataset as a proof of concept and compare it to other works in the literature.

The results obtained show that our approach is able to enhance the overall payoff and quality of service in the IoV paradigm.

5.2 Introduction

Internet of Vehicles (IoV) emerged recently as part of the Internet of Things (IoT) to enhance the driving experience of drivers (Yang, Wang, Li, Liu & Sun, 2014). It is a distributed network that supports the use of data created by connected cars and vehicular ad-hoc networks (VANETs) (Hammoud *et al.*, 2020c). One of the features of IoV is to allow interaction with surroundings in real-time by relying on Vehicle to Everything (V2X) communication to enable Intelligent Transportation Systems (Zhou, Xu, Chen & Wang, 2020b). IoV applications, such as collision avoidance and object detection, require instant processing that the vehicles themselves might not be able to offer (Dai, Liu, Chen & Lai, 2020). Offloading tasks to the cloud comes at a cost; the high latency for sending the data and receiving the response plays a major bottleneck for these time-sensitive applications (Jebbaoui, Mourad, Otrok & Haraty, 2015). To overcome this limitation, the fog computing concept was introduced (Yi, Li & Li, 2015). Fog servers are similar to cloud servers in terms of functionality, except that they offer fewer resources. Fog servers are physically placed near the end-users in a way that transferring data to the fog is relatively less expensive than to the cloud. In addition, offloading tasks to the fog yields faster results in terms of delays and latency.

Due to the dynamic nature of IoV and the resource shortage of the fog providers (Ghobaei-Arani, Souri & Rahmanian, 2020), the latter may not be able to fulfill the quality of service (QoS) requirements as vehicles move away from their fog server's coverage (Zhang, Zhang & Chao, 2017). To cope with the environment dynamicity, scholars offered techniques to form collaborative clusters of fog providers in order to support task offloading in larger zones and maintain low service delays (Hammoud *et al.*, 2022b). Collaborative fog computing can compensate for the resource shortage, by allowing resource sharing among the providers, and connection handovers when users are better to switch to a different server. The main idea behind clustering is to enhance service availability and increase the profitability of the providers. Nevertheless, when the clusters are not well engineered, the servers may perform lower than what is expected (Hammoud *et al.*, 2020a).

Team and federation (coalition) formation has been studied in depth in the literature in many fields, including Social Networks (Anagnostopoulos, Becchetti, Castillo, Gionis & Leonardi, 2012), Robotics (Smirnov, Sheremetov & Teslya, 2019), Project Management (Tseng, Huang, Chu & Gung, 2004), Sports (Tavana, Azizi, Azizi & Behzadian, 2013), etc. Recent efforts addressed formation techniques in cloud computing (Hammoud *et al.*, 2020a), wireless networks (Nolan & Doyle, 2007), and fog/edge computing (Hammoud *et al.*, 2022a). Some of the techniques used include game theoretical models and meta-heuristics, which provide a performance increase when it comes to refining the IT infrastructure. In cloud resource management, a federation formation was mainly highlighting the payoff maximization of the cloud providers without being considerate of the service quality provided (Hammoud *et al.*, 2020a). Whereas, a federation of fog resources should consider the interaction between the end users with the fog servers and their coverage area in order to provide a satisfying service (Shamseddine *et al.*, 2020b). For instance, as shown in Fig. 5.1, we can observe a scenario where two vehicles are connected to specific fog servers, and these fog servers form a federation. Notably, when the serviced vehicles exit the coverage zones of their respective fog servers, the communication is seamlessly maintained by other fog servers within the federation. This ensures the preservation of quality of service throughout the transition. Therefore, an efficient federation formation mechanism should consider the location of the users jointly with the allocated servers.

In fog federations, all the nodes are expected to well-behave in the sense that fog providers are willing to allocate the resources they agreed on. A passive malicious fog node is considered as misbehaving when it reneges on its service level agreement (Hammoud *et al.*, 2018). These malicious fog nodes pose a significant threat to the IoV paradigm, and their presence can have severe consequences. When fog providers fail to fulfill their service level agreements or engage in malicious behavior, the QoS of the entire fog federation is compromised, leading to increased processing delays and reduced user satisfaction. These negative outcomes directly impact the

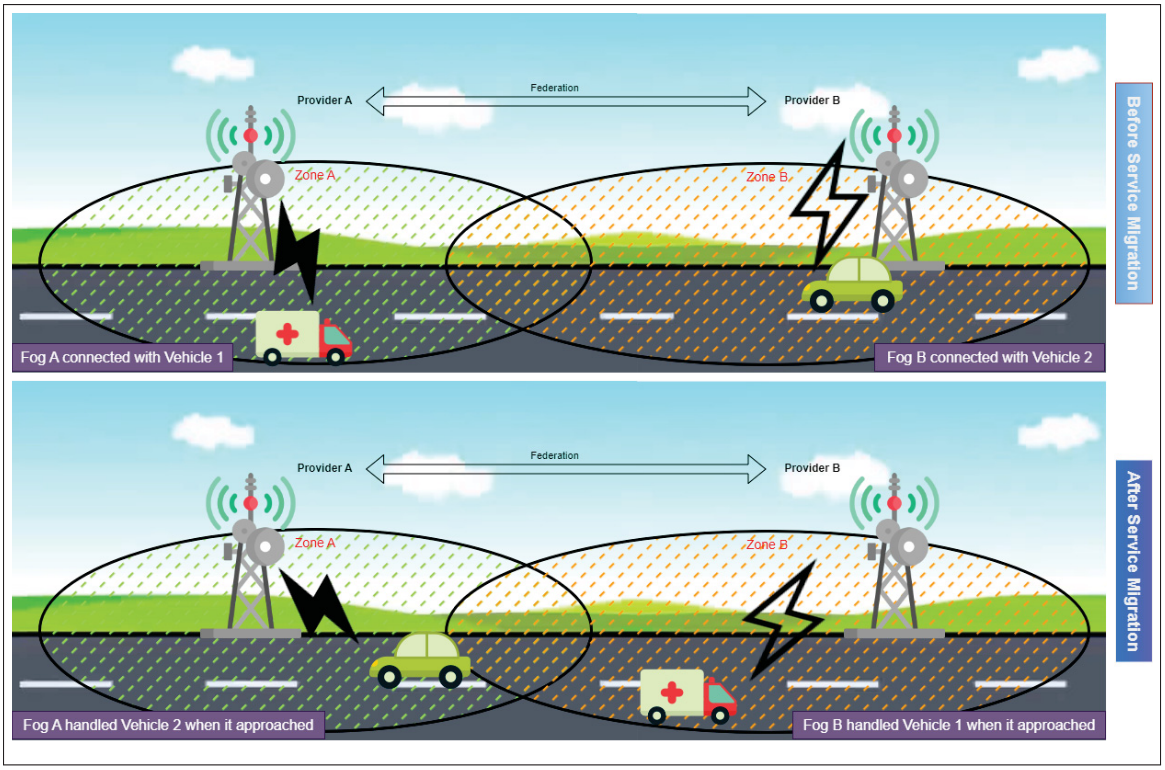


Figure 5.1 Fog federations interchanging services in IoV

driving experience, road safety, and overall efficiency of Intelligent Transportation Systems (ITS) that rely on IoV applications such as collision avoidance and object detection.

Hence, it becomes an essential aspect to exclude malicious providers from the formed federations as harm the network (Wahab, Bentahar, Otrok & Mourad, 2015; Moati, Otrok, Mourad & Robert, 2014). To resolve such a problem, an important factor must be considered when seeking a formation; the reputation of the fog provider. As rational entities, fog providers are willing to cooperate with others that are well-behaving in order to yield a higher payoff. The reputation of a provider can be obtained through historical interactions. This way, a provider can build a preference table to decide with whom to cooperate for doing a certain job. Nevertheless, managing the reputation in a centralized or local manner is not the optimal way for organizing federations (Yu, Zang & Reagor, 2007). A centralized solution would first entail points of failure, lack of transparency, and inflexibility. In a local manner, each node would individually store the reputation of all other members following interactions. This means that nodes are not aware of the reputations assigned by other nodes. In other words, in order to avoid interacting with a particular malicious provider p_m among all providers, the remaining providers would need to engage with it at least once to classify it as untrustworthy. Consequently, the assessment of provider p_n towards p_m is not taken into account when provider p_o interacts with p_m.

Blockchain is an open and distributed ledger that records transactions between entities in a verifiable and permanent way (Zheng, Xie, Dai, Chen & Wang, 2018). Instead of relying on a centralized entity, Blockchain allows the storage of transactions to take place on multiple entities. Tampering with records from the ledger is still considered impossible with today's computers due to the complexity of the process. In addition, the source of the transaction can be traced back to its original owner. Furthermore, the Ethereum Blockchain allows the integration of smart contracts, where programs can be executed in a fully trusted and automated manner without any human interaction (Khan, Loukil, Ghedira-Guegan, Benkhelifa & Bani-Hani, 2021). In this work, we leverage Ethereum Blockchain technology as an enabler to form reputable fog federations. Unlike the local approach mentioned earlier, where reputation information is stored locally by each node, our proposed approach utilizes the inherent properties of Blockchain to enable decentralized decision-making and reputation management, while making reputation information available across the entire network. We employ a Hedonic game theoretical model to allow decentralized decision-making when establishing federations that are based on their preferences (Banerjee, Konishi & Sönmez, 2001). Game Theory proved itself to be a useful technique when it comes to forming stable clusters due to its way to model the strategic behavior of self-interested agents. In addition, we reinforce the formation with an intelligent feedback-based trust establishment mechanism that allows providers to rate the behavior of the other members in their federations through smart contracts to limit the impact of biased feedback. Furthermore, we penalize misbehaving providers by excluding them from the formation game when they fell below a certain reputation threshold. To prepare for our testbed, we rely on EUA dataset[1] for fog location placement in a certain area, along with vehicular traffic generated by SUMO (Simulation of Urban MObility) in that area. In addition, we use Solidity and Python to program our on-chain

[1] https://github.com/swinedge/eua-dataset

and off-chain operations, respectively. Results show that our approach is cost-effective and can yield an increased QoS and profit while reducing the number of misbehaving nodes in the environment when compared to other works in the literature. The summarized contributions are as follows:

- Devising a Blockchain-tailored Hedonic game formation scheme for establishing stable fog federations while considering user-fog interaction.
- Deploying smart contracts on the chain to enable fog providers' reputation calculation based on federation members' feedback.
- Developing the formation scheme and extensively evaluating its effectiveness with simulations and results in IoV environments.

The remainder of this paper is organized as follows. In Sec. II, we present the related work in the context of fog federation formation. We propose our Blockchain-based framework in Sec. III. The details of our solution implementation and the numerical results are discussed in Sec. IV. Finally, we summarize the findings of this paper and highlight its possible extension in Sec. V.

5.3 Related Work

In this section, we explore recent efforts focused on optimizing fog federations and their formation mechanisms. Additionally, we highlight the emerging trend of utilizing blockchain technology for resource-based endeavors. As far as we know, there were limited efforts addressing the providers' reputation when forming federations, thus we do not limit our related work findings strictly to fog federations.

5.3.1 Coalition-based Solutions

The authors of (Veillon *et al.*, 2019a) devise a technique for reducing latency in video streaming applications, especially for end-users who are far location-wise, through fog delivery network

Table 5.1 Related Work Summary

Work	**Coalitions**	**Utility**	**Reputation**	**Formation Method**	**Secure Formation**
(Niyato *et al.*, 2011)	Cloud	Revenue	False	Stochastic LP	False
(Wahab *et al.*, 2016b)	Services	Trust	True	Hedonic	False
(Wahab *et al.*, 2016a)	Services	Revenue	True	Stackelberg	False
(Anglano *et al.*, 2018a)	Fog	QoS	False	Hedonic	False
(Smirnov *et al.*, 2019)	Robots	# of Tasks Completed	False	Fuzzy coop. game	True
(Hammoud *et al.*, 2021)	Fog	Revenue	False	Evolutionary	False
This Work	Fog	QoS + Reputation	True	Hedonic	True

federations. Their technique works by pre-processing video streams that are trending in a certain region. Specifically, since fog is limited in resources, they suggested pre-processing only the popular parts of the video and the remaining parts to be processed on-demand. Furthermore, to limit the on-demand processing, they reuse preprocessed video data available at the neighboring provider, thus establishing coalitions. In (Anglano *et al.*, 2018a), the authors present a distributed game model approach to form fog coalitions where the fog providers rely on their own preferences to select which coalition to join, in order to maximize their individual monetary profit resulting from the formed coalitions. They claim that their approach yields stable and profitable coalitions. However, their main objective is to address the profit of the participants and ignore the user-fog interactions (i.e. QoS). Additionally, they neglect the reputation of the fog providers and their effects on the formed coalitions. The authors in (Shamseddine *et al.*, 2020b) advance a novel federated fog architecture with a formation based on merging both the genetic algorithm with machine learning. They claim to obtain overall good results while increasing the percentage of satisfied end users. Nonetheless, the reputation of the providers was not considered in their approach. In (Khosrowshahi-Asl *et al.*, 2020), the authors propose a strategic Distributed Decision-making Mechanism for forming communities of cloud providers under incomplete information. In (Gu, Tang, Jiang & Jia, 2020), the authors propose a resource allocation scheme for fog providers based on a reputation mechanism. Their approach consists of calculating the reputation of the provider by taking into consideration its internal reputation, direct reputation, and indirect reputation factors. In their work, the community's reputation is stored locally on the provider's level and the information is forwarded upon request, which requires time and resources to calculate the final reputation. The authors in (Niyato *et al.*, 2011) develop a stochastic linear programming game model to model the uncertainty of the fog providers while forming coalitions. They claim to reach stable coalitions and increase the profit of the providers. In (Wahab *et al.*, 2016b), the authors propose a Hedonic game to orchestrate the formation of cloud coalitions in a reputable manner. Nevertheless, like most of the cloud coalition formation schemes, the authors neglect the user-server QoS metrics due to the fact that the formation is on the cloud level. In addition, their trust mechanism, similar to (Gu *et al.*, 2020), is calculated by forwarding the feedback from one node to another, which may cause inaccuracies especially when the middle node is malicious. In (Wahab *et al.*, 2016a), the authors address the competition among Web services in the cloud computing market and propose a cooperative model using a distributed Stackelberg game where all services are totally autonomous in making their decisions. The proposed model is able to increase the satisfaction of the Web service agents and the users. Nevertheless, there exist several challenges in that work, especially when applied in the fog context. For instance, in the fog computing paradigm, the providers are limited to relatively small geographical areas, thus, forming leaders and followers based on reputation metrics would be very challenging, whereas in cloud services there are preexisting giants as the authors stated (e.g., Amazon, Google, eBay, etc.). Moreover, the calculation for the reputation metric lacks consensus and its scalability remains questionable. The authors in (Smirnov *et al.*, 2019) propose a dynamic formation approach to form coalitions of autonomous robots based on the integration of a game model with smart contracts. Their objective is to maximize the efficiency of the work. Their work is based on a negotiation scheme but it neglects the reputation of the players. Some works propose tackling each of the problems (i.e., network delays and reputation) individually without a coalition-based framework. To resolve the delays, (Iqbal, Malik, Rahman & Noor, 2020) advance a reputation-based mechanism running on top of the Blockchain. In their work, roadside units are assumed to offload tasks to nearby fog vehicles. Nevertheless, nowadays vehicles are still not capable of handling critical external tasks, which makes the applicability of their method doubtful. Apart from the reputation of the providers, some works studied the trustworthiness of the users. For instance, the authors of (Hussain *et al.*, 2020) propose a trust feedback system to evaluate the users' trustworthiness in the fog-IoT paradigm. In our previous work (Hammoud *et al.*, 2021), we consider forming stable federated fog providers based on evolutionary game theory. The main focus of the latter work is to resolve the instabilities of the providers. The formation, however, neglects the reputation and the security of the environment. The related work summary presented in Table 5.2 indicates that a secure and efficient architecture for assisting fog federation formation has not been yet proposed and that the remainder of this paper is addressing this issue.

5.3.2 Blockchain-based Solutions

Several works have explored the integration of Blockchain technology in the domains of cloud computing to address various challenges related to security, resource management, and decentralized decision-making. For instance, the authors in (Taghavi, Bentahar, Otrok & Bakhtiyari, 2018) introduce Cloudchain, a blockchain-based cloud federation for enabling resource trading among cloud service providers. They utilize smart contracts on the Ethereum network to create a fully distributed structure, allowing cloud providers to engage in coopetitions through a differential game. While the paper emphasizes the coopetition aspect among cloud providers within the Cloudchain, it does not elaborate on the specific mechanisms used to form federations initially or how cloud providers decide to join or leave a federation. In (Taghavi, Bentahar, Otrok & Bakhtiyari, 2019), the authors propose a blockchain-based model with quality verification. They introduce an oracle as a verifier agent to monitor the quality of service provided by cloud providers and report to smart contracts on the blockchain. By employing a Stackelberg differential game, their model achieves optimized cost for using the verifier agent and maximized profit for the involved providers. In the domain of federated learning (FL), (ur Rehman, Salah, Damiani & Svetinovic, 2020) presents the concept of fine-grained FL on edge servers. The authors emphasize the importance of personalization, decentralization, and trust in FL systems and introduced a blockchain-based reputation-aware approach to ensure trustworthy collaborative training in mobile edge computing. Nevertheless, none of these aforementioned works specifically address the challenges of forming reputable fog federations in the context of IoV. In contrast, our proposed approach leverages Blockchain and game theory to establish decentralized fog federations while considering user-fog interactions and reputation management.

5.4 Proposed Framework

In this section, we present our framework that addresses forming reputable coalitions in the highly dynamic IoV environment. First of all, we demonstrate the architecture and list the components and their roles. Afterward, we define the variables and the system model. Then, we discuss the off-chain game formation technique used in our framework and the on-chain reputation module.

5.4.1 Architecture and Components

As previously implied, the main components of our architecture, which are depicted in Fig. 1, can be summarized in the following:

- IoV applications are the main applications for enabling smooth IoV integration in smart cities. In general, any service that a vehicle can use is considered an IoV application, e.g. object detection, collision alert, traffic sign recognition (TSR), etc...
- End-users are the parties that need to use external resources to process their tasks. Usually, the tasks are related to the applications they are subscribed to. In our work, we are addressing vehicular nodes as the end nodes. Thus, we are taking into consideration their dynamicity in moving from one place to another at different timestamps. It is worth mentioning that the architecture still holds even when the users are standing still, i.e. velocity = 0, which makes it applicable to IoT environments as well.

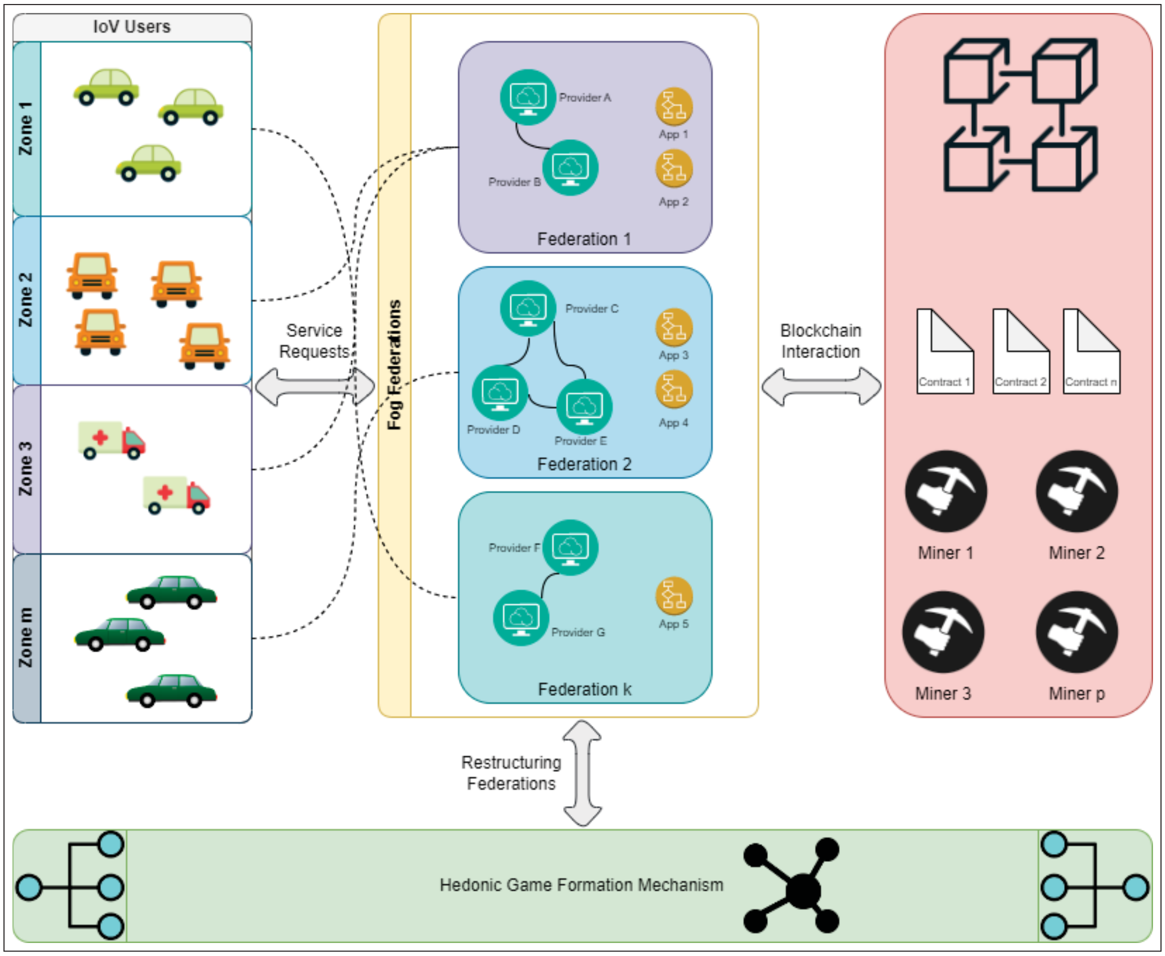


Figure 5.2   Reputation-Based Fog Coalitions Architecture for IoV

- Fog servers are the nodes situated near the end-users that await to receive a task and process it. IoV applications are mainly deployed on fog servers as services to be offered to the end users. Aside from its computational features, the main characteristic of this component is its geographical location and its coverage in which it can interact with certain end-users within a particular range.

- Fog providers are the owners of the fog servers deployed near the edge. Mainly, a fog provider can own more than one fog server. They manage their resources and interact with IoV application creators to deploy their services on their nodes in exchange for monetary return. Aside from its resources, a fog provider is characterized by a reputation to define how well-behaving they are when dealing with service-level agreements.

- Fog federation is a coalition that can be defined as a set of fog providers seeking to share their resources into one pool in order to increase the geographical footprints and enhance the service quality when serving end-users.
- Formation mechanism: Fog coalitions can be formed in many ways. In our system, we optimize the formation procedure by periodically reinforcing the stability of the coalitions in a dynamic manner in order to fit the IoV environment.
- Smart contracts are small applications deployed on the Blockchain for automatic execution once their conditions are met. We rely on smart contracts to enhance the formation mechanism of the fog coalitions

5.4.2 System Model

To maintain a decent computing and networking infrastructure provided by fog providers in IoV, we model the interactions among the entities of our system as follows. Initially, there are $V = v_1, v_2, ..., v_n$ end-users, i.e. vehicles, each subscribed to a set of services (applications) A_{v_i}. We assume the existence of m applications, $A = a_1, a_2, ..., a_m$. At any given time, a user may request to access any of its applications and also may request additional resources from the supporting infrastructure to speed up the processing time. The fog provider $p_i \in P$ is presumed to own one or more fog nodes (servers) S_{p_i}, each characterized by a certain range that it can potentially serve the users within and its dedicated bandwidth. These servers are set to deploy the applications so users can access them. The connection state $\epsilon^{v_k}_{a_i,s_j}$ models the interaction between vehicle k and service i through fog node j, as follows:

$$\epsilon^{v_k}_{a_i,s_j} = \begin{cases} 1, & \text{if the fog node } s_j \text{ can handle the connection} \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

The main factors defining the connection state are: (1) the user is within the coverage area of the fog server, and (2) the server has the physical resources to serve the request. In a similar manner, handling the connection by a provider can be modeled by the variable $\zeta^{v_k}_{a_i,p_j}$ defined as:

$$\zeta^{v_k}_{a_i,p_j} = \begin{cases} 1, & \text{if } \epsilon^{v_k}_{a_i,s_l} = 1 \text{ and } s_l \in \mathrm{S}_{p_j} \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

Maximizing the success rate of the connections can be a challenge, especially with the dynamicity of the end users. Therefore, we consider a coalitional-based solution for service deployment and task offloading. The coalition Λ_i between two or more fog providers means that these providers agree to utilize each other's resources to migrate/deploy services. To illustrate, the provider p_j, that offers service s_k, can establish a cooperation agreement with another provider p_l. Such cooperation can be referred to as $\Lambda_i = p_j, p_l$, and s_k in this case can be migrated/replicated into any of the members' servers upon demand if the available resources allow it. With the cooperation in hand, we introduce variable $\kappa^{v_k}_{a_i,\Lambda_j}$ such that:

$$\kappa^{v_k}_{a_i,\Lambda_j} = \begin{cases} 1, & \text{if } \zeta^{v_k}_{a_i,p_l} = 1 \text{ and } p_l \in \Lambda_j \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

To fulfill the dynamicity requirements of IoV, a maximization of variable $\kappa^{v_k}_{a_i,\Lambda_j}$ should be investigated for $v_k \in V$ and $a_i \in A_{v_k}$. It is worth mentioning that we are studying the IoV environment in real-time, therefore, all decisions are being made for a time t.

One important factor also to consider is the behavior of the fog provider itself. A well-behaving fog provider fulfills its tasks by faithfully allocating its available resources to serve the tasks. Such a provider is considered trustworthy among the fog provider community and, therefore, can be reliable when interacting within a coalition. In parallel, there can be an untruthful fog provider who is not willing to dedicate the agreed-upon resources and decides to save a chunk of the resources, or the whole, to serve another task simultaneously. To quantify this issue we can define the honesty metric h_{p_i} that measures the cooperation of a provider i within its allocated coalition. h_{p_i} has a decimal value bounded by 0 and 1, where $h_{p_i} = 1$ implies that p_i is fully cooperative, whereas $h_{p_i} = 0$ implies the non abidance of provider i within the coalition in terms of resource allocation. However, the honesty of the provider is not a metric that can be easily measured by others. Therefore, we apply metric $r_{p_i} \mid 0 \leq r_{p_i} \leq 1$ that represents the reputation of the provider within the fog community, which indicates the aggregation of others' opinions towards provider i's honesty. Hence, the success rate of handling connection gets uncertain due to the fact that a fog server may or may not be willing to cooperate, which is directly related to its reputation.

The problem of optimally forming the coalitions is an NP-hard problem. Therefore, we model it as a Hedonic game model and integrate the preferences of the players when deciding who to interact with. The details of the game are given in the next section.

5.4.3 Reputable Hedonic Fog Federation Formation Scheme

A coalitional game model is a sequence of decisions made by the players in order to maximize their utilities. A Hedonic game is a special type of coalitional game that abides by the following rules: the player's utility can be decided solely by the players within the considered coalition, and the coalitions are formed according to the preference function of the players interacting (Hammoud *et al.*, 2022b). In this paper, the fog coalition (federation) formation game models the interaction between the fog providers and the coalitions which they can join according to their preferences. The utility of a provider to whether or not join a coalition reflects the preference of the player to which coalition it is willing to join by assigning a bigger weight for it, and is defined as follows:

$$U_{p_i}^{\Lambda_j} = \sum_{a_k}^{A^{p_i}} \sum_{v_l}^{V^{a_k}} \kappa_{a_k,\Lambda_j}^{v_l} \times r_{\Lambda_j} \tag{5.4}$$

where A^{p_i} is the set of services hosted by provider i, V^{a_k} is the set of vehicles subscribed to service a_k, and r_{Λ_j} is the average reputation of coalition j. This utility function can determine the rational decision of any player when choosing a coalition to join. The preferences can be translated into the following equation:

$$\Lambda_j >_{p_i} \Lambda_k \iff \varrho_{p_i}(\Lambda_j) > \varrho_{p_i}(\Lambda_k) \tag{5.5}$$

where $>_{p_i}$ is the preference relation that reflects the decision of the service provider when facing two distinct coalitions. To elaborate, we assume the existence of two coalitions, j and k, and fog provider i deciding which coalition to join due to its resource shortage to handle its requests. Assuming that coalition j might be richer in terms of resources, p_i can still choose coalition k if it offers relatively better reliability (i.e., utility) when compared to coalition j. According to the game detailed above, we can conclude that this game is Hedonic since both of its conditions are valid when it comes to constructing the coalitions. In coalitional game models, we are more interested in studying the coherence of the players when interacting in communities. One of the important properties is the stability of the community. A community is stable if every player is satisfied with its current coalition. In other words, there is no player that can find a better decision (utility) other than the one it is currently inheriting

$$\forall p_i \in P, \Lambda_j \geq_{p_i} \Lambda_k \mid p_i \in \Lambda_j \ \ \& \ \ k \neq j \ \ \& \ \ 1 \leq k \leq m \tag{5.6}$$

Providing this condition for all fog providers in the system yields a stable environment as long as there are no significant changes in terms of users' displacements, and in terms of providers' reputations.

5.4.4 Off-Chain Hedonic Algorithm Execution

To resolve the above equations and stabilize the infrastructure, we devise a decentralized algorithm that can be executed by the providers at each time t, or whenever the overall QoS falls below a certain threshold. The main reasons behind shifting this process to be executed by the providers rather than the Blockchain are due to it (1) being a provider-centered operation and (2) being expensive in terms of computation and network as preferences games require continuous input from the players to achieve a stable solution that satisfies all involved parties. Algorithm 5.1 reflects solving Eq. 5.6 by executing it at each time t for all the providers inside the system. Before initiating the start of this algorithm at time t, it takes the status of the infrastructure in terms of federation structure up until time t as input, i.e. F^t. The variable F^t represents the current state of the federation formations at time t. It contains information about

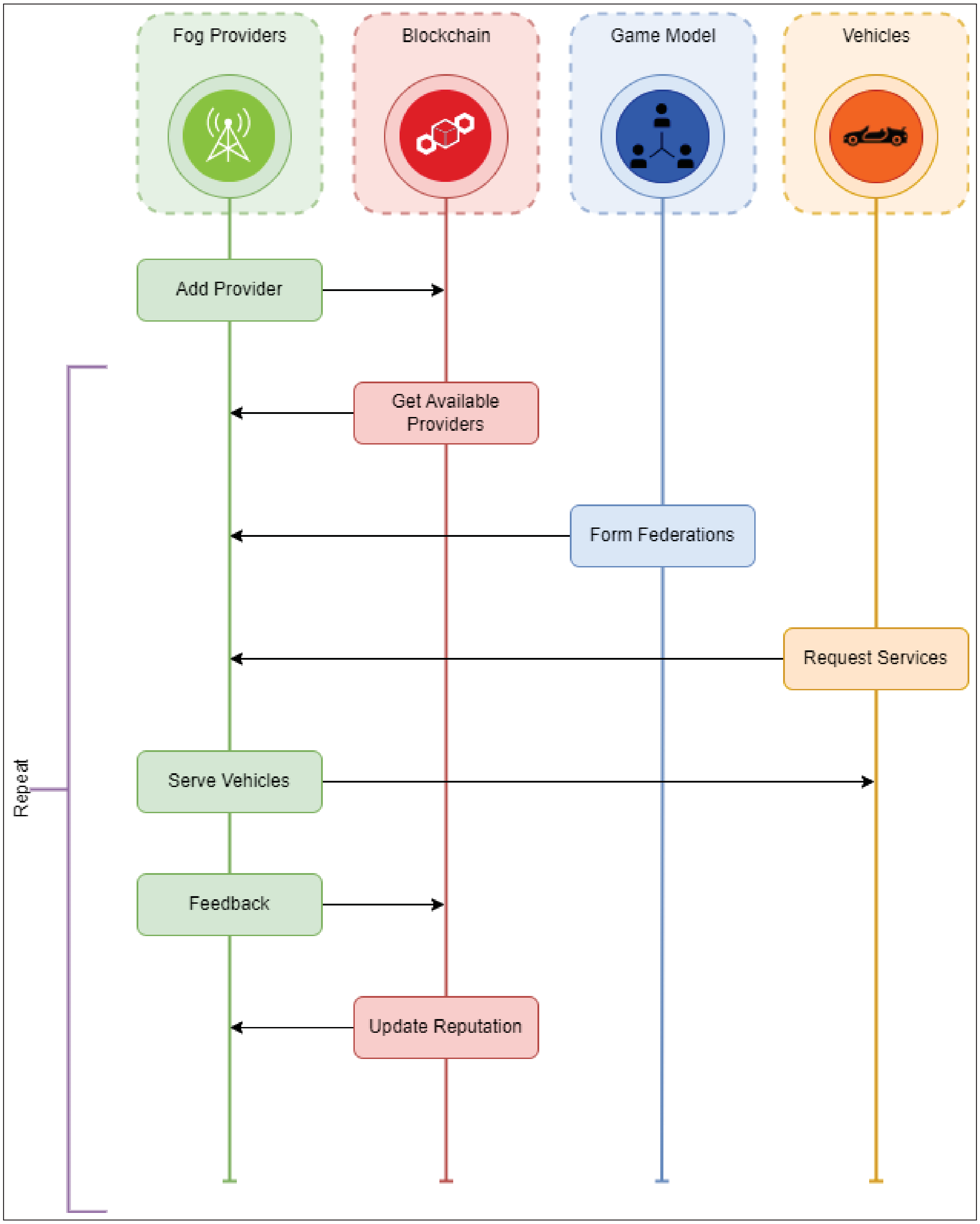


Figure 5.3 Reputable Fog Federation Formation Scheme: Process Timeline

the current coalitions and their members. The result of this algorithm is the most fitting set of federations to be formed at time $t+1$. F^* is the variable introduced here to store the previous federation formation, F^{t+1} is the result of this algorithm, and L is the temporary variable that

Algorithm 5.1 Federation Formation Mechanism

Input: F^t
Output: F^{t+1}
2 $F^* \leftarrow F^t$;
4 $F^{t+1} \leftarrow \emptyset$;
6 $L \leftarrow \emptyset$;
8 **while** $F^* \neq F^{t+1}$ **do**
10     $F^{t+1} \leftarrow F^*$;
12   **foreach** $f_i \in F^*$ **do**
14   **foreach** $fp_j \in f_i$ **do**
16   **if** $fp_j \notin L$ **then**
18                 $Curr \leftarrow U_{p_i}^{\Lambda_j}$;
20   **foreach** $f_k \in F^*$ **do**
22                     $New \leftarrow U_{p_i}^{\Lambda_j}$;
24   **if** $New > Curr$ **then**
26                         $f_i \leftarrow f_i \setminus \{fp_j\}$;
28                         $f_k \leftarrow f_k \cup \{fp_j\}$;
30                         $Curr \leftarrow New$;
31   **end if**
32   **end foreach**
34                 $L \leftarrow \{fp_j\} \cup L$;
35   **end if**
36   **end foreach**
37   **end foreach**
39     $L \leftarrow \emptyset$;
40 **end while**
42 return F^{t+1}

indicates the players who made at least one decision for the current round. The while statement in Line 4 indicates that the algorithm only stops when no further changes in the structure take place. The decision of whether or not a player joins a certain federation solely depends on the preference function of all the available federations to join. Even though this process is being executed off-chain, however, it does not affect the reliability of the proposed framework as each player/provider is responsible for its own decision.

5.4.5 Contract-Based Reputation Module

In order to assess the cooperation of a fog provider when he is integrated within a coalition, we rely on their reputation to decide how convenient it is for others to cooperate with it. We rely on a feedback-based trust establishment mechanism to estimate a proper reputation for the providers within the community. In the beginning, each provider is expected to be fairly behaving with an initial reputation value equal to a predefined default value, $reputation\_DEFAULT$, once it starts interacting in the system. When the formed coalitions are ready to be decomposed, each provider is supposed to submit feedback about the other members within the same coalition that is based on the cooperation of the latter. An aggregation technique takes place for a fair trust assessment and, accordingly, a provider is tagged with a new reputation value. A provider with a low reputation value may be penalized by a temporary ban from joining other coalitions.

To carry out the above in a non-tampered manner, we invest in a Blockchain-based solution empowered by smart contracts built on top of the Ethereum network. Blockchain is a decentralized ledger system that stores information in a way that cannot be tampered with. The ledger is maintained by multiple entities and thus the availability of the records is considered reliable when compared to other centralized systems. Ethereum[2], the second biggest cryptocurrency after Bitcoin[3], is a platform empowered by Blockchain to create decentralized applications. It features smart contracts which allow predefined programs to execute on the Blockchain peers in a verifiable way. Therefore, executing certain algorithms on smart contracts can assure that the results obtained are not tampered with by other parties. To leverage the power of Blockchain 2.0, we rely on implementing two smart contracts, *Provider Contract* and *Reputation Contract*. Provider Contract is a smart contract that keeps track of the providers in the system and allows them to retrieve information about other providers including their reputation. A reputation Contract is a contract to manipulate the reputation of the providers after being assigned a task. We detail both contracts below:

---

[2] https://ethereum.org/

[3] https://bitcoin.org/

5.4.5.1 Provider Contract Functions

Table 5.2 Provider Contract Functions

function add_provider() public	-
function update_reputation(uint provider_id, uint rep) external	-
function ban_provider(uint provider_id, uint until) external	-
function get_available_providers() public view	returns(Provider[] memory providers)

First of all, we devise a contract to maintain the fog providers that register in the system. This contract has multiple functionalities, as can be seen in Table 5.2. For instance, the function *add_provider()* takes the necessary initiates for adding the caller provider, i.e. the one who calls the function, into the system. It initializes the default variables to store the provider in the system, in addition to its Blockchain address. *get_available_providers()* is a function that returns the providers who are registered in the system that are (1) available in terms of resources and (2) not banned due to malpractices. *update_reputation()* and *ban_provider()* are functions that can be called by an external contract when necessary. The first one changes the provider's old reputation at time t to a new reputation that fits at time t. The second one penalizes the provider by banning them from participating in the coalition formation game for a predefined period of time.

5.4.5.2 Coalition Contract Functions

Table 5.3 Coalition Contract Functions

function create_coalition(address[] memory providers)public	-
function feedback(uint[] memory feedback) public	-
function append_feedback() private	-

As expected, this contract handles the federations' registration on the chain and receives trust values from the federations' members. *create_coalition()* function receives the addresses of the providers as input and it creates a coalition denoting the members after the off-chain game model yields the result. Only one member needs to execute this function to establish the coalition. When the tasks come to an end, the providers within a single coalition are expected to rate the other members by evaluating their cooperation as a trust value by relying on the function *feedback()*. For the provider i to evaluate the other members of coalition Λ_j, it provides an array of size $n-1$, where $n = |\Lambda_i|$, that contains the reputation metrics of the other members and how they were handling the offloaded tasks. The last member of the same coalition to send its feedback, i.e. the n^{th} member, triggers an automatic call for calculating the aggregated trust value by calling the function *append_feedback()*. *append_feedback()* embeds an aggregation technique of an arithmetic mean mechanism. Each coalition holds a matrix of size $n \times n$ that stores the evaluation metrics received from the members. Once the aggregated value is calculated, it triggers the *update_reputation()* function inside the *Provider contract* in order to update the reputation of the providers.

The historical reputations of the providers are taken into consideration when calculating the new reputation by assigning weights to the old and new reputation, and adding them together, as follows:

$$r^t_{p_i} = \alpha \times r^{t-1}_{p_i} + (1-\alpha) \times \dot{r}^t_{p_i} \tag{5.7}$$

where $\dot{r}^t_{p_i}$ is p_i's aggregated reputation from the $append\_feedback()$ function.

Fig. 5.3 summarizes the interactions among modules in our proposed system.

5.4.6 Security Analysis

In this section, we conduct a security analysis of our fog federation framework to evaluate its resilience against misbehaving providers. We identify potential threats and present the mechanisms put in place to mitigate these risks.

5.4.6.1 Misbehaving Providers

Misbehaving providers pose a significant challenge to the stability and efficiency of the fog federation. We considered various forms of misbehavior, including SLA violations and non-cooperation within coalitions. As providers may fail to meet their agreed-upon service-level agreements with end-users, this can lead to a degraded user experience. We addressed this issue through the reputation module based on feedback from other federation members. The low reputation of members can result in diminished opportunities to join federations or potential temporary bans.

5.4.6.2 Detection and Mitigation

To detect and mitigate malicious behaviors, our framework employs several mechanisms: **Consensus Mechanisms:** Our blockchain-based approach utilizes consensus mechanisms to ensure agreement on the validity of transactions and coalition formations. **Smart Contract Enforcement:** Smart contracts enforce the rules and penalties defined for providers' behavior, such as temporary bans or reputation adjustments. The *feedback* and *banning* actions are core functions of the Blockchain contracts of our approach. **Feedback Aggregation:** The reputation module aggregates feedback from coalition members to calculate a provider's reputation. The use of multiple data points ensures a fair assessment of a provider's behavior and minimizes the impact of individual biased opinions.

5.5 Experimental Evaluation

In this section, we explain our setup and analyze the results of our implemented framework while comparing it to the benchmark.

5.5.1 Setup

To set up the environment, we rely on several applications, datasets, and other metrics combined, as follows:

5.5.1.1 Tools

All simulations were conducted on a computer equipped with 12th Gen Intel(R) Core(TM) i7-1260P, with memory up to 16384MB. The following software was used:

- **PyCharm:** in order to execute the off-chain coalition formation, we modeled the Hedonic game model using Python language.
- **SUMO (Simulation of Urban MObility) :** is an open-source microscopic traffic simulation software. We use it here for simulating traffic in IoV.
- **Ethereum Remix:** remix IDE allows developing, deploying, and administering smart contracts for Ethereum. It can be used directly through its online web interface. Remix was used to program the smart contracts related to the providers and coalitions. We assign an address to each provider in order to join coalitions.

5.5.1.2 Datasets

The following datasets were used to populate our environment.

- **Fog Nodes:** to conduct the experiments using realistic geographical areas due to the nature of IoV, we use EUA that contains the coordinates of the fog devices based in Australia.
- **Providers:** due to the lack of fog providers datasets, we randomly assign providers to own one or more nodes from the list of fog nodes in the EUA dataset. We set up 10 fog providers in the system.
- **Vehicles:** we generate vehicular traffic using SUMO software. We set up the simulator to export the traffic in Australian terrains. The number of vehicles in the system is set to 500.

5.5.1.3 Misc. Provider Mechanics

There are multiple methods and attributes that have been used throughout this simulation in order to be conducted.

- **Reputation:** is the value that represents how cooperative and trustworthy a provider is. This value is well-known to the community.

- **Honesty:** is the provider's actual percentage of cooperation and trustworthiness. This value is private.
- **Credibility:** is how credible a provider is when submitting feedback about another provider. The higher the credibility, the better the feedback accuracy toward the other provider. This parameter is private to each provider.
- **Feedback:** at the end of each interaction between providers, each provider estimates the reputation of the other members in the federations.

5.5.1.4 Benchmark Model

- **Trust-based Service Communities(Wahab *et al.*, 2016b):** this approach is based on clustering and forwarding reputation information about others from a provider to its neighbors. In other words, nodes transfer their knowledge to neighboring nodes without relying on a central entity or Blockchain to manage the process.
- **Anglano et al. (Anglano *et al.*, 2018a):** this is a game model approach to form fog coalitions where the fog providers rely on their own preferences to select which coalition to join, in order to maximize their individual monetary profit resulting from the formed coalitions.

5.5.1.5 Comparison Metrics

The analyzed system is evaluated in terms of 4 metrics:

- **Reputation prediction accuracy:** it shows the average reputation accuracy predicted by all providers towards each other. This metric is important because federations are formed based on it, in addition to other metrics discussed in the methodology section. We exclude the work of (Anglano *et al.*, 2018a) from this metric due to its irrelevance in such a case.

- **Service availability:** the availability percentage of the resources when requested by the end users.
- **Serviced user rate:** the rate of the users who can be serviced with a decent QoS when communicating with the fog servers.
- **Average provider payoff:** the reward of the provider after completing the requested tasks. Similar to (Hammoud *et al.*, 2021), we assume a fair monetary distribution among the federation participants.

5.5.2 Results

Before detailing the performance of our proposed framework, it is worth mentioning that deploying smart contracts on the Ethereum Blockchain costs "gas". Gas is paid in Ether (ETH), the native cryptocurrency of Ethereum, and the amount is related to the complexity of the contract. In our implementation, deploying the *Provider Contract Function* on the chain costs 0.000775827 ETH, which is equivalent to 1.47 USD. Similarly, deploying the *Coalition Contract Function* costs 0.001067240 ETH, which is equivalent to 2.02 USD[4].

We run the experiments 10 times, each time for different vehicular trajectories, and then we averaged the results to display them in figures. The performance of our approach is evaluated against the benchmark models using four metrics: reputation prediction accuracy (exclusively for our approach and the Trust-based Service Communities approach), service availability, serviced user rate, and average provider payoff. Fig. 5.4 shows the average reputation accuracy predicted by the whole population towards other providers. The Blockchain-based approach starts at 93.3% and increases to reach 95.5% at the end of the simulation. At the same time, the Trust-based Service Communities benchmark model starts at 85% and maintains a slow average decrease until 81.7% at the end of the 40^{th} round. The main reason behind such a gap between the two curves is the fact that our mechanism relies on a Blockchain-based storage mechanism where the reputations of the fog providers at time t are all stored on-chain and accessible to the public

[4] The value is based on the conversion rate of the following date: 17-08-2022, where 1 ETH was equal to 1884.94 USD

of providers which is essential when predicting the reputation for time $t + 1$, while the other approach relies on forwarding reputation which can be outdated, thus, inconvenient for basing the prediction upon.

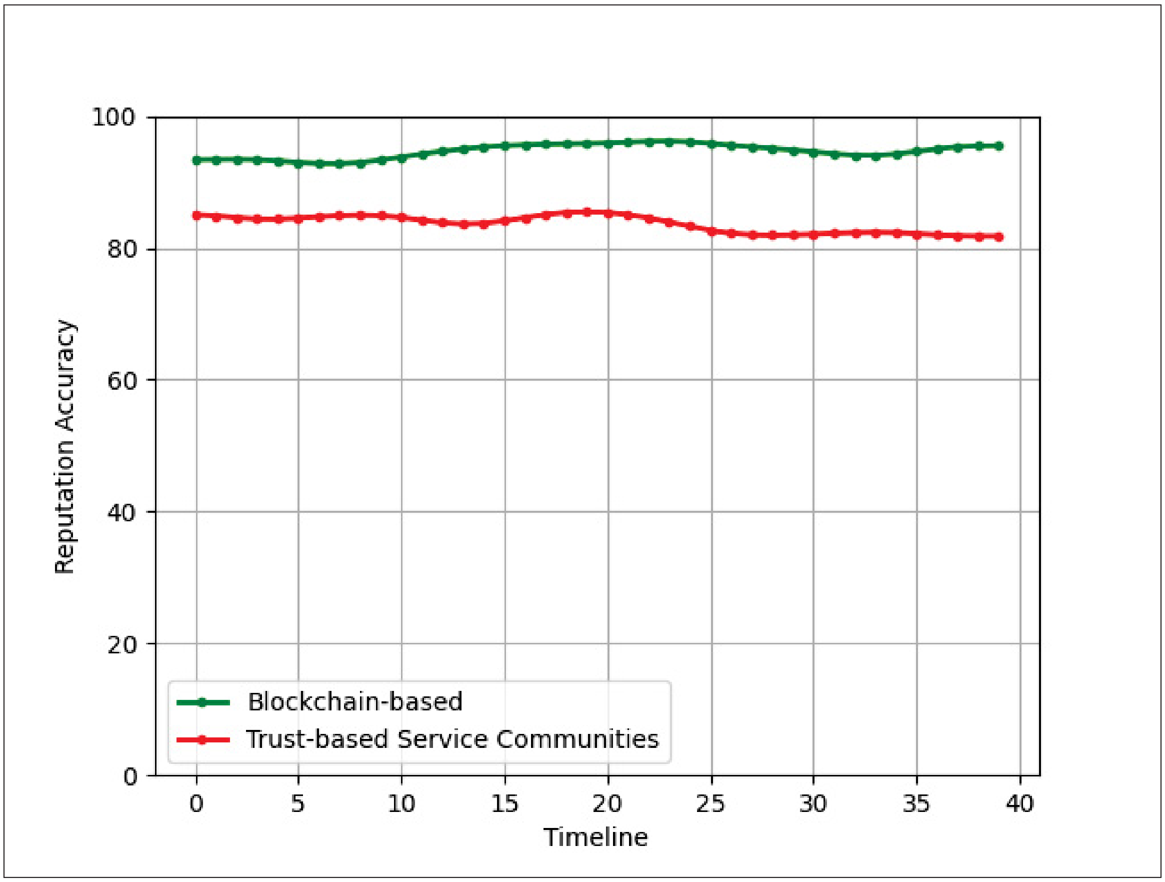


Figure 5.4 Reputation Prediction Accuracy

Furthermore, we assess the service availability of the fog providers when they are receiving requests from the vehicles in Fig. 5.5. At the beginning of the simulation, both the Blockchain-based and Trust-based models were able to achieve 84.4% of full-service availability. However, our model increases the availability to reach 87.1% at the end of the simulation whereas the other model fluctuates and decreases the availability to reach 80.4% instead. The Anglano et al. model failed to cope with its peers due to its incapability of dealing with malicious providers. The main reason behind the difference between the former two is the early detection of dishonest providers and avoiding them according to the preference function in our model. On the other hand, the Trust-based model suffers from late detection of the misbehaving providers which leads to immature decision-making for the other providers.

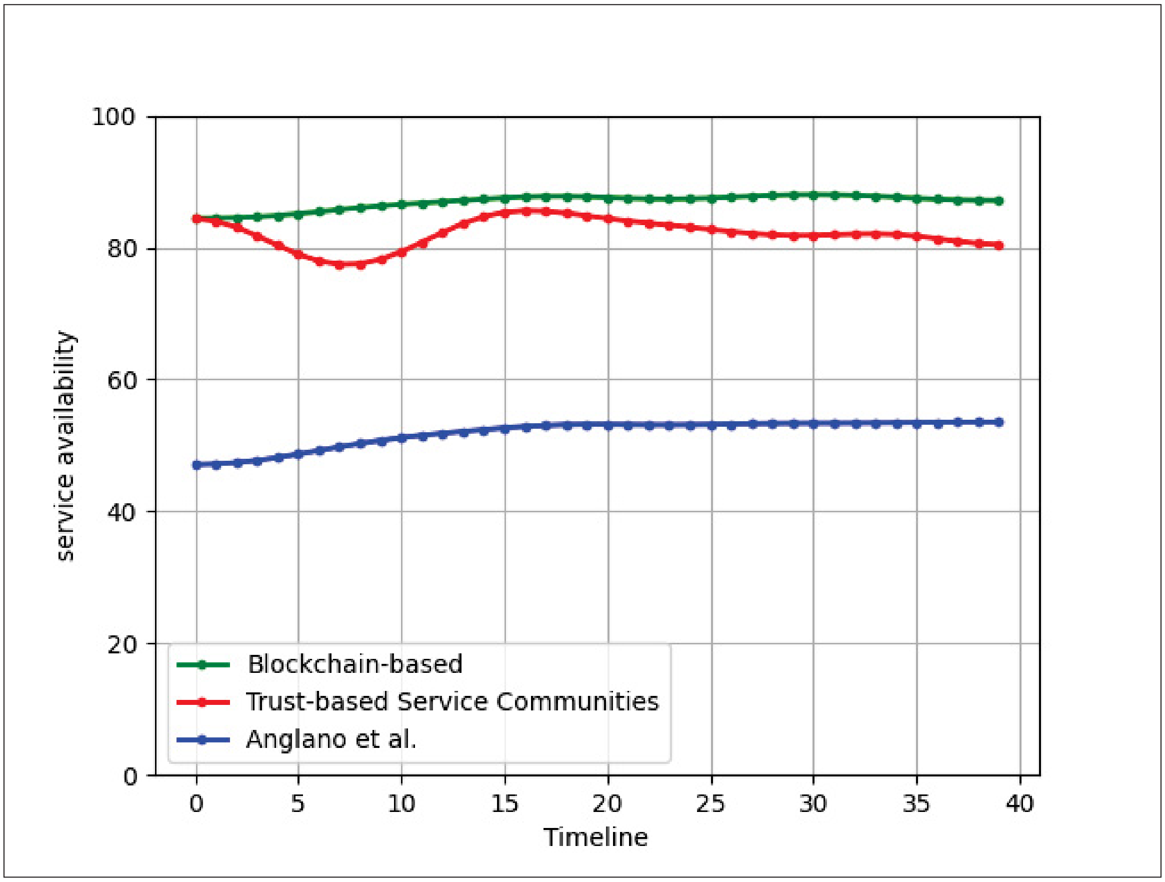


Figure 5.5 Service Availability

In Fig. 5.6, we display the serviced user rate. It can be seen that the Trust-based benchmark approach starts at 0.15 and fluctuates slowly without any significant change until the end of the simulation. The Anglano approach is almost stable with a low rate of 0.08. Whereas our approach starts at 0.35 and rises to reach 0.48 at time $t = 40$. We can conclude from such a plot that our approach is capable of serving more vehicular users when compared to the other ones. It is worth mentioning that the serviced user rate is not near its upper limit due to the simulation settings where the system is overloaded with incoming requests, in addition to the hardware limitation of the wireless connections between the servers and the vehicles. Last but not least, Fig. 5.7 reflects the theoretical payoff of the average fog provider. Our method outperforms significantly both benchmark models that are consistent with the gain in the serviced user rate. In summary, our framework outperforms existing benchmark models due to its innovative combination of on-chain and off-chain mechanisms, as well as its consideration of provider reputation. The use of Blockchain-based storage for reputations allows our approach to access up-to-date and reliable reputation information, enabling more accurate predictions and decision-making. Additionally, our mechanism employs an early detection system for dishonest providers, allowing us to avoid them and maintain higher service availability. The other benchmark models are not as effective as our proposed framework because 1) the Trust-based Service Communities approach relies on forwarding reputation, which can be outdated and unreliable for predicting future reputations accurately. This leads to suboptimal decision-making and lower reputation prediction accuracy. 2) The Anglano et al. model lacks a comprehensive reputation management system and early detection mechanism, making it less capable of coping with malicious providers and ensuring stable service availability. As a result, our framework achieves a higher serviced user rate, serving more vehicular users compared to the benchmark models.

5.6 Conclusion

In this work, we addressed the problem of having untrustworthy fog providers in fog federations. In particular, we developed a Blockchain-based scheme that can significantly enhance the performance of fog federations. We also developed an on-chain and off-chain mechanism for handling fog federations formation in the presence of dishonest fog providers. The Hedonic game theoretical approach was devised in this paper to form the federation by relying on Ethereum's smart contracts. the simulation results show a significant improvement in terms of several metrics when compared to the benchmark model, and this is mainly due to the fact that Blockchain asserts the propagation of global information to be within reach of all of the involved participants in the system. For future work, we plan to work on enhancing the architecture by shifting the whole formation process to on-chain at low costs. For future work, we will consider the limitations imposed by the employment of blockchain infrastructure in the context of real-time applications and the block-out speed entailed by such technology. We will investigate techniques

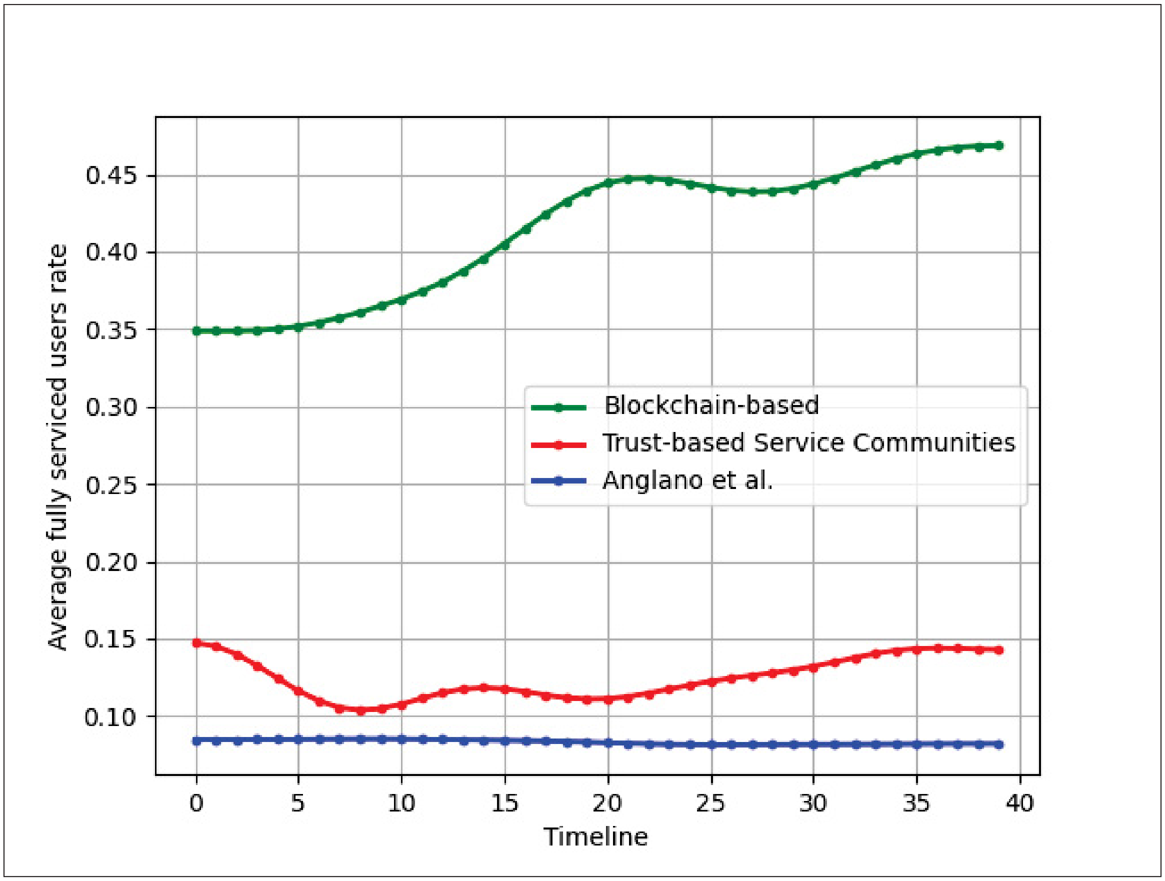


Figure 5.6 Serviced User Rate

to predict demand ahead of time and prepare for the federation formation in advance to avoid any imposed delays. By leveraging predictive analytics and machine learning algorithms, it may be possible to anticipate the resource requirements and latency-sensitive needs of the IoV applications.

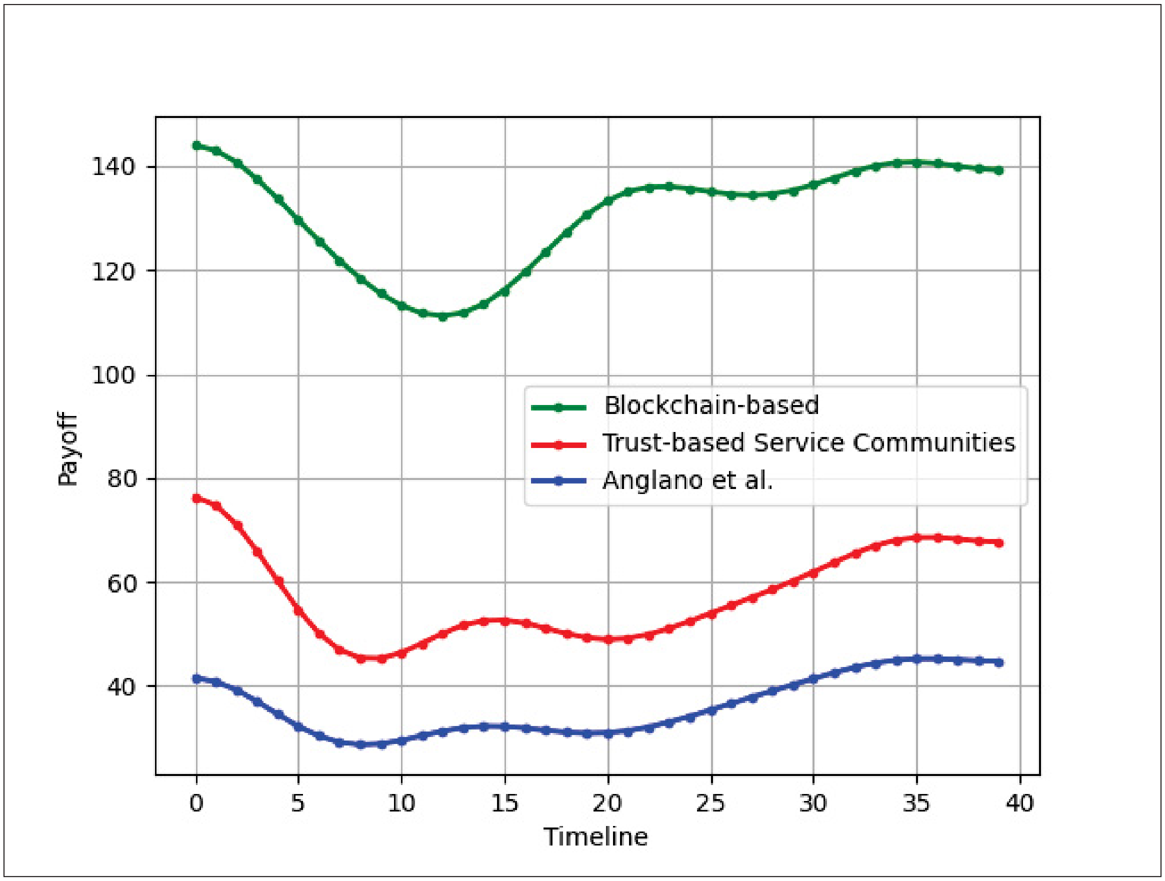


Figure 5.7 Average Payoff

CONCLUSION AND RECOMMENDATIONS

This thesis addresses the challenges and constraints in the realm of federated fog computing, with a specific focus on its application within IoT and IoV contexts. Through a series of comprehensive studies, innovative proposals, and rigorous evaluations, we have made significant contributions to the field, advancing the understanding and practical implementation of federated fog computing for intelligent IoT and IoV applications.

Our research objectives were aimed at addressing the key challenges that hinder the realization of efficient and stable federated fog computing architectures for IoT and IoV applications in the existing literature. We highlighted 4 main sub-objectives in order to realize our architecture: 1) Stabilizing the fog federation architecture. 2) Supporting mobility for critical IoV applications. 3) Empowering federated learning applications through an extended federated fog architecture. 4) Securing the formation process and eliminating malicious nodes.

Our summarised list of contributions consists of the following:

- We proposed an evolutionary game-theoretical approach to create stable fog federations, ensuring that member fog providers have no incentives to leave their respective federations.
- We introduced a dynamic fog federation scheme that accounts for the mobile nature of IoV environments. We formed adaptive federations using a Hedonic Coalition Formation mechanism to maximize QoS.
- We extended the concept of federated learning to the IoV domain by creating a horizontal federated learning architecture. This architecture leveraged a Hedonic-game theoretical model to adapt the formation of fog federations for training intelligent vehicular applications while preserving the data privacy of the data providers.

- We harnessed the power of Ethereum Blockchain technology to enhance the trust and reputation aspects of fog federation formation. By integrating on-chain smart contracts and off-chain Hedonic game theory, we introduced a secure and reputable fog federation formation process.

By implementing our architecture, we demonstrated a series of advantages that could facilitate the realization of the federated fog computing concept in a smooth and effective manner. First of all, we showed that the problem of federation instability could be resolved with a game model while showcasing its potential to improve the QoS and overall performance. Moreover, we demonstrated how we can still maintain QoS despite the mobile environment through a dynamic coalition approach. Furthermore, our extended federated fog architecture has the capability to handle multiple learning applications simultaneously while achieving higher accuracy and lower model loss in the context of IoV. Finally, we ensured fair and reliable federation establishment while penalizing untrustworthy providers through blockchain integration. Collectively, our research has provided a comprehensive and insightful understanding of the challenges and opportunities in federated fog computing for IoT and IoV applications.

As we conclude this thesis, it is clear that the journey toward efficient and secure federated fog computing for IoT and IoV is ongoing. We thoroughly investigate this paradigm and our work has laid a solid foundation in its early stages of development. Nevertheless, there are still avenues to explore and expand upon, including the integration of edge intelligence for real-time decision-making for offloading tasks among federation members, the development of fully autonomous management systems for this paradigm, and the imperative phase of real-world deployments and testing. We hope that our contributions will inspire and guide future researchers in their path to shape the paradigm of federated fog computing for years to come.

LIST OF REFERENCES

AbdulRahman, S., Tout, H., Mourad, A. & Talhi, C. (2020). FedMCCS: multicriteria client selection model for optimal IoT federated learning. *IEEE Internet of Things Journal*, 8(6), 4723–4735.

Abououf, M., Singh, S., Otrok, H., Mizouni, R. & Damiani, E. (2021). Machine Learning in Mobile Crowd Sourcing: A Behavior-Based Recruitment Model. *ACM Transactions on Internet Technology (TOIT)*, 22(1), 1–28.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials*, 17(4), 2347-2376. doi: 10.1109/COMST.2015.2444095.

Al-Hilo, A., Samir, M., Assi, C., Sharafeddine, S. & Ebrahimi, D. (2020). Cooperative Content Delivery in UAV-RSU Assisted Vehicular Networks. *Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond*, (DroneCom '20), 73–78. doi: 10.1145/3414045.3415947.

Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A. & Leonardi, S. (2012). Online team formation in social networks. *Proceedings of the 21st international conference on World Wide Web*, pp. 839–848.

Anastasi, G. F., Carlini, E., Coppola, M. & Dazzi, P. (2017). QoS-aware genetic cloud brokering. *Future Generation Computer Systems*, 75, 1–13.

Anglano, C., Guazzone, M. & Sereno, M. (2014). Maximizing Profit in Green Cellular Networks through Collaborative Games. *Computer Networks*, 75, 260–275. doi: 10.1016/j.comnet.2014.10.003.

Anglano, C., Canonico, M., Castagno, P., Guazzone, M. & Sereno, M. (2018a). A game-theoretic approach to coalition formation in fog provider federations. *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 123-130. doi: 10.1109/FMEC.2018.8364054.

Anglano, C., Canonico, M., Castagno, P., Guazzone, M. & Sereno, M. (2018b). A game-theoretic approach to coalition formation in fog provider federations. *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 123–130.

Apt, K. R. & Radzik, T. (2006). Stable partitions in coalitional games. *arXiv preprint cs/0605132*.

Balasubramanian, V., Otoum, S., Aloqaily, M., Al Ridhawi, I. & Jararweh, Y. (2020). Low-latency vehicular edge: A vehicular infrastructure model for 5G. *Simulation Modelling Practice and Theory*, 98, 101968.

Banerjee, S., Konishi, H. & Sönmez, T. (2001). Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1), 135–153.

Caldas, S., Konečny, J., McMahan, H. B. & Talwalkar, A. (2018). Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*.

Chamra, A. & Harmanani, H. (2020). A smart green house control and management system using IoT. *17th International Conference on Information Technology–New Generations (ITNG 2020)*, pp. 641–646.

Chiang, M. & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6), 854–864.

Dai, C., Liu, X., Chen, W. & Lai, C.-F. (2020). A low-latency object detection algorithm for the edge devices of IoV systems. *IEEE Transactions on Vehicular Technology*, 69(10), 11169–11178.

Dhole, A., Thomas, M. V. & Chandrasekaran, K. (2016). An efficient trust-based Game-Theoretic approach for cloud federation formation. *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1, 1–6.

Ennya, Z., Hadi, M. Y. & Abouaomar, A. (2018). Computing Tasks Distribution in Fog Computing: Coalition Game Model. *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1-4. doi: 10.1109/WINCOM.2018.8629587.

Evans, B. (2018, May). Top cloud vendors will crush $100 billion in 2018 revenue; Microsoft, Amazon, IBM hit $75 billion? Retrieved from: https://www.forbes.com/sites/bobevans1/2018/05/21/top-cloud-vendors-will-crush-100-billion-in-2018-revenue-microsoft-amazon-ibm-hit-75-billion/#3aaed51c7548.

Fawaz, W. (2018). Effect of non-cooperative vehicles on path connectivity in vehicular networks: A theoretical analysis and UAV-based remedy. *Vehicular Communications*, 11, 12–19.

Feng, C., Yu, K., Aloqaily, M., Alazab, M., Lv, Z. & Mumtaz, S. (2020). Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV. *IEEE Transactions on Vehicular Technology*, 69(11), 13784–13795.

Ghobaei-Arani, M., Souri, A. & Rahmanian, A. A. (2020). Resource management approaches in fog computing: a comprehensive review. *Journal of Grid Computing*, 18(1), 1–42.

Goiri, I., Guitart, J. & Torres, J. (2010). Characterizing cloud federation for enhancing providers' profit. *2010 IEEE 3rd International Conference on Cloud Computing*, pp. 123–130.

Gu, K., Tang, L., Jiang, J. & Jia, W. (2020). Resource allocation scheme for community-based fog computing based on reputation mechanism. *IEEE Transactions on Computational Social Systems*, 7(5), 1246–1263.

Guazzone, M., Anglano, C. & Sereno, M. (2014). A Game-Theoretic Approach to Coalition Formation in Green Cloud Federations. *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 618-625. doi: 10.1109/CCGrid.2014.37.

Hammoud, A., Otrok, H., Mourad, A., Wahab, O. A. & Bentahar, J. (2018). On the Detection of Passive Malicious Providers in Cloud Federations. *IEEE Communications Letters*, 23(1), 64–67.

Hammoud, A., Mourad, A., Otrok, H., Wahab, O. A. & Harmanani, H. (2020a). Cloud federation formation using genetic and evolutionary game theoretical models. *Future Generation Computer Systems*, 104, 92–104.

Hammoud, A., Sami, H., Mourad, A., Otrok, H., Mizouni, R. & Bentahar, J. (2020b). AI, Blockchain, and Vehicular Edge Computing for Smart and Secure IoV: Challenges and Directions. *IEEE Internet of Things Magazine*, 3(2), 68-73. doi: 10.1109/IOTM.0001.1900109.

Hammoud, A., Sami, H., Mourad, A., Otrok, H., Mizouni, R. & Bentahar, J. (2020c). AI, Blockchain, and Vehicular Edge Computing for Smart and Secure IoV: Challenges and Directions. *IEEE Internet of Things Magazine*, 3(2), 68–73.

Hammoud, A., Otrok, H., Mourad, A. & Dziong, Z. (2021). Stable federated fog formation: An evolutionary game theoretical approach. *Future Generation Computer Systems*.

Hammoud, A., Kantardjian, M., Najjar, A., Mourad, A., Otrok, H., Dziong, Z. & Guizani, N. (2022a). Dynamic Fog Federation Scheme for Internet of Vehicles. *IEEE Transactions on Network and Service Management*.

Hammoud, A., Otrok, H., Mourad, A. & Dziong, Z. (2022b). On Demand Fog Federations for Horizontal Federated Learning in IoV. *IEEE Transactions on Network and Service Management*.

Hao, M., Li, H., Luo, X., Xu, G., Yang, H. & Liu, S. (2019). Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10), 6532–6542.

Herrmann, A., Brenner, W. & Stadler, R. (2018). *Autonomous driving: how the driverless revolution will change the world*. Emerald Group Publishing.

Hogan, M., Liu, F., Sokol, A. & Tong, J. (2011). Nist cloud computing standards roadmap. *NIST Special Publication*, 35, 6–11.

Huang, T., Lin, W., Wu, W., He, L., Li, K. & Zomaya, A. Y. (2020). An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Transactions on Parallel and Distributed Systems*, 32(7), 1552–1564.

Hussain, Y., Zhiqiu, H., Akbar, M. A., Alsanad, A., Alsanad, A. A.-A., Nawaz, A., Khan, I. A. & Khan, Z. U. (2020). Context-aware trust and reputation model for fog-based IoT. *IEEE Access*, 8, 31622–31632.

Iqbal, S., Malik, A. W., Rahman, A. U. & Noor, R. M. (2020). Blockchain-based reputation management for task offloading in micro-level vehicular fog network. *IEEE Access*, 8, 52968–52980.

Islambouli, R. & Sharafeddine, S. (2019). Optimized 3D deployment of UAV-mounted cloudlets to support latency-sensitive services in IoT networks. *IEEE Access*, 7, 172860–172870.

Jebbaoui, H., Mourad, A., Otrok, H. & Haraty, R. (2015). Semantics-based approach for detecting flaws, conflicts and redundancies in XACML policies. *Computers & Electrical Engineering*, 44, 91–103.

Kaddoum, G., Nijsure, Y. & Tran, H. (2015). Generalized code index modulation technique for high-data-rate communication systems. *IEEE Transactions on Vehicular Technology*, 65(9), 7000–7009.

Khabbaz, M., Assi, C. & Sharafeddine, S. (2020). Multi-Hop V2U Path Availability Analysis in UAV-Assisted Vehicular Networks. *IEEE Internet of Things Journal*, PP, 1. doi: 10.1109/JIOT.2020.3049018.

Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E. & Bani-Hani, A. (2021). Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications*, 14(5), 2901–2925.

Khosrowshahi-Asl, E., Bentahar, J., Estrada, R., Otrok, H., Mizouni, R. & Khosravifar, B. (2020). Dynamic formation of service communities in the cloud under distribution and incomplete information settings. *Concurrency and Computation: Practice and Experience*, 32(1), e4338.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T. & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Krumke, S. O. & Thielen, C. (2013). The generalized assignment problem with minimum quantities. *European Journal of Operational Research*, 228(1), 46-55. doi: https://doi.org/10.1016/j.ejor.2013.01.027.

Lai, P., He, Q., Abdelrazek, M., Chen, F., Hosking, J., Grundy, J. & Yang, Y. (2018). Optimal edge user allocation in edge computing with variable sized vector bin packing. *Proceedings of the International Conference on Service-Oriented Computing*, pp. 230–245.

Li, Y., Anh, N. T., Nooh, A. S., Ra, K. & Jo, M. (2018). Dynamic mobile cloudlet clustering for fog computing. *2018 international conference on electronics, information, and communication (iceic)*, pp. 1–4.

Lim, W. Y. B., Xiong, Z., Niyato, D., Huang, J., Hua, X.-S. & Miao, C. (2020). Incentive mechanism design for federated learning in the internet of vehicles. *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–5.

Liu, S., Yu, J., Deng, X. & Wan, S. (2021). FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks. *IEEE Transactions on Intelligent Transportation Systems*.

Madakam, S. & Ramachandran, R. (2015). Barcelona smart city: the Heaven on Earth (internet of things: technological God). *ZTE Communications*, 13(4), 3–9.

Maheswaran, M., Yang, T. & Memon, S. (2019, 07). A Fog Computing Framework for Autonomous Driving Assist: Architecture, Experiments, and Challenges.

Mahmud, R., Ramamohanarao, K. & Buyya, R. (2018). Latency-aware application module management for fog computing environments. *ACM Transactions on Internet Technology (TOIT)*, 19(1), 1–21.

Manoochehri, H. E. & Wenkstern, R. Z. (2017). Dynamic coalition structure generation for autonomous connected vehicles. *2017 IEEE International Conference on Agents (ICA)*, pp. 21-26. doi: 10.1109/AGENTS.2017.8015295.

Mashayekhy, L., Nejad, M. M. & Grosu, D. (2014). Cloud federations in the sky: Formation game and mechanism. *IEEE Transactions on Cloud Computing*, 3(1), 14–27.

Mashayekhy, L., Nejad, M. M. & Grosu, D. (2015). Cloud federations in the sky: Formation game and mechanism. *IEEE Transactions on Cloud Computing*, 3(1), 14–27.

Moati, N., Otrok, H., Mourad, A. & Robert, J.-M. (2014). Reputation-based cooperative detection model of selfish nodes in cluster-based QoS-OLSR protocol. *Wireless personal communications*, 75, 1747–1768.

Mourad, A., Tout, H., Wahab, O. A., Otrok, H. & Dbouk, T. (2020). Ad Hoc Vehicular Fog Enabling Cooperative Low-Latency Intrusion Detection. *IEEE Internet of Things Journal*, 8(2), 829–843.

Networkers. (2018, May). Sky-high market growth driving demand for cloud infrastructure specialists. Retrieved from: https://www.networkerstechnology.com/growth-cloud-demand-infrastructure-specialists.

Nishio, T. & Yonetani, R. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7.

Niyato, D., Vasilakos, A. V. & Kun, Z. (2011). Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach. *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 215–224.

Nolan, K. E. & Doyle, L. E. (2007). Teamwork and collaboration in cognitive wireless networks. *IEEE Wireless Communications*, 14(4), 22–27.

Novet, J. (2018, Feb). Amazon cloud revenue jumps 45 percent in fourth quarter. Retrieved from: https://www.cnbc.com/2018/02/01/aws-earnings-q4-2017.html.

Oueis, J., Strinati, E. C. & Barbarossa, S. (2015). The fog balancing: Load distribution for small cell cloud computing. *2015 IEEE 81st vehicular technology conference (VTC spring)*, pp. 1–6.

Poirot, M. G., Vepakomma, P., Chang, K., Kalpathy-Cramer, J., Gupta, R. & Raskar, R. (2019). Split Learning for collaborative deep learning in healthcare. *arXiv preprint arXiv:1912.12115*.

Pokhrel, S. R. & Choi, J. (2020). Improving TCP performance over WiFi for internet of vehicles: A federated learning approach. *IEEE Transactions on Vehicular Technology*, 69(6), 6798–6802.

Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Yu, S., Li, B. & Zheng, G. (2020). Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet of Things Journal*, 7(6), 5171–5183.

Rahman, S. A., Tout, H., Talhi, C. & Mourad, A. (2020). Internet of things intrusion detection: Centralized, on-device, or federated learning? *IEEE Network*.

Ray, B. K., Saha, A. & Roy, S. (2018). Migration cost and profit oriented cloud federation formation: hedonic coalition game based approach. *Cluster Computing*, 21(4), 1981–1999.

Rebai, S., Hadji, M. & Zeghlache, D. (2015). Improving profit through cloud federation. *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 732–739.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J. et al. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4), 4–1.

Saad, W., Han, Z., Debbah, M., Hjorungnes, A. & Basar, T. (2009). Coalitional game theory for communication networks. *IEEE Signal Processing Magazine*, 26(5), 77-97. doi: 10.1109/MSP.2009.000000.

Saha, R., Misra, S. & Deb, P. K. (2020). FogFL: Fog-Assisted Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet of Things Journal*, 8(10), 8456–8463.

Salahuddin, M. A., Al-Fuqaha, A., Guizani, M. & Cherkaoui, S. (2014). RSU cloud and its resource management in support of enhanced vehicular applications. *2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 127–132.

Sami, H. & Mourad, A. (2020). Dynamic On-Demand Fog Formation Offering On-The-Fly IoT Service Deployment. *IEEE Transactions on Network and Service Management*.

Sami, H., Mourad, A., Otrok, H. & Bentahar, J. (2020). Fscaler: Automatic resource scaling of containers in fog clusters using reinforcement learning. *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1824–1829.

Saputra, Y. M., Dinh, H. T., Nguyen, D., Tran, L.-N., Gong, S. & Dutkiewicz, E. (2021). Dynamic Federated Learning-Based Economic Framework for Internet-of-Vehicles. *IEEE Transactions on Mobile Computing*.

Schuster, P. & Sigmund, K. (1983). Replicator dynamics. *Journal of theoretical biology*, 100(3), 533–538.

Shamseddine, H., Nizam, J., Hammoud, A., Mourad, A., Otrok, H., Harmanani, H. & Dziong, Z. (2020a). A Novel Federated Fog Architecture Embedding Intelligent Formation. *IEEE Network*, 1-7. doi: 10.1109/MNET.011.2000505.

Shamseddine, H., Nizam, J., Hammoud, A., Mourad, A., Otrok, H., Harmanani, H. & Dziong, Z. (2020b). A Novel Federated Fog Architecture Embedding Intelligent Formation. *IEEE Network*.

Sharaf, M. & El-Ghazawi, T. (2019). Preference-based and Homogeneous Coalition Formation in Fog Computing. *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1-6. doi: 10.1109/CSCN.2019.8931361.

Sharmin, Z., Malik, A. W., Rahman, A. U. & Noor, R. M. (2020a). Towards sustainable micro-level fog federated load-sharing in internet of vehicles. *IEEE Internet of Things Journal*.

Sharmin, Z., Malik, A. W., Ur Rahman, A. & MD Noor, R. (2020b). Toward Sustainable Micro-Level Fog-Federated Load Sharing in Internet of Vehicles. *IEEE Internet of Things Journal*, 7(4), 3614-3622. doi: 10.1109/JIOT.2020.2973420.

Shih, Y.-Y., Chung, W.-H., Pang, A.-C., Chiu, T.-C. & Wei, H.-Y. (2016). Enabling low-latency applications in fog-radio access networks. *IEEE network*, 31(1), 52–58.

Shih, Y.-Y., Chung, W.-H., Pang, A.-C., Chiu, T.-C. & Wei, H.-Y. (2017). Enabling Low-Latency Applications in Fog-Radio Access Networks. *IEEE Network*, 31(1), 52-58. doi: 10.1109/MNET.2016.1500279NM.

Shurrab, M., Singh, S., Mizouni, R. & Otrok, H. (2022). IoT Sensor Selection for Target Localization: A Reinforcement Learning based Approach. *Ad Hoc Networks*, 134, 102927.

Smirnov, A., Sheremetov, L. & Teslya, N. (2019). Robot Coalition Formation Based on Fuzzy Cooperative Games over Blockchain-Based Smart Contracts. *IFIP International Conference on Product Lifecycle Management*, pp. 346–355.

Smith, J. M. & Price, G. R. (1973). The logic of animal conflict. *Nature*, 246(5427), 15.

Sorkhoh, I., Ebrahimi, D., Assi, C., Sharafeddine, S. & Khabbaz, M. (2020). An infrastructure-assisted workload scheduling for computational resources exploitation in the fog-enabled vehicular network. *IEEE Internet of Things Journal*, 7(6), 5021–5032.

Stantchev, V., Barnawi, A., Ghulam, S., Schubert, J. & Tamm, G. (2015). Smart items, fog and cloud computing as enablers of servitization in healthcare. *Sensors & Transducers*, 185(2), 121.

Stojmenovic, I. & Wen, S. (2014). The fog computing paradigm: Scenarios and security issues. *2014 federated conference on computer science and information systems*, pp. 1–8.

Sun, Y., Dang, T. & Zhou, J. (2016). User scheduling and cluster formation in fog computing based radio access networks. *2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pp. 1–4.

Taghavi, M., Bentahar, J., Otrok, H. & Bakhtiyari, K. (2018). Cloudchain: A blockchain-based coopetition differential game model for cloud computing. *Service-Oriented Computing: 16th International Conference, ICSOC 2018, Hangzhou, China, November 12-15, 2018, Proceedings 16*, pp. 146–161.

Taghavi, M., Bentahar, J., Otrok, H. & Bakhtiyari, K. (2019). A blockchain-based model for cloud service quality monitoring. *IEEE Transactions on Services Computing*, 13(2), 276–288.

Tavana, M., Azizi, F., Azizi, F. & Behzadian, M. (2013). A fuzzy inference system with application to player selection and team formation in multi-player sports. *Sport Management Review*, 16(1), 97–110.

Tello, G., Gianini, G., Mizouni, R. & Damiani, E. (2019). Machine learning-based framework for log-lifting in business process mining applications. *International Conference on Business Process Management*, pp. 232–249.

Tran, H.-V., Kaddoum, G. & Truong, K. T. (2018). Resource allocation in SWIPT networks under a nonlinear energy harvesting model: Power efficiency, user fairness, and channel nonreciprocity. *IEEE Transactions on Vehicular Technology*, 67(9), 8466–8480.

Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R. & Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 1–11.

Tseng, T.-L. B., Huang, C.-C., Chu, H.-W. & Gung, R. R. (2004). Novel approach to multi-functional project team formation. *International Journal of Project Management*, 22(2), 147–159.

ur Rehman, M. H., Salah, K., Damiani, E. & Svetinovic, D. (2020). Towards blockchain-based reputation-aware federated learning. *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 183–188.

Veillon, V., Denninnart, C. & Salehi, M. A. (2019a). F-FDN: Federation of Fog Computing Systems for Low Latency Video Streaming. *2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC)*, pp. 1-9. doi: 10.1109/CFEC.2019.8733154.

Veillon, V., Denninnart, C. & Salehi, M. A. (2019b). F-FDN: Federation of Fog Computing Systems for Low Latency Video Streaming. *2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC)*, pp. 1–9.

Wahab, O. A., Bentahar, J., Otrok, H. & Mourad, A. (2015). A survey on trust and reputation models for Web services: Single, composite, and communities. *Decision Support Systems*, 74, 121–134.

Wahab, O. A., Bentahar, J., Otrok, H. & Mourad, A. (2016a). A Stackelberg game for distributed formation of business-driven services communities. *Expert Systems with Applications*, 45, 359–372.

Wahab, O. A., Bentahar, J., Otrok, H. & Mourad, A. (2016b). Towards trustworthy multi-cloud services communities: A trust-based hedonic coalitional game. *IEEE Transactions on Services Computing*, 11(1), 184–201.

Wainakh, A., Guinea, A. S., Grube, T. & Mühlhäuser, M. (2020). Enhancing privacy via hierarchical federated learning. *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 344–347.

Wang, S., Ruan, Y., Tu, Y., Wagle, S., Brinton, C. G. & Joe-Wong, C. (2021). Network-aware optimization of distributed learning for fog computing. *IEEE/ACM Transactions on Networking*.

Wu, Z., Lu, Z., Hung, P. C., Huang, S.-C., Tong, Y. & Wang, Z. (2019). QaMeC: A QoS-driven IoVs application optimizing deployment scheme in multimedia edge clouds. *Future Generation Computer Systems*, 92, 17–28.

Xia, Q., Ye, W., Tao, Z., Wu, J. & Li, Q. (2021). A Survey of Federated Learning for Edge Computing: Research Problems and Solutions. *High-Confidence Computing*, 100008.

Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J. & Wang, F. (2021). Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1), 1–19.

Xu, R., Baracaldo, N., Zhou, Y., Anwar, A. & Ludwig, H. (2019). Hybridalpha: An efficient approach for privacy-preserving federated learning. *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 13–23.

Xue, K., Hong, J., Ma, Y., Wei, D. S., Hong, P. & Yu, N. (2018). Fog-aided verifiable privacy preserving access control for latency-sensitive data sharing in vehicular cloud computing. *IEEE Network*, 32(3), 7–13.

Xue, K., He, P., Zhang, X., Xia, Q., Wei, D. S., Yue, H. & Wu, F. (2019). A secure, efficient, and accountable edge-based access control framework for information centric networks. *IEEE/ACM Transactions on Networking*, 27(3), 1220–1233.

Yang, F., Wang, S., Li, J., Liu, Z. & Sun, Q. (2014). An overview of internet of vehicles. *China communications*, 11(10), 1–15.

Yang, J., Chen, Y., Xue, K., Han, J., Li, J., Wei, D. S., Sun, Q. & Lu, J. (2022). IEACC: An Intelligent Edge-Aided Congestion Control Scheme for Named Data Networking with Deep Reinforcement Learning. *IEEE Transactions on Network and Service Management*.

Yao, J. & Ansari, N. (2020). Enhancing Federated Learning in Fog-Aided IoT by CPU Frequency and Wireless Power Control. *IEEE Internet of Things Journal*.

Yi, S., Li, C. & Li, Q. (2015). A survey of fog computing: concepts, applications and issues. *Proceedings of the 2015 workshop on mobile big data*, pp. 37–42.

Yu, M., Zang, W. & Reagor, B. (2007). Decentralized Trust Management based on the Reputation of Information Sources. *2007 IEEE International Conference on Networking, Sensing and Control*, pp. 212–217.

Zhang, W., Zhang, Z. & Chao, H.-C. (2017). Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management. *IEEE Communications Magazine*, 55(12), 60–67.

Zhao, Z., Feng, C., Yang, H. H. & Luo, X. (2020). Federated-Learning-Enabled Intelligent Fog Radio Access Networks: Fundamental Theory, Key Techniques, and Future Trends. *IEEE Wireless Communications*, 27(2), 22–28.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X. & Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4), 352–375.

Zhou, C., Fu, A., Yu, S., Yang, W., Wang, H. & Zhang, Y. (2020a). Privacy-Preserving Federated Learning in Fog Computing. *IEEE Internet of Things Journal*.

Zhou, H., Xu, W., Chen, J. & Wang, W. (2020b). Evolutionary V2X technologies toward the Internet of vehicles: Challenges and opportunities. *Proceedings of the IEEE*, 108(2), 308–323.