January 2022

Sdn-Enabled Efficient Resource Utilization In A Secure, Trustworthy And Privacy Preserving Iov-Fog Environment

Jamal N. Alotaibi
Wayne State University

SDN-ENABLED EFFICIENT RESOURCE UTILIZATION IN A SECURE, TRUSTWORTHY AND PRIVACY PRESERVING IOV-FOG ENVIRONMENT

by

JAMAL ALOTAIBI

DISSERTATION

Submitted to the Graduate School,

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2022

MAJOR: COMPUTER ENGINEERING

Approved By:

_____

Advisor Date

_____

_____

_____

_____

DEDICATION

To my Father , Mother, Brothers,Sisters, and All my family.

ACKNOWLEDGEMENTS

First and foremost, I am grateful to my advisor Professor Lubna Alazzawi and my Co-advisor Professor Harpreet Singh,for their essential counsel, unwavering support, and patience throughout my PhD studies. Their extensive expertise and depth of knowledge have aided me throughout my academic career and daily life. I'd like to express my gratitude to the members of my Ph.D. dissertation committee, Professor Ivan Avrutsky, Professor Gholam-Abbas Nazri, and Professor Ali Elkateeb , for their valuable feedback and suggestions on my dissertation. I'd like to express my gratitude to everybody at Wayne State University. It is because of their generous assistance and support that my studies and life in the United States have been so enjoyable. I'd also like to express my gratitude to Computer Collage at Qassim University for awarding me a scholarship to complete my Ph.D. Finally, I would like to express my gratitude to my Mother, my Brothers and my Sisters. It would have been impossible for me to finish my studies without their tremendous understanding and encouragement over the last few years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

The Intelligent Transportation System (ITS) is a decentralized environment in which multiple services are integrated. The goal of this thesis is to investigate application areas within the broad domain of ITS. In particular, Internet-of-Vehicles (IoV) is one application area in ITS, which requires both fast and secure communications [1, 2]. ITS is a network of connected systems that can provide a more efficient and effective transportation system to address a rapidly expanding and significant societal challenge. It enables intelligent traffic management, monitoring, and dynamic information services, according to [3]. Furthermore, it solves issues such as traffic congestion, road safety, transportation efficiency, and environmental conservation, among others, by leveraging advanced communication and computing technologies. However, with advancement in wireless communication and computing technologies, researchers are prompted to reconsider and redevelop ITS.

Wireless communication has traditionally either point-to-point (one transmitter to one receiver) or point-to-multipoint (one sender to several receivers) (e.g., radio broadcast). Both types of communications leverage limited mobility to relay signals over long distances. When it comes to vehicle–to–vehicle (V2V)communication or vehicle–to– infrastructure (V2I) communication, mobility is both high and sporadic, and therefore it gets its name: Vehicular Ad–Hoc Networks (VANET) [4]. The topology of VANETs changes relatively quickly due to one-time interactions between vehicles. As a result, failures of V2V links are common in these types of vehicular networks. Links between vehicles traveling in opposite directions, in particular, only last a few seconds, so the network is constantly disconnected.

The Internet of Things (IoT) [5], on the other hand, is a network that links many sorts of devices and systems to allow a connection between them at any time, from any place. For example, smart home systems, smart energy, e-health, as well as ITS are all IoT applications that are appealing. A new paradigm known as Internet-of-Vehicles (IoV) [1, 6] has recently arisen as part of ITS by combining IoT with mobile internet technologies (i.e. 3G/4G/5G). IoV is a distributed wireless networking and information exchange infrastructure that evolved from VANETs and is expected to develop into the Internet of Autonomous Vehicles (AV) in the future [7]. In contrast to VANET's V2V communication paradigm, IoV uses communication technologies such as IEEE 802.11p WAVE (Wireless Access in Vehicular Environment), cellular 4G or 5G for data transfer. Multiple vehicles are connected by a road side unit (RSU), which forms multiple RSU zones.

In IoV, the RSU is connected to remote (or virtual) storage and computational resources through cloud computing. Therefore, IoV applications are able to use data aggregation, data mining, data storage, processing power and some of the services that are available through the cloud integration [8]. Moreover, cloud computing promises versatile and dynamic IT infrastructures, QoS-assured computing environments, and configurable software services [9]. It has already surpassed grid computing due to its numerous benefits. Although Cloud computing has received a lot of attention, there are still no widely agreed definitions. Several factors have led to this situation:

1. Researchers and engineers from various fields, such as Grid computing, software engineering, and database management, are involved in cloud computing. They approach Cloud computing from different angles.

2. Cloud computing enabling technologies such as Web and Service-Oriented Computing, are also dynamic as well as evolving and advancing.

Although the cloud layer of the IoV has massive storage and processing capabilities, uploading or downloading data from the cloud layer may create significant delays if numerous vehicles accessing the cloud layer at the same time. This may induce core/backbone network congestion, resulting in considerable quality-of-service (QoS) deterioration and prolonged end-to-end delays.

The problem of end-to-end delays in IoV can be addressed with the advent of a new paradigm called vehicular fog computing (VFC) [10]. As the fog computing offers a layer between the cloud and the IoV, it minimizes delays and improves QoS. Fog computing is formally defined as an extension of the cloud computing paradigm from the network's core to the network's edge, according to Cisco [11]. It's a highly virtualized platform that connects end-devices to traditional cloud servers and provides computing, storage, and networking services. As fog computing enables computing at the network's edge, it allows new applications and services to be delivered with low latency. Commercial edge routers, for example, advertise the speed and cores of the processor along with the built-in network storage. Those routers could be used to create new servers. Furthermore, the facilities that can provide resources for services at the network's edge in fog computing are referred to as fog nodes. However, due to the very dynamic environment of IoV, there still exist challenges in developing fog-based IoV applications [12]. To develop an effective and secure VFC system, challenges such as efficient resources utilization, security, and privacy must be addressed [13]. A viable solution can be using software-defined networking (SDN),

PKI-based authentication and blockchain technologies paired with VFC.

SDN decouples the network control and data planes in order to enable central network intelligence management. Whereas, in traditional network architecture, control and data planes are tightly coupled in forwarding devices. The tight coupling of data and control planes means that the software and hardware are strongly coupled in network elements. All network policies in the traditional network are installed manually, and any adjustment in policy changes is managed in each device manually. Therefore, the traditional network architecture is distributed in nature, which makes the overall operation cumbersome and challenging for its managing, controlling, configuring, and innovations to be implemented down the line. SDN controller is directly programmable, and by using SDN controller it is easy to control and manage the network. Any communication between forwarding device and applications occur by directing towards the controller. The communication between controllers and switches and other network nodes is maintained by southbound API, e.g., OpenFlow. Different SDN controller devices are interconnected through east and west APIs. In contrast to southbound API, northbound API allows communication among the higher-level components. Furthermore, network function virtualization (NFV) offers a potential solution for dynamically programming service functionalities such as firewalls, domain name systems (DNS), network address translation (NAT), and video transcoding as software instances known as virtual network functions (VNFs) at edge servers without incurring significant costs [14]. The NFV may be used with fog to achieve computation-oriented service provisioning at the network edge. By allowing for network-level resource allocation and flexible service provisioning, the combination of SDN and NFV has the potential to improve the performance of IoV systems in terms of end-to-end latency and

quality-of-service (QoS).

The limited computing resources and capacity of on-board equipment attached to vehicles remains a significant challenge to overcome in connected cars. This is because many new types of applications emerging in IoV, such as AI-based environment detection and customized navigation, are response-sensitive and require complex computing and real-time analysis. Therefore, SDN and fog computing-based IoV must support not only task offloading but also efficient utilization of available fog resources. To put it another way, IoV systems with SDN controllers must improve fog computing utilization while avoiding task offloading to cloud computing. The required bandwidth between fog and cloud computing, as well as the latency, can be decreased by maximizing fog computing resources utilization. This will result in allowing the task deadline to be reached efficiently. The SDN controller has a global view of the network architecture and may use AI modules to efficiently distribute load to fog nodes.

Furthermore, as machine-learning evolves, fog computing becomes more efficient and has performed exceptionally well in tasks such as resource scheduling, prediction, and classification, etc. Nevertheless, most machine learning techniques bundle nodes' private data onto a central database (i.e., on a cloud data center) to accomplish model training, which can result in privacy leaks. Private data (i.e., in the case of IoVs) comprises vehicular confidential information such as location, speed, and driving preference, which is strongly linked to drivers' safety and privacy. As a result, novel machine learning strategy such as Federated learning (FL) [15] can be used to develop a common prediction model collectively at the nodes while keeping all the training data private to the nodes. As a result, nodes collaborate to train a global model in a decentralized manner, decoupling

the capacity to execute machine learning from the necessity to store data in the cloud, to preserve the privacy of private data.

In addition, in the IoV, messages transferred between various types of access layer nodes, such as vehicle-to-vehicles (V2V), vehicle-to-roadside units (V2R), and vehicle-to-everything (V2X), must be secured. A vehicle exchanges 100's of messages in a $\mathtt{ms}$, which can be tampered by an attacker to get access to the main IoV network [16]. This makes IoV vulnerable to security and privacy threats. As a result, IoV may be exposed to security and privacy risks. For example, IoV could be vulnerable to DDoS (Distributed Denial of Service) attacks or lack authentication capabilities. Due to such vulnerabilities, compromised communications can be delivered to vehicles; messages can be modified with to communicate inaccurate information, and other malicious activities, such as brake failure or steering impairment, can also occur. For example, consider AUVs (Autonomous Vehicles) that use the IoV network, due to lack of a driver capable of controlling such emergency scenarios, this might result in a very dangerous situation.

As a result, authentication is the first and most important step in addressing security issues and preventing unauthorized access on the IoV network, as well as defending against attacks such as man-in-the-middle, privilege insider, impersonation, and known key. Furthermore, pseudonyms (or pseudonymous credentials) have been included into authentication techniques to deliver services without including any personally identifiable information that may be connected to the pseudonym holder's genuine identity [17]. In the context of providing anonymity for electronic transactions, a digital pseudonym is described as a "public key used to validate signatures issued by the anonymous holder of the matching private key" [18]. As a result, a pseudonym allows the verification of a cer-

tain entity without disclosing the bearer's true identity. Furthermore, institutions such as registration authority (RA), RSUs, and vehicles communicate with one another in a conventional IoV architecture. RAs are often in charge of registering vehicles and RSUs into the IoV network before to deployment.

However, the FL technique to model training is vulnerable to model poisoning attacks [19] or considering the risks of information leaking in a fog computing server run by a third party. Moreover, in FL, the global model is updated on the cloud that vehicles can access for inference. As a result, blockchain technology can be used to ensure the trust is maintained throughout the FL process [20, 21]. The blockchain consortium is responsible to ensure the cloud, fog nodes and vehicles trust each other when the updates are shared with the cloud and the global learned model is used by vehicles for inference. Blockchain technology has been widely adopted because of its secure, anonymous, and decentralized trust features, as well as the fact that it is a transparent peer-to-peer (P2P) distributed ledger [22]. There are various blockchain functions such as smart contacts, proof-of-work, and the public ledger principle, which can be used to create an effective VFC system and solve the privacy issues in IoV.

The thesis aims to ensure secure real-time data transmission, effective resource management, and flexible networking in IoV. This chapter begins with the motivation for implementing a fast and secure communication framework for IoV in Section 1.1, which is followed by the research objectives in Section 1.2. The novel contributions of this research are reported in Section 1.3. Finally, in Section 1.4, we provide the complete outline of this thesis.

1.1 Motivation

In this work, we are motivated to use the aforementioned (in the Introduction) enabling technologies to achieve a fast and secure communication in a highly dynamic IoV environment. We use SDN-enabled VFC to achieve reliable communication, high QoS, and balanced data flows in a highly dynamic IoV environment, which can not be assured by the existing underlying networks. Moreover, we use Reinforcement Learning (RL) algorithm on SDN controller for efficient utilization of fog resources. When SDN and IoV are coupled, the result is a centralized Software-Defined Internet of Vehicles (SD-IoV), which provides capabilities for successfully managing and controlling the IoV network. The SDN controller selects the best fog nodes at the fog layer, which efficiently distribute the load, utilizes the available resources, and minimizes the end-to-end delay. In addition, we propose a lightweight Public Key Infrastructure-based (PKI) authentication scheme for vehicle and fog node authentication and registration to the SD-IoV network. It can improve the access layer's overall security, which covers the communication link between V2V or vehicle-to-RSUs, as well as the core network's security. Moreover, we propose a blockchain-based trust mechanism to update a global machine learning model in federated learning that can be used by vehicles for inference.

1.2 Research Objectives

The aim of the research is to create a fast and secure communication system for the IoV's extremely dynamic environment. Furthermore, we employ a variety of enabling technologies to aid in the achievement of various research objectives as well as the development of the system. SDN-based vehicular fog computing is at the heart of the system

design, and it is essential to meet the following objectives:

1. Minimizing the end-to-end latency between end-devices (i.e., vehicles, RSUs) and fog nodes while improving the QoS.

2. Efficiently utilizing the available resources at the fog nodes while delivering the result on time to the end-devices (i.e., vehicles). In other words, balancing the load in the fog layer will efficiently use the available resources while not depending too much to offload to the cloud.

In addition, this work include the design of a lightweight authentication scheme to secure all kinds of communications in the IoV access layer (e.g., vehicles-to-vehicles, vehicles-to-RSU, vehicles-to-fog nodes). The authentication scheme achieves the following objectives:

1. Allow legitimate vehicles and RSUs access to use the resources, as well as providing a secure communication channel for sending messages.

2. Defend against various attacks such as: Man-in-the-middle Attack, Privilege Insider Attack, Impersonation Attack, and Known key.

3. Prevent any association with the actual information (or identification) shared by vehicles over the IoV network.

Moreover, our work is based on the federated learning (FL) approach to create a machine learning model that can be applied to a variety of analytics. The vehicles are fitted with on-board units (OBU), which collect road and driving data. This data is critical and private. Using the FL technique, we achieve the following objectives.

1. Preserve the privacy of the users (or vehicles) by not sending the private data to the cloud.

2. Train a global model that could be used for inference by the vehicles/RSUs connected to the IoV network.

3. Allowing only honest and reliable vehicles to take part in the training in order to guarantee the reliability of knowledge generated as a consequence of model training.

Furthermore, we use blockchain technology to achieve the objective of ensuring fog nodes and the cloud can trust each other while sharing model updates.

1.3 Contributions

The following paragraphs highlight the main contributions of this work.

Contribution 1: Our approach enables task offloading from end-devices in highly dynamic IoV settings. Task offloading enables computationally-intensive tasks of resource-constrained devices (sensors attached to vehicles and RSUs) to be executed on resource-rich fog nodes. The end-devices in the IoV network are primarily mobile vehicles. However, static devices such as RSUs can also be considered as end-devices to use the virtual resources of the cloud or fog.

Contribution 2: Our approach efficiently uses the available resources in order to improve the overall QoS of applications and provide real-time results to the end-devices in the IoV environment. Using the re-enforcement learning (RL) algorithm in the SDN controller, the framework learns efficiently distributing the load among fog nodes while minimizing the

latency.

Contribution 3: Our approach secure all sort of communications in the IoV access layer using a lightweight authentication scheme. In the IoV environment, vehicles send messages to RSUs, other vehicles, and fog nodes or the cloud for computation and other resource utilization. These messages are encrypted using PKI in order to protect all connected devices in the IoV network. Therefore, providing trust among devices.

Contribution 4: Given the high dynamic environment and local data available to vehicles in IoV, our proposed approach explores the learning of global model using FL in conjunction with fog computing. Furthermore, the benefits of both fog computing and FL are utilized to overcome the end-to-end delays and learning of global model.

Contribution 5: Our proposed appraoch is based on a reputation scheme that rewards honest vehicles, which improves the accuracy of the FL global learned model. Also, it ensures trust between fog nodes and the cloud when local learned model updates from the fog nodes are shared with the cloud and vice versa.

1.4 Thesis Outline

This thesis starts by investigating the highly dynamic mobile environment of IoV. The thesis are particularly focused on challenges such as efficient utilization of available resources, secure and trustworthy communication and privacy preservation. To achieve the goals, we discussed the background of the enabling technologies and the current state-of-the-art in Chapter 2. The framework for effective resource utilization, secure and trustworthy communication, and privacy preservation is introduced using the basis laid forth

in Chapter 2. In Chapter 3, we go over our framework in depth. Furthermore, in the next three Chapters, we will discuss the simulation studies. Chapter 4, *Load Balancing in Fog Computing-based IoV*: we will discuss our proposed method to efficiently utilize the available resources in the fog layer. Chapter 5, *A Lightweight and Fog-based Authentication Scheme for Internet-of-Vehicles*: we will discuss our PKI-based authentication scheme for secure communication in IoV. Chapter 6, *A Privacy Preserving-based and Trust-based IoV-Fog Environment*: we will discuss our proposed method for ensuring trust among different communication nodes as well as privacy preservation in IoV. Chapter 7 is the Conclusion in which we provide an overview of what we have achieved.

1.5 Research Publications

This work resulted in the following publications.

1. Peer-reviewed: Alotaibi, Jamal, and Lubna Alazzawi. **Safiov: A secure and fast communication in fog-based internet-of-vehicles using sdn and blockchain.** *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2021.*

2. Peer-reviewed: Alotaibi, Jamal, and Lubna Alazzawi. **A Lightweight and Fog-based Authentication Scheme for Internet-of-Vehicles.** *In Proceedings of the IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, 2021.*

3. Accepted: Alotaibi, Jamal, and Lubna Alazzawi. **PPIoV: A Privacy Preserving-based Framework for IoV-Fog environment using Federated Learning and Block-chain.** *Accepted for publication: to appear in the proceedings of IEEE World AI IoT Congress 2022.*

CHAPTER 2 BACKGROUND

In this chapter, we will do a literature review of Internet-of-Vehicles (IoV). In particular, we will explore the backdrop of enabling technologies in order to create ultra-reliable and secure communication in the IoV. Software-defined networking, fog and cloud computing, and blockchain technologies provide virtual computing and storage resources, as well as load balancing and secure data traffic communication. We'll emphasize the most important parts of these strategies and describe them in such a way that the reader gets a holistic perspective of all the related areas and disciplines. We discuss the enabling technologies in the below sections.

2.1 Cloud computing

Cloud computing is a parallel and distributed architecture that consists of a series of virtualized machines that are gradually networked. Virtualization is a method for dividing a computing asset into many autonomous execution conditions, known as Virtual Machines (VMs). These VMs are used to deliver services throughout a system. The amount of virtual and physical assets in a cloud domain is a critical component of the administration framework, as its productivity has a direct influence on the overall performance and cost of the system. An inefficient resource distribution has a direct detrimental influence on execution and cost in this way. Given this, the primary goal of asset portioning in such situations is to make use of the foundation assets and connect them to achieve better throughput in order to address large-scale computation concerns. When a client sends a solicitation with their requirements, computing assets are dispersed in this situation [23].

2.2 Fog Computing

Fog proposes the deployment of storage, computing and networking resources anywhere in the device-to-cloud continuum, and provides task offloading with flexibility and low latency [24]. Fog computing is also beneficial for dealing with the heterogeneity of IoV access layer devices since it abstracts the underlying communication technology. Fog nodes are placed near devices and end users in fog computing. A fog node is any device that delivers storage, communication, and computation services at the network edge (i.e., routers, switches, VMs, or access points (APs)). The closeness of fog nodes to access devices reduces end-to-end latency and bandwidth consumption. Fog computing, in particular, is designed to process time critical IoV applications in real-time, in sub-seconds. However, due to the cost issues, ubiquitous fog nodes deployment is still unrealistic.

In addition, an infrastructure is required to allocate computational and network resources to each application in order to meet QoS requirements. It's difficult to manage and control distributed fog nodes while also coordinating their activities with a cloud that's located distantly. As a result, while offloading tasks from IoT devices, the logically unified fog computing confronts numerous issues, including identifying a reliable path, minimizing end-to-end latency, minimizing traffic overhead, network scalability, node mobility, and real-time data delivery. To solve these issues and meet the QoS requirements of logically unified fog computing, a new paradigm called SDN-based fog computing is proposed, which combines the advantages of two emerging technologies: SDN and fog computing.

2.3 Software-Defined Networking

Traditional Internet infrastructure is distributed in nature, resulting in the following constraints. First, network administrators use numerous command-line-interface (CLI) instructions to manually set up-each switch's control and management plan. It makes configuring the conventional Internet a time-consuming and error-prone task. According to one research, manual configuration accounts for 62% of network outages. According to the research, network administration consumes 80% of an organization's IT expenditure [25]. Therefore, the traditional Internet infrastructure has a number of flaws that make managing, regulating, configuring, and implementing innovations difficult and time-consuming.

To solve the problem of a traditional network, the researchers proposed Software-Defined Networking (SDN), a new logically centralized architecture in which the control plane and management plane are combined into a single centralized entity called "SDN controller," which is directly programmable [26].

SDN Controller is a key component of SDN, and it is easy to monitor and manage the network performance, load-balancing, security enforcement, and optimization decisions using the controller. In ddition, SDN architecture is composed of three planes, as shown in Fig 1. The application plane is the topmost level, followed by the data plane and finally the control plane [27]. All communication between the forwarding device and applications is routed through the controller. Southbound APIs, such as OpenFlow, maintain communication between the application plane and the control plane. Different SDN controller devices are interconnected through east and west APIs.

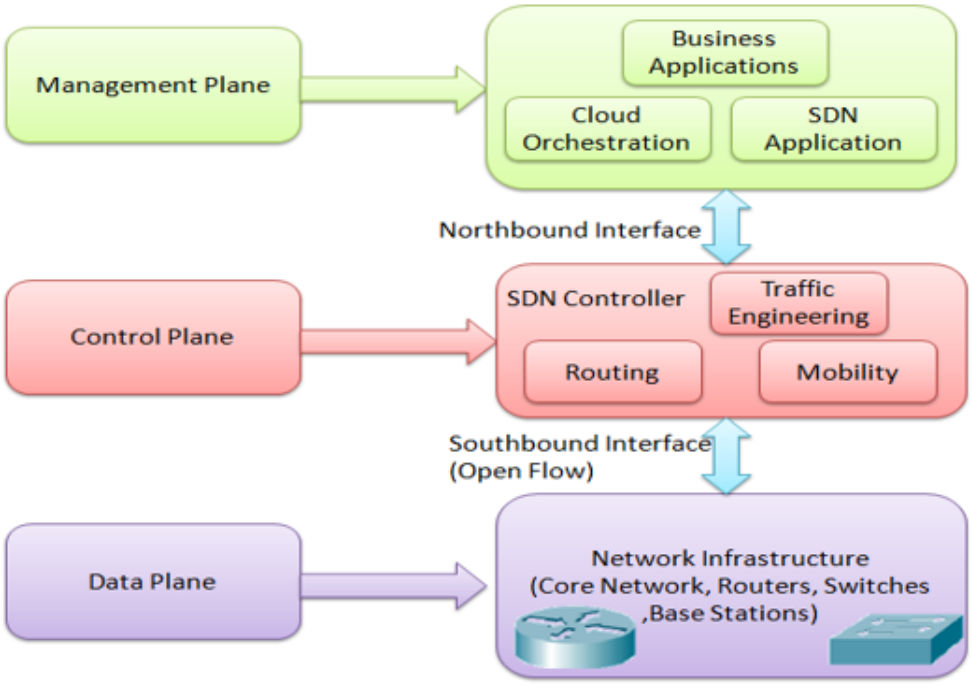

Figure 1: SDN Architecture

2.3.1 Data Plane

Network components, such as routers and switches, make up the data plane. These devices are dumb since all they do is route network data according to the controller's commands.

2.3.2 Control Plane

The control plane sees the whole network (topology of the network, the traffic statistics over the links, the link bandwidth, etc.). It uses a control logic method to route data flow in the data plane according to network needs. The data plane and the controller exchange connection statistics on a regular basis. POX, NOX, ONOS, Open Daylight, Ryu, and Floodlight are among the controllers being developed.

2.3.3 Management Plane

It consists of a number of apps (for example, a load balancer, a security module, and so on) that are used to describe network requirements. The low-level instructions are utilized by the southbound API, such as the OpenFlow protocol, which is utilized by the controller to govern the data plane. It makes it hard for a network administrator to configure and specify the network requirements using the Open Flow protocol's low-level commands. A network administrator can provide network configurations and control at an abstract level using the application plane. Routing, firewall, and load balancer, for example, are control programs used by network administrators to express network needs at a high abstract level. Pyretic, Frenetic, Maple, and PGA are some SDN languages suggested by the research community to express network applications at the abstract level.

2.3.4 Southbound Interface

This interface is used to send and receive data between the data plane and the control plane. The most often used open-source protocol for the southbound interface is OpenFlow. The data packet is sent by the standard data plane depending on the destination IP address. The SDN data plane, on the other hand, uses generic forwarding that is based on source and destination MAC addresses, source and destination IP addresses, VLAN number, input switch port number, and transport-layer port numbers. It gives us the option for enforcing traffic engineering in a variety of ways.

2.3.5 Northbound Interface

The northbound interface connects the application and control planes. A network administrator can configure the network at a high abstract level using the application plane.

The OpenFlow protocol's low-level instructions are used by the SDN controller to regulate the data plane. As a result, the northbound API is used to transform network configuration supplied at the application plane into low-level OpenFlow commands that the controller can comprehend, and vice versa.

2.3.6 SDN Advantages

Compared to traditional networking architecture, SDN provides a number of advantages.

- SDN makes network management and control straightforward due to its centralized nature. If a new routing protocol has to be installed, for example, it may be done in software at the SDN controller without affecting the underlying switches.

- SDN lowers the cost of a network in the following ways. Because the whole control logic is implemented in the software's centralized controller, the SDN switches are dummy devices. The SDN devices use the open-source OpenFlow protocol, which makes debugging and testing the protocols easy. This lowers infrastructure costs while increasing user trust that network devices are functioning properly.

- The centralized SDN controller gives more control over traffic engineering by enabling the use of more granular parameters.

2.3.7 SDN Challenges

Despite its many benefits, SDN poses a number of research challenges, such as controller scalability, effective traffic engineering, SDN adoption in practice, fault tolerance, and so on. In SDN, one aspect of fault tolerance is how to manage connection failure. Due

to congestion, port downtime, or any other reason, a connection may go down regularly. The present ways of dealing with link failure may be split into two types, reactive and proactive [28].

When a switch detects link failure, it alerts the controller in the reactive manner. The controller starts by running the routing procedure to find an alternate path for the flows that are affected by the connection loss. Second, the controller installs the alternate path in the relevant switches for each impacted flow. Third, data packets from impacted flows begin to be sent to their intended destinations. The delay in telling the controller is the fundamental drawback of the reactive failure technique. The controller then computes alternate pathways for the impacted flows, which are then installed. As a result, the impacted flows will begin forwarding after a significant delay.

To avoid the delay, proactive link failure measures are provided. In the Proactive method, the controller creates numerous data plane channels for a flow. In the event of a link failure, the switch redirects the flows of the failed link across the alternative paths preinstalled in the data plane. These techniques do not consult the controller in the event of a link failure, allowing them to recover fast.

However, these methods use a lot of a switch's Ternary Content Addressable Memory (TCAM) memory. When numerous routes are deployed, the TCAM memory quickly fills up. In the TCAM of an SDN switch, for example, about 1500 flow rules can be installed.

2.4 Load-Balancing at Fog Layer: Related Work

Cloud computing offers major support for future intelligent systems. In addition, several studies show support for the creation and deployment of edge and fog computing

units in networks, in particular for IoT, IoV, and smart city networks [5,29,30]. Moreover, fog computing is the most appropriate medium for IoV applications where a very complex mobile environment exists. It was introduced by CISCO as a form of edge computing, but later studies have made further improvements in terms of its growth and integration into intelligent systems. Many IoT application frameworks in combination with fog computing have also been successfully implemented [31]. Therefore, fog computing can be deployed in proximity to users to meet their needs with minimal Internet infrastructure support. Several recent studies have used fog computing in IoV environment [32–34]. In [10], the authors have proposed a solution called Fog Following Me (Folo) to minimize the latency in vehicular fog computing. The main idea behind the Folo design is its support for vehicle mobility.

The scalability of a vehicular network using fog computing, on the other hand, remains an open issue. In this context, SDN-based architecture can provide a scalable solution [35–38]. Nobre et al. [35] focused on the design concepts for fog-enabled Vehicular Software Defined Networking (VSDN), with an emphasis on systems, networking, and services. In [36], the authors proposed an adaptive approach for multihop routing in a complex wireless environment. The model also aids in the development of an SDN-enabled topology, which simplifies the management of SDN switches. In the context of a complex wireless network environment (i.e. IoV), sending data to the fog using an SDN-based clustering approach has been proposed in [36]. Moreover, the authors also ensure the trust of the cluster head node to provide secure communication in their proposed model.

In the area of fog computing, task offloading in conjunction with load balancing is often considered a resource management approach [39]. More specifically, the main goal of

load balancing in fog computing is to delegate a fog node to a task effectively such that a range of objectives can be accomplished (i.e., maximizing hardware utilization, minimizing latency). In [40], the authors have proposed a scheduling algorithm for fog-based IoT systems. Their findings have shown that minimizing the latency in IoT infrastructures can be achieved. Meanwhile, in [41], the authors showed that distributions of tasks among fog nodes based on information such as resource consumption, response time, and fog node location can be optimized. In addition, load-balancing algorithms and techniques have been extensively researched in cloud computing. In [42], the authors have proposed a resource intensity-aware load-balancing (RIAL) model for cloud data centers. RIAL moves virtual machines from overburdened physical servers to under-burdened ones. RIAL-based algorithms can be used in fog computing to enable the execution of specific IoV applications. In comparison to cloud computing, fog computing must take into account the spatial distribution of fog nodes. As a result, SDN is critical to the future of fog computing in terms of providing global network information.

2.4.1 Shortcomings in the Current State-Of-the-Art

The above mentioned approaches have proposed different solutions to utilize the available resources efficiently on the fog layer. Some of these approaches [43–45] were focused on developing trust by verifying application identities, and to assign the permission level. However, these types of prepositions are not suitable in a dynamic mobile network like SD-IoV. This work aims to go further than these works by employing a reinforcement-learning based solution for a dynamic and distributed SD-IoV environment to efficiently utilize the resources of fog computing layer.

2.5 PKI-Based Secure Communication: Related Work

Many articles have recently been proposed that use factor-based authentication such as articles [46–49], which use two factors, and articles [50–53], which use three factors authentication protocols. On the other hand, complex cryptographic calculations are used in several sophisticated authentication methods. However, neither the IoV devices nor the fog nodes have the necessary hardware to conduct such a complex cryptographic calculation [54,55]. Moreover, Kerberos scheme [56] conspires to work on the trusted third party, whereas the mainstream techniques like Diffie Hellman [57] uses shared secret keys for key exchange. Since nearly all cryptographic calculations rely on keys, these approaches impose additional overhead to gain more management control. There are a variety of ways to verify a user's identity, and one of them is through the use of a secret key or password. However, it has been discovered to be vulnerable to well-known dictionary attacks. Alternatively, *biometric authentication* might also be a good option. Nonetheless, since its security correlates to time complexity, it takes significantly longer to execute [58].

A biometric, password-based authentication solution for wireless sensor networks proposed in [59] is intended for achieving privacy preservation in the cloud. The difficulty with sensor nodes or any mobile devices, as mentioned in [60, 61], is that they are equipped with less battery power. As a result, energy efficiency and security become significant issues. In addition, Li et al. [62] proposed a cloud-computing-based architecture with an authenticated key agreement mechanism. However, this system is vulnerable to smart card theft, replay, and privileged insider attacks.

The authors in [63] proposed a PUF-based lightweight authentication mechanism for

RFID tags. Tag recognition, verification, and update are the three transactions that make up the protocol. The tag reader identifies the tag in the first transaction. The second transaction is the verification, in which the reader and the tag check each other's validity. For the next verification, one should maintain track of the most recently used key from the previous transaction (Update). The authors of [64] proposed a lightweight two-factor authentication technique based on one-way hashing and XOR operation to enable authentication for IoT systems integrated into cloud computing environments. There are three phases to the authentication process: registration, verification, and password renewal. The cost of computing such a system is evaluated, and its efficiency in resource-constrained situations is demonstrated.

Ahamed et al. [65] presented an anonymous mutual and batch authentication (EMBA) system that allows the RSU to authenticate many cars at the same time. The system also enables RSU-to-vehicle batch authentication, which allows n cars to be authenticated in a single batch. However, after the initial deployment, dynamic vehicles and RSU augmentation stages are not enabled in their design. A fog computing-based distributed and secure key management and user authentication method called SAKA-FC has been presented in [66]. The utilization of lightweight operations like bitwise exclusive-OR and a one-way cryptographic hash function is a significant benefit of SAKA-FC. Also, fog computing shifts the centralized cloud computing-based architecture to a distributed manner.

2.5.1 Shortcomings in the Current State-of-the-Art

While the researchers have created several authentication techniques to date, a suitable distributed and reliable system that meets all requirements such as low latency, cost-effectiveness, and lightweight has yet to be recognized and implemented. In this work,

we aim to address not just the issues highlighted by centralized authentication systems, but also to manage the heterogeneity of devices in IoV networks, and their associated data while staying resistant to malicious attacks. Particularly, this work aims to investigate the privacy and security concerns in IoV networks using a secure registration and authentication scheme.

2.6 Privacy-Preservation and Trust-Based IoV: Related Work

With the widespread use of AI in the development of next-generation applications [67–69], the growth of AI across different disciplines raises concerns regarding the use of AI technologies that are human-centered, such as in IoV, where driver security and privacy are at risk. The concerns that arises frequently when dealing with such systems may include, but not limited to, “Can the non-adversarial locally trained model provided by vehicles be trusted?", “Can the vahicles have confidence in using the cloud global model for inference?", “Is the local model genuinely trained by the vehicles?". All these concerns are the accountability challenges that are currently faced by AI systems such as FL.

Many studies have been conducted to address the issue of FL system accountability, with the bulk of them relying on blockchain for its integrity and traceability. For example, Bao et al. [70] proposed using the FLChain to create an auditable decentralized federated learning system that rewards honest trainers while detecting fraudulent nodes. Kang et al. [71] devised a trustworthy worker selection strategy, to protect against faulty model updates, which leverage blockchain for trainer reputation management. Zhang et al. [72] proposed a blockchain-based federated learning solution, for IIoT device failure detection. The method used a merkle tree to capture client data and preserve it on a blockchain,

which ensures client data's traceable accountability and integrity. Kim et al. [73] developed a blockchained federated learning architecture, for the interchange and verification of local model changes.

Since the advent of new machine-learning technology such are Deep Neural Networks or reinforcement learning, they are also applied to address the problem. For example, in [74], a deep reinforcement learning (DRL) was used by the authors for job offloading and transmission scheduling. However, new security concerns such as cyber stealth assaults [75] have emerged, necessitating additional security criteria [76] for preserving data privacy during the sharing process. The authors suggested a blockchain-enabled efficient data collecting and trustworthy sharing scheme in [77], which used Ethereum blockchain with DRL to produce a trustworthy and secure environment. In all these studies, the enabling technology was the consensus protocols, which is a key component in achieving consensus among all participating nodes. The miner who solves a mathematical challenge first earns the right to build a block in proof-of-work (PoW) [78]. However, PoW-based consensus procedures are limited in their applicability due to the high resource requirements for solving those puzzles.

In addition, fog computing, for example, is a suitable option for IoV instead of using cloud computing because it has a distributed design that minimizes latency. However, fog computing provides significant issues in terms of privacy protection in real-world applications [79]. Hu et al. [80] designed a unique Identity-based method to tackle device-to-device and device-to-server communications in fog computing. However, because the system's master key is stored in every end device, there are major risks [81]. Gu et al. [82] proposed a customizable privacy protection strategy that uses a Markov decision method

to enable customized privacy-preserving data exchange. This allows for confidential data transfer between end devices and fog servers, meeting a variety of privacy requirements.

2.6.1 Shortcomings in the Current State-of-the-Art

To the best of our knowledge, no previous research has looked at the usage of blockchain and federated learning together, in the context of effectively privacy-preserving communication in a fog-cloud computing scenario, while incentivizing vehicle (based on honesty) to improve the cloud's overall model accuracy.

CHAPTER 3 FAST AND SECURE COMMUNICATION IN INTERNET-OF-VEHICLES

3.1 Introduction

The Internet-of-Vehicles (IoV) is a highly dynamic network architecture consisting of connected vehicles, RSUs, users, and other smart devices or "things." In IoV, vehicles are essential nodes, and users are the humans involved in the system such as drivers, passengers, and even roadside pedestrians. Vehicles are also equipped with a range of sensors that generate a great quantity of data. The data can be kept locally (for example, in a vehicle's internal storage) or transferred to a distant node (for example, a fog or cloud) for processing, storage, or analysis. When sent to a vehicular cloud, the data yields valuable information, which is then passed on to each vehicle connected into the IoV network. However, the data transmission between the cloud and vehicles could result in a high end-to-end delays (or latency). This is due to the fact that the cloud is located far away from the nodes where the data is created in the network topology (i.e., vehicles in IoV). Therefore, fog nodes are deployed in proximity to the vehicles, which provides services such as code-offloading, and minimizes the latency.

Fog computing is a fully virtualized infrastructure that connects end-devices to conventional cloud computing data centers, and provide compute, storage, and networking services. In the proposed approach, the fog nodes are used to train a machine learning (ML) model using Federated Learning (FL), which preserve the users' privacy. Moreover, in order to decrease the cloud's dependency and reduce task processing latency, our method efficiently incorporates cloud computing and fog computing, and integrates software defined networking (SDN). Through decoupling the data plane and control plane, SDN manage

the IoV network in a structured, centralized, and customized form. Since a fog node is resource-constrained that may not be able to handle large amounts of data efficiently, distributed processing in the fog layer is needed. As this chapter will present, we employ machine learning (ML) techniques to efficiently allocate resources (i.e., load balancing) available at the fog nodes. This will result in minimizing latency by preventing network congestion and providing a fast mechanism for data transmission in the IoV network. In addition, this chapter will present, a reliable, cost-effective, and distributed authentication system for access layer of the IoV network using fog computing. We'll also talk about how blockchain ensures trust when ML model updates are transferred between fog nodes and the cloud. We begin with describing the system architecture of our method as follows.

3.2 System Architecture

Our proposed approach uses a three tier architecture (fog-blockchain-cloud), as shown in Figure 2. The system architecture is based on SDN-based fog layer integrated with the blockchain and cloud layers to achieve reliable, fast and secure communication in the IoV environment. More specifically, the system architecture is designed to distribute the tasks (originating from vehicles in the perception layer) efficiently to the fog nodes and to serve the vehicles with very low latency along with providing security. In general, the IoV perception layer consists of vehicles, road side infrastructure and provides vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) communications. Also, the vehicles collect road and sensors data and send to the fog servers for model training using FL. The high-level system architecture in Figure 2 involves the following four layers.

1. **Perception Layer:** The sensors and actuators attached to vehicles, road-side units

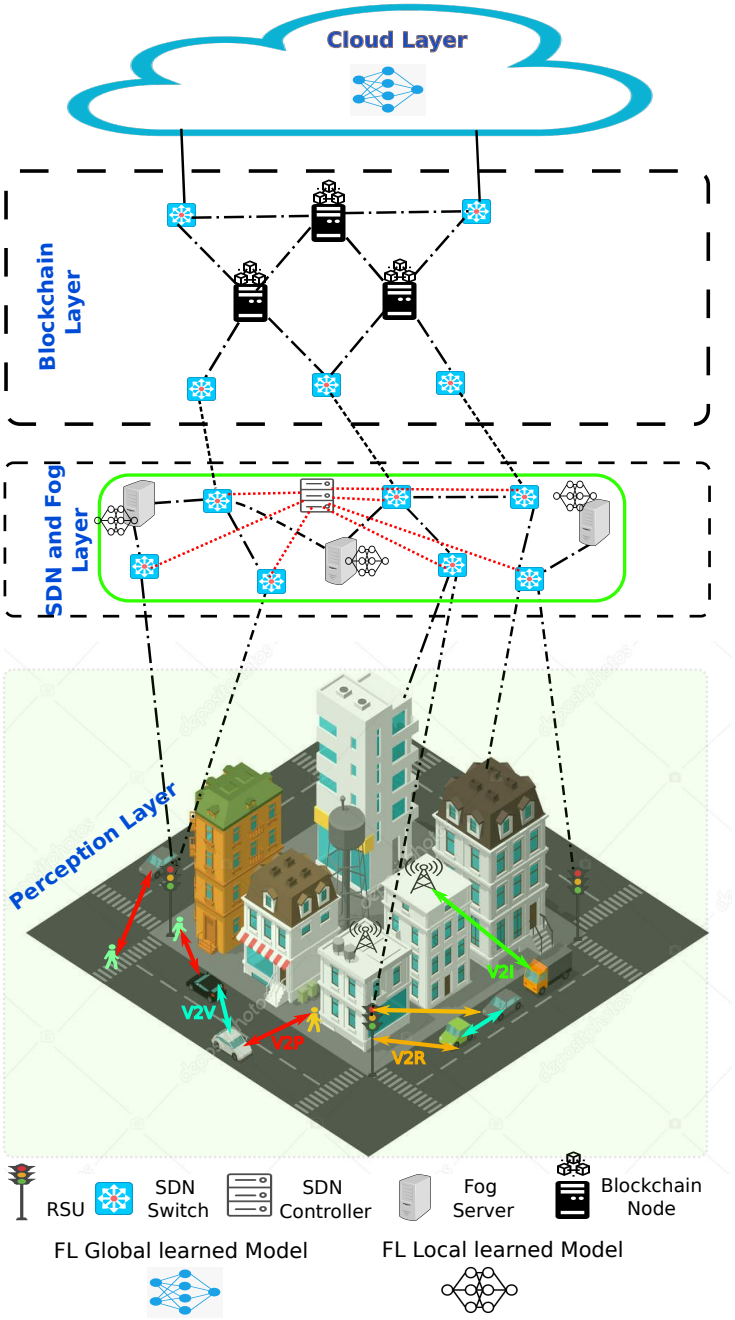

Figure 2: System architecture.

(RSUs), and other devices connected to the IoV network make up the first layer of the architecture [83]. Onboard units (OBUs) connect different sensors, a navigation system, and an entertainment system, among other things, within a vehicle. The

vehicles communicate with each other in a P2P manner, and also with the RSUs, infrastructure, and pedestrians. Moreover, the vehicles collect data to conduct the FL process, in the perception layer. The vehicles observe various driving routes and capture varying amounts of environmental data as they move. The vehicles are connected to a serving RSU that makes an RSU zone. The RSU zone may be wide in terms of its service area in places where traffic is sparse (i.e., motorways). There may be many RSU zones serving heavy traffic in urban areas (i.e. city centers). The communication technologies such as Wireless Access in Vehicular Environments (WAVE) or WiMAX/3G/4G are used for different types of communications.

2. **SDN and Fog Layer:** This layer contains stationary modules such as fog servers, openFlow switches and SDN Controller. The fog servers are connected to the vehicles through the RSUs. The fog servers can give real-time compute and networking services to the vehicles in the IoV. At least one fog server serves each RSU zone in the perception layer. In addition, the fog server in the RSU zone is configured to act as a registration authority, which is used to register IoV devices (such as vehicles and RSUs). Also, the fog servers receive training data or trained model parameters from vehicles in the perception layer to train/update their individual *FL local learned model*. Moreover, the network core in this layer consists of OpenFlow switches based on SDN to form the network topology. SDN-based switches are attached to the SDN controller, and they are responsible for delivering network status reports to the controller in real-time. The controller maintains a logical and high-level view of the network topology. This is important for load balancing to be carried out to minimize

latency and allow efficient use of the resources available at the fog servers.

3. **Blockchain Layer:** A consortium of blockchain nodes form the blockchain network, which create a separate layer in the system architecture, as shown in Fig. 2. The blockchain network is responsible for ensuring trust between fog servers and the cloud when they share the ML model updates. The blockchain transaction is used to transfer local learned model updates that have been trained by vehicles or fog servers. Through a consensus mechanism, all peers in the blockchain network audit the uploaded learning models, which are then stored in a tamper-proof ledger. The blockchain network also uses the Algorand consensus mechanism, which is based on Proof of Stake (PoS) and Byzantine fault tolerance (BFT) [84]. The BFT algorithm enables the Algorand protocol to commit transactions. Therefore, the blockchain maintain a distributed and encrypted record of all the transactions, which is hard to modify without the agreement of the overall network.

4. **Cloud Layer:** The cloud is at the very top of the system architectural hierarchy. The cloud layer provides large compute and storage resources that can be used for big data analytic and decision-making. Moreover, the cloud obtains the local learned model parameters from the blockchain and updates the *global learned model*, as shown in Fig. 2. This model is then available for inference, i.e., creation of live dashboards that decision-makers can use to track data and make strategic decisions [85, 86]. Furthermore, the cloud maintains long-term data such as RSU positions, car registration numbers, and owner information, among other things. Important temporary data, including as session keys, is also stored in the cloud so that

a vehicle does not have to restart the registration procedure when it enters a new RSU zone. In Section 3.3.3, we go over the registration procedure.

3.3 System Modeling

In this section, we explain different parts of the proposed approach from the mathematical modeling perspective. We start by explaining the load balancing using reinforcement learning (RL) in a fog computing-based IoV environment. Fog is made up of distributed, integrated, and small-scale virtualized data centers known as fog nodes (FNs) or fog servers, and it provides cloud-like services to adjacent vehicles. The RSUs connect vehicles to FNs, which connect to the cloud, forming a three-tier architecture (perception-fog-cloud) that allows for task offloading and fog-to-fog resource sharing. However, since fog nodes (servers) have limited processing power, allocating computationally intensive tasks to them and balancing the entire load for delay reduction depending on performance and communication overhead is a challenging task.

3.3.1 Load Balancing: Using Reinforcement Learning in the SDN Controller

In fog computing, load balancing is defined as efficiently distributing the arriving packets around a cluster of fog nodes for processing. Generally, in a high dynamic environment such as IoV, the main objective of designing a load balancing algorithm should be to maximize the efficiency as well as the capacity of concurrent requests received from the vehicles in the perception layer. Previous works [87–91] have considered task offloading for mobile nodes to minimize the overall cost of energy, computation, and delay. However, these works can not be applied to dynamic environments such as IoV, where the links' status changes very rapidly. Therefore, to minimize the total delay and reliably use the available

resources, load balancing is considered a daunting task.

In a broader context, load balancing algorithms can be broken down into two types: static and dynamic. Both types inherently are based on two different methods [92]. Static load balancing uses advanced knowledge of task requests, calculated at the outset of the execution, to spread the workload. The key downside to static methods is that during the execution of the operation, the distribution of tasks cannot be modified to accommodate traffic load changes. In comparison, as one node becomes under-loaded, dynamic load balancing automatically allocates tasks. In other words, based on the current information of traffic loads, it will constantly change the allocation of tasks. Therefore, for efficient load balancing, effective real-time load prediction is established.

A multi-agent-based approach can be formulated to accomplish dynamic load balancing, as in [93], where the fog nodes can be picked as agents - to make a series of decisions using *reinforcement learning* (RL). Specifically, RL is concerned with learning to solve a problem by trial and error. The agents can be designed to act in the IoV environment and be rewarded for it. A Markov Decision Process (MDP) based model, motivated by recent progress in implementing RL techniques, can be used for synchronization of the multi-agents (i.e., fog nodes). Furthermore, synchronization is needed for multi-agents to maintain the IoV network's overall QoS.

Synchronized multi-agents We consider an IoV network (i.e, as in Fig. 2), and assume that a fog node fn^i can serve a vehicle better than another node fn^j in terms of computing and total delay costs. Furthermore, fog nodes (agents) may not be synchronized due to a high-dynamic IoV environment. As a result, a task from the vehicle is delegated to fn^j,

based on obsolete information, which will lead to performance loss. Therefore, we address the problem of load balancing by formulating it as MDP, using which the synchronization between nodes (multi-agents) is achieved.

More precisely, MDP is represented by a tuples (S, A, P, R, γ), where S and A are sets representing the finite states and actions respectively, P and R are state transition probability and reward functions respectively, and γ is a discount factor such that $\gamma \in [0, 1]$. For a total of n states (where $s \in S$), the range is from s_0 to s_n. Similarly, $a \in A$ is the total actions. The transition function for each state is represented as, $P : S \times A \mapsto \Delta(S')$. It gives the probability $P(s_{t+1}|s_t, a_t)$ of taking an action a_t in state s_t that leads to the next state s_{t+1}. The reward function R is, $mathbbE_R[R(s_t, a_t)|s_t, a_t]$.

Load balancing at the fog servers Our proposed model is tailored to provide the best offloading action possible, maximizing utility while reducing processing time and load balancing. As a result, given an action a at state s, we define the immediate reward function $R(s, a)$ as follows.

$$R(s, a) = \mu(s, a) - (\beta(s, a) + \delta(s, a)), \tag{3.1}$$

where, $\mu(s, a)$ denotes the utility function, $\beta(s, a)$ denotes the traffic load probability function, and $\delta(s, a)$ denotes the fog node's end-to-end delay function. More specifically, we define the utility function $\mu(s, a)$ as,

$$\mu(s,a) = \theta_u log(1 + k^o),$$

where, θ_u and k^o indicate the utility reward and number of tasks to be offloaded to a fog node, respectively.

Furthermore, we model the traffic load probability function $\beta\,(s,a)$ of a fog node as follows:

$$\beta(s,a) = \kappa_o \frac{k^{pro} P + \kappa_o P}{k^{pro} + \kappa_o},$$

where, κ_o and k^{pro} indicates the traffic load weight and the processing tasks, respectively. Also, the SDN controller uses this function. P can be modeled by using a *Poisson* process as,

$$P = \frac{\max(0, \sigma - (Q_{j,max} - Q_j^{'}))}{\sigma},$$

where σ indicates the rate for the arrival of tasks at a fog node. The next estimated queue state is $Q_j^{'}$, given a state s and action a. It can be written as,

$$Q_j^{'} = min(max(0, Q_j - c_p wr^j) + k^j, Q_{j,max}).$$

Moreover, the SDN controller also calculates the end-to-end delay for a task, when it is sent to a fog node and the result is returned successfully. The delay function $\delta\,(s, a)$ of a fog node can be modeled as,

$$\delta(s, a) = \kappa_d \frac{d^t + d^q + d^e}{k^o} + k^{pro}, \tag{3.2}$$

where, d^t, κ_d, d^q, and d^e indicate transmission delay, delay weight, queue delay, and execution delay, respectively.

3.3.2 Secure Communication: Using Public Key Infrastructure (PKI)

In this section, we'll look at how fog computing can be used to create a reliable, cost-effective, and distributed authentication solution for IoV networks. We've assumed that the registration center (also known as the fog server) is totally reliable. In addition, cars and RSUs will be provided their public keys ahead of time, which they will use to register with the IoV network in order to utilize the services. The architecture of the secure data access mechanism in an IoV network is depicted in Fig. 3. A data storage cloud service provider is included in the model, as well as a fog server in the RSU zone that acts as an IoV device registration authority. The fog server can provide real-time data, while the cloud can provide access to permanently archived historical records. In order to use an RSU zone, a vehicle and an RSU must first register with the registration authority (fog

server), which is assumed to be secure and dependable. Every time the RSU or another vehicle (through the RSU) needs to access the vehicle's essential data, the consent must be sought beforehand.

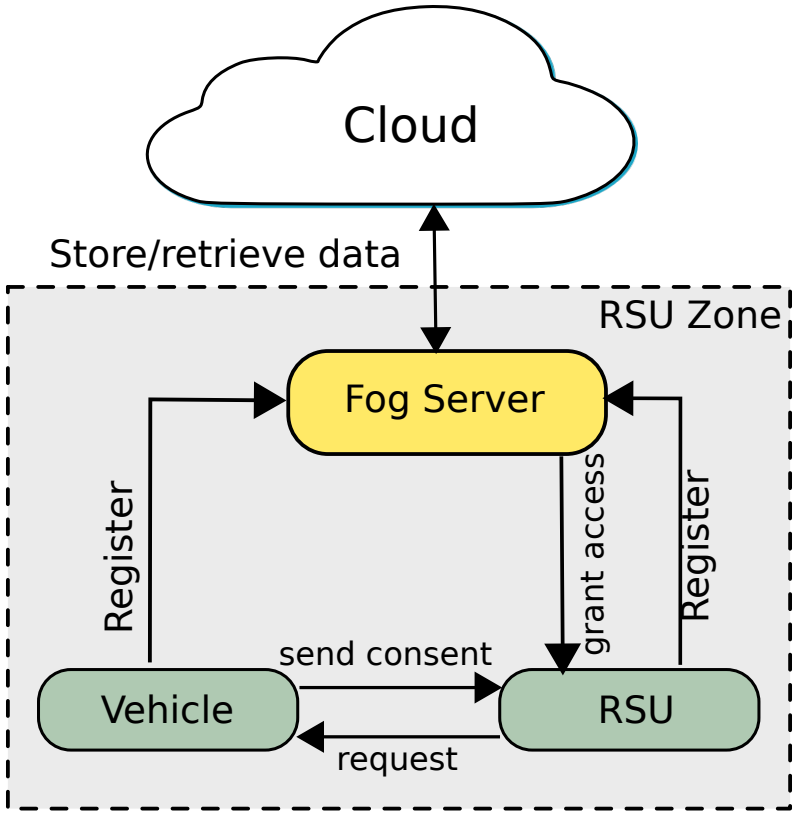

Figure 3: Flowchart for requesting vehicles data.

If an RSU ($Rpki$) wants to obtain current or real-time data from a vehicle in its zone (stored on the fog server) or historical data (stored on the cloud server), it must first make sure it's a genuine RSU. After adequate verification, the vehicle will authorize the request, and the linked fog server will provide access.

We'll go through the mutual authentication technique in the IoV network in the rest of this section. RSUs and users (i.e., vehicles) will be able to authenticate one another, and RSUs will be able to connect with vehicles (i.e., real-time data exchange) or safely access

historical data from the cloud server (with proper consent). The suggested authentication technique is made up of three phases. The initial step is to register IoV devices (i.e., RSUs and Vehicles). Mutual authentication is the second phase, in which the devices authenticate each other before utilizing the services. During the authorisation process, the cars allow RSUs to interact with them or access their data in the cloud. Before we go into the details of these stages, we will discuss the security requirements and notations, as follows.

Security Requirements Information is valuable in today's world, thus it must be treated with care during its path. The security of communicating devices, as well as data integrity, communication, and storage in the IoV network, must all be maintained in a secure IoV system. The following are the most important problems that need to be addressed:

- Passengers' Privacy and Location: To preserve the privacy of drivers and passengers within vehicles, some restrictions should be implemented, such as providing only authorized users access to the resources for which they have been granted permission. It's also crucial to keep track of where the vehicles are. Furthermore, vehicles would want to send their personal information to the nearest fog server. If the fog server is hacked, the hacker will have access to user data, vehicle information, and the vehicle's location in the worst-case scenario. As a result, non-real characteristics, such as pseudonyms, are used to show the identity of IoV devices. We verify that the non-real characteristics of IoV devices have no bearing on their genuine identification.

- Authentication: Before communicating with any IoV device or fog server, make sure it's legitimate. This is done to prevent unauthorized access to sensitive data, and

only authenticated devices have access to restricted resources. As a result, the parties involved in the communication should be verified. Furthermore, because IoV applications operate in a highly dynamic environment, authentication systems that can manage and adapt to the growing number of IoV devices (such as vehicles, RSUs, and other IoV devices) are necessary. As a result, while creating an authentication system for the IoV, scalability is also a critical element to consider.

- Data Storage: It's vital to prevent such a massive volume of data from being manipulated with or distorted as sensors linked to vehicles and RSUs continue to generate data.

Notations Each RSU zone will have its own registration authority, which will be under the supervision of the fog server. All security parameters will be accepted as input by the registration authority, and system parameters will be created as a consequence. It will also compute the private key (K_{pvt}) and public key (K_{pb}) as follows:

$$\mathbb{K}_{pvt} = (s \# ts) \cdot P,$$

$$\mathbb{K}_{pb} = h\left(s \oplus ts\right),$$

where, s is fog server secret key, P is 512-bit prime generator, and ts is timestamp. Moreover, the registration authority chooses Γ_1 and Γ_2 of order $\mathbb{R}$ with the generator as described in [94]. Also, the cryptographic hash functions are calculated as [95],

$$H : \{0,1\}^n \rightarrow Z_q, \tag{3.3}$$

where n is the bit-length of the plain-texts. Finally, the registration authority publishes $\{\mathbb{K}_{pb}, P, \mathbb{R}, \Gamma_1, \Gamma_2, H\}$. The details of these notations are listed in Table 1.

Table 1: Notations and their description

Notation	Description
$\mathbb{K}_{pvt}$	Private key of the fog server
$\mathbb{K}_{pb}$	Public key of the fog server
s	Secret key of the fog server
R_{pk_i}	Public key of an RSU
VId_{R_i}	Virtual ID (pseudonym) of an RSU
υ_{pk_j}	Public key of a vehicle
VId_{υ_j}	Virtual ID (pseudonym) of a vehicle
P	Prime generator (512-bit)
Γ_1	A cyclic Additive group
Γ_2	A cyclic Additive group
h(.)	Cryptographic Hash function
$\mathbb{R}$	Large prime number (160-bit)
ts	Timestamp
$\shortparallel$	Concatenation sign

3.3.3 Registration

Both RSUs and vehicles in the IoV network are registered at this phase. It's crucial to register RSUs since an attacker may impersonate a legitimate RSU and seek data access. In order to broadcast its RSU zone to vehicles, an RSU (with public key R_{pk_i}) must also register. The RSU will do so by selecting an identity ID_{R_i} and generating two nonces n_i and n_i^*. The virtual identity (or pseudonym) is then calculated as $VId_{R_i} = (ID_{R_i} \oplus n_i^*)$ and sent to the registration authority as $E_{K_{pk}}\langle VId_{R_i}, n_i\rangle$.

The registration authority will generate a pseudonym r_i and calculate the following.

$$\mathfrak{R}_i = (r_i \oplus n_i) \tag{3.4}$$

$$VId'_{R_i} = h\left(VId_{R_i} \mathbin{+\!\!+} n_i\right) \tag{3.5}$$

$$\alpha_i = r_i \cdot P \tag{3.6}$$

$$\beta = (\alpha_i \mathbin{+\!\!+} n_i) \tag{3.7}$$

$$R_{pk_i} = h\left(VId'_{R_i} \mathbin{+\!\!+} \beta\right) \tag{3.8}$$

$$\Upsilon_i = h\left(VId'_{R_i} \mathbin{+\!\!+} r_i\right) \tag{3.9}$$

$$\delta_i = \alpha_i \oplus \beta \cdot \Upsilon_i \pmod{P} \tag{3.10}$$

$$\chi_i = \left(VId'_{R_i} \mathbin{+\!\!+} h\left(\beta\right) \mathbin{+\!\!+} ts_i \mathbin{+\!\!+} R_{pk_i}\right) \tag{3.11}$$

$$\Psi = h\left(\delta_i \mathbin{+\!\!+} r_i\right) \oplus ts_i \tag{3.12}$$

$$h_1 = h\left(\delta_i \oplus VId'_{R_i} \mathbin{+\!\!+} \Psi\right) \tag{3.13}$$

After computing the above functions, the registration authority (fog server) will store $\{VId_{R_i}, n_i, \mathfrak{R}_i, \chi_i\}$ with itself, and send $\{\left(\delta_i \oplus VId'_{R_i}\right), \Psi, h_1\}$ to the RSU. The RSU will validate the timestamp and the h_1 on the receiving end. The whole process of registration of RSU with fog server is illustrated in Fig. 4. After an RSU is registered successfully, vehicles will be registered with the fog server in order to be deemed valid devices.

Therefore, to use the services, a vehicle (υ_j) will need to register after joining the IoV network. As a result, the vehicle (υ_j) will produce two nonce, n_j and n_j^*, and choose an identity ID_{υ_j}. Then it will calculate the virtual identity that corresponds to it.

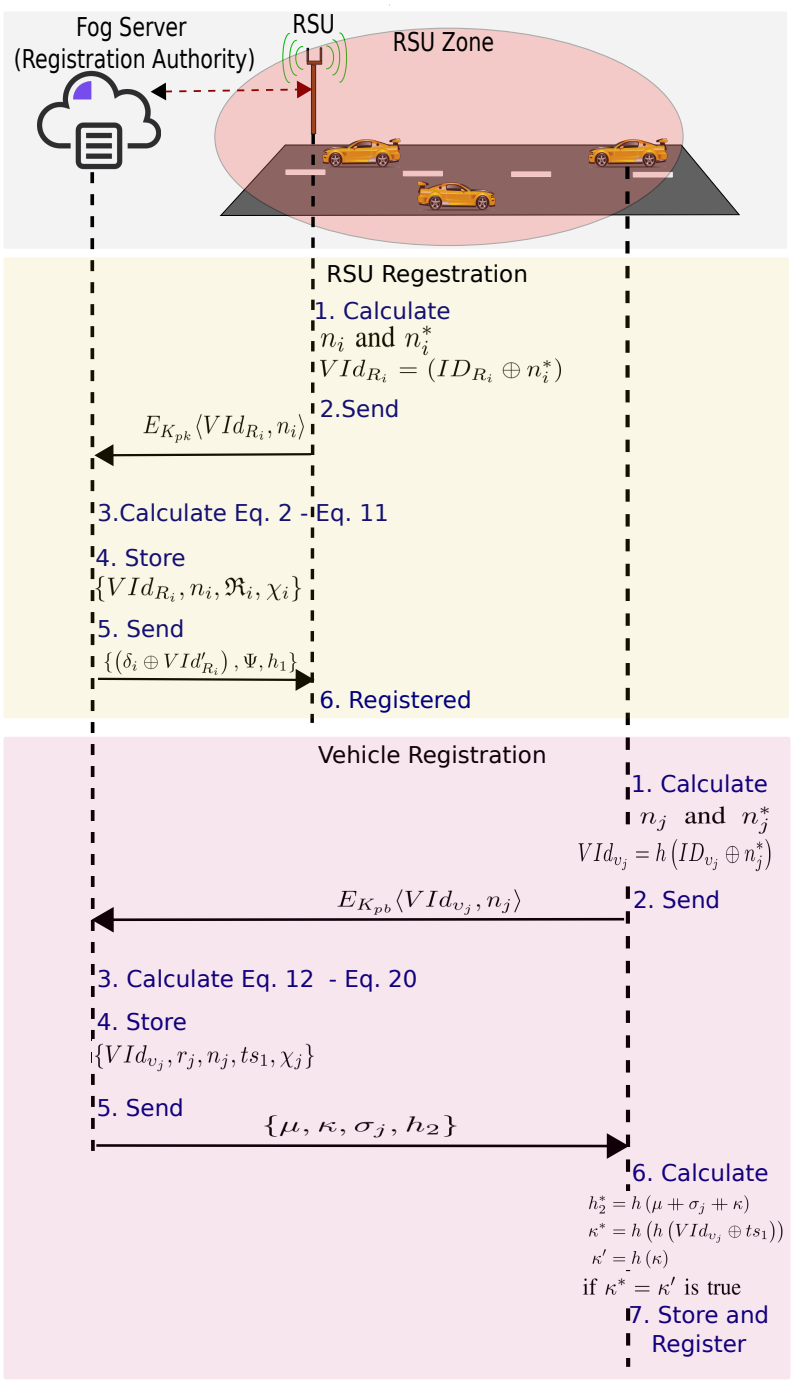

Figure 4: Sequence diagram showing the registration process of RSU and vehicle with the fog server (registration authority).

$$VId_{v_j} = h\left(ID_{v_j} \oplus n_j^*\right)$$

Next, it will send $E_{K_{pb}}\langle VId_{v_j}, n_j\rangle$ to the registration authority via any channel (secure or insecure). When the registration authority receives the message, it will create a random number r_j and compute the following:

$$VId'_{v_j} = h\left(VId_{v_j} \mathbin{\#} n_j\right) \tag{3.14}$$

$$\kappa = h\left(VId'_{v_j} \oplus ts_1\right) \tag{3.15}$$

$$\mu = r_j \oplus n_j \tag{3.16}$$

$$\alpha_j = r_j \cdot P \tag{3.17}$$

$$v_{pk_j} = h\left(VId'_{v_j} \mathbin{\#} \alpha_j \mathbin{\#} n_j\right) \tag{3.18}$$

$$\beta_j = h\left(r_j \mathbin{\#} n_j\right) \tag{3.19}$$

$$\sigma_j = ts_1 \oplus \left(\alpha_j \cdot \mathbb{K}_{pvt}\right) \pmod{p} \tag{3.20}$$

$$\chi_j = \left(VId'_{v_j} \mathbin{\#} \beta_j \mathbin{\#} ts_1 \mathbin{\#} v_{pk_j} \mathbin{\#} \sigma_j\right) \tag{3.21}$$

$$h_2 = h\left(\mu \mathbin{\#} \sigma_j \mathbin{\#} \kappa\right) \tag{3.22}$$

After the above are calculated, the fog server will store $\{VId_{v_j}, r_j, n_j, ts_1, \chi_j\}$ and send back $\{\mu, \kappa, \sigma_j, h_2\}$ to the vehicle. At the receiving end, the vehicle will calculate the hash as:

$$h_2^* = h\left(\mu \mathbin{+\!\!+} \sigma_j \mathbin{+\!\!+} \kappa\right)$$

$$\kappa^* = h\left(h\left(VId_{v_j} \oplus ts_1\right)\right)$$

$$\kappa' = h\left(\kappa\right)$$

Finally, the vehicle will check if $\kappa^* = \kappa'$ is true, then it will store the received data; otherwise, it will end the session. The whole registration process of the vehicle with fog server is shown in Fig. 4.

3.3.4 Authentication

To use the services, a vehicle must first make contact with the RSU in its RSU zone and submit a request message $msg_{request}$. In the meanwhile, the vehicle will compute the following.

$$\mu = r_j \oplus n_j \tag{3.23}$$

$$r_j = \mu \oplus n_j \tag{3.24}$$

$$\sigma_j' = \left(\sigma_j \mathbin{+\!\!+} h\left(r_j \mathbin{+\!\!+} n_j\right)\right) \tag{3.25}$$

The vehicle will then produce a new nonce, n_x, in order to do the following calculations.

$$\lambda^1_{\upsilon_j} = \left(VId'_{\upsilon_j} \oplus r_j\right) \tag{3.26}$$

$$\lambda^2_{\upsilon_j} = (ts_2 \oplus n_x) \oplus \sigma'_j \tag{3.27}$$

$$h_3 = \left(\lambda^1_{\upsilon_j} \mathbin{+\!\!+} \lambda^2_{\upsilon_j} \mathbin{+\!\!+} ts_2\right) \tag{3.28}$$

Finally, the vehicle will send $\{\lambda^1_{\upsilon_j}, \lambda^2_{\upsilon_j}, h_3, ts_3\}$ to the RSU. At the receiving end, the RSU will verify the timestamp first: $\Delta t = ts_{recvd} - ts_{sent}$. The message is deleted, and the session is ended if the computed Δt is not within the allowed range; otherwise, the verification procedure proceeds to validate the message's integrity by comparing it to the calculated hash value. When both parties (RSU and vehicle) agree, the procedure continues. Following that, the RSU carry out the following computations:

$$\alpha_i^* = r_i^* \cdot P \tag{3.29}$$

$$\beta^* = (\alpha_i^* + n_i) \tag{3.30}$$

$$\Upsilon_i^* = h\left(VId'_{R_i} + r_i^*\right) \tag{3.31}$$

$$\delta_i^* = \alpha_i^* \oplus \beta^* \cdot \Upsilon_i^* \pmod{P} \tag{3.32}$$

$$R_{pk_i}^* = h\left(VId'_{R_i} + \beta^*\right) \tag{3.33}$$

$$\chi_i^* = \left(VId'_{R_i} + h\left(\beta^*\right) + ts_2 + R_{pk_i}\right) \tag{3.34}$$

$$\zeta_a = \left(VId'_{R_i} \oplus \lambda_{v_j}^1\right) + \chi_i^* \tag{3.35}$$

$$\zeta_b = (\delta_i^* \oplus n_x) + \zeta_a \tag{3.36}$$

$$\lambda_{v_j}^2 = \lambda_{v_j}^2 \oplus ts_3 \tag{3.37}$$

$$h_4 = \left(\lambda_{v_j}^2 + \zeta_a + \zeta_b\right) \tag{3.38}$$

The RSU will send $\{\zeta_a, \zeta_b, \lambda_{v_j}^2, h_4\}$ to the fog server, which will calculate the following after receiving the message:

$$\zeta_a = \left(VId'_{R_i} \oplus \lambda_{v_j}^1\right) + \chi_i$$

$$VId'^*_{R_i} = (\zeta_a - \chi_i) \oplus \lambda_{v_j}^{1*}$$

Next, the fog server will check in its database if $VId'^*_{R_i} = VId'_{R_i}$ is true; then the fog server will create a new nonce n_y, and will do further calculations as follows.

$$\Phi_1 = h\left(\left(VId'_{v_j} \mathbin{+\!\!+} \chi_j\right) \oplus (ts_3 \mathbin{+\!\!+} n_y)\right)$$

$$\Phi_2 = h\left(\left(VId'_{R_i} \mathbin{+\!\!+} \chi_i\right) \oplus (ts_3 \mathbin{+\!\!+} n_y)\right)$$

The fog server then transmits $Z_{v_{pk_j}}\langle\Phi_1\rangle$ to the vehicle, while simultaneously sending $Z_{R_{pk_i}}\langle\Phi_2\rangle$ to the RSU. The RSU and vehicle may now mutually verify each other using the received messages. They're now ready to start the session and interact securely. The entire authentication procedure is depicted in Fig. 5.

3.3.5 Privacy Preservation: Federated Learning

In FL, the vehicles in the perception layer are connected to a serving RSU inside an RSU zone, as shown in Figure 6. The RSU zone may be wide in terms of its service area in places where traffic is sparse (i.e., motorways). There may be many RSU zones serving heavy traffic in urban areas (i.e. city centers). Let's consider the total number of vehicles registered on the IoV network is represented by X, such that a vehicle set is obtained as $|V| = X$. The whole road network in the perception layer of IoV is further divided into Z RSU zones. Therefore, the number of vehicles connected to $k-th$ RSU zone can be expressed as;

$$V_k = v_{k_1}, v_{k_2}, v_{k_3}, ..., v_{k_{x_k}}, \tag{3.39}$$

where $|V_k| = x_k$, $k \in [1, Z]$, and $x_1 + x_2 + x_3 + ... + x_Z = X$. In addition, the number of RSUs is denoted as;

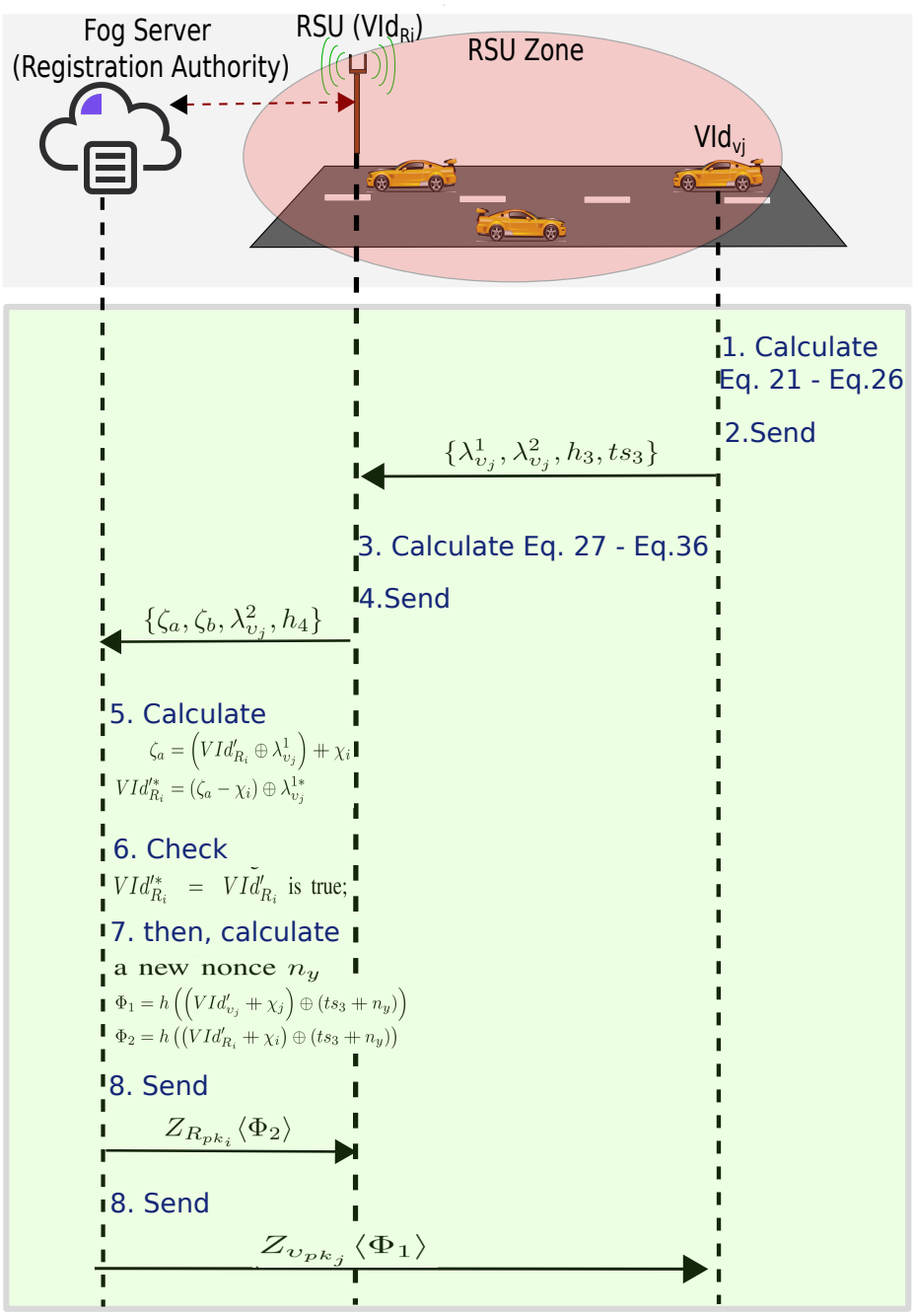

Figure 5: Sequence diagram showing the authentication process.

$$R = R_1, R_2, R_3, ..., R_Z. \tag{3.40}$$

The RSU in $i-th$ zone can be expressed as $R_i, i \in [1, Z]$, which is securely registered on

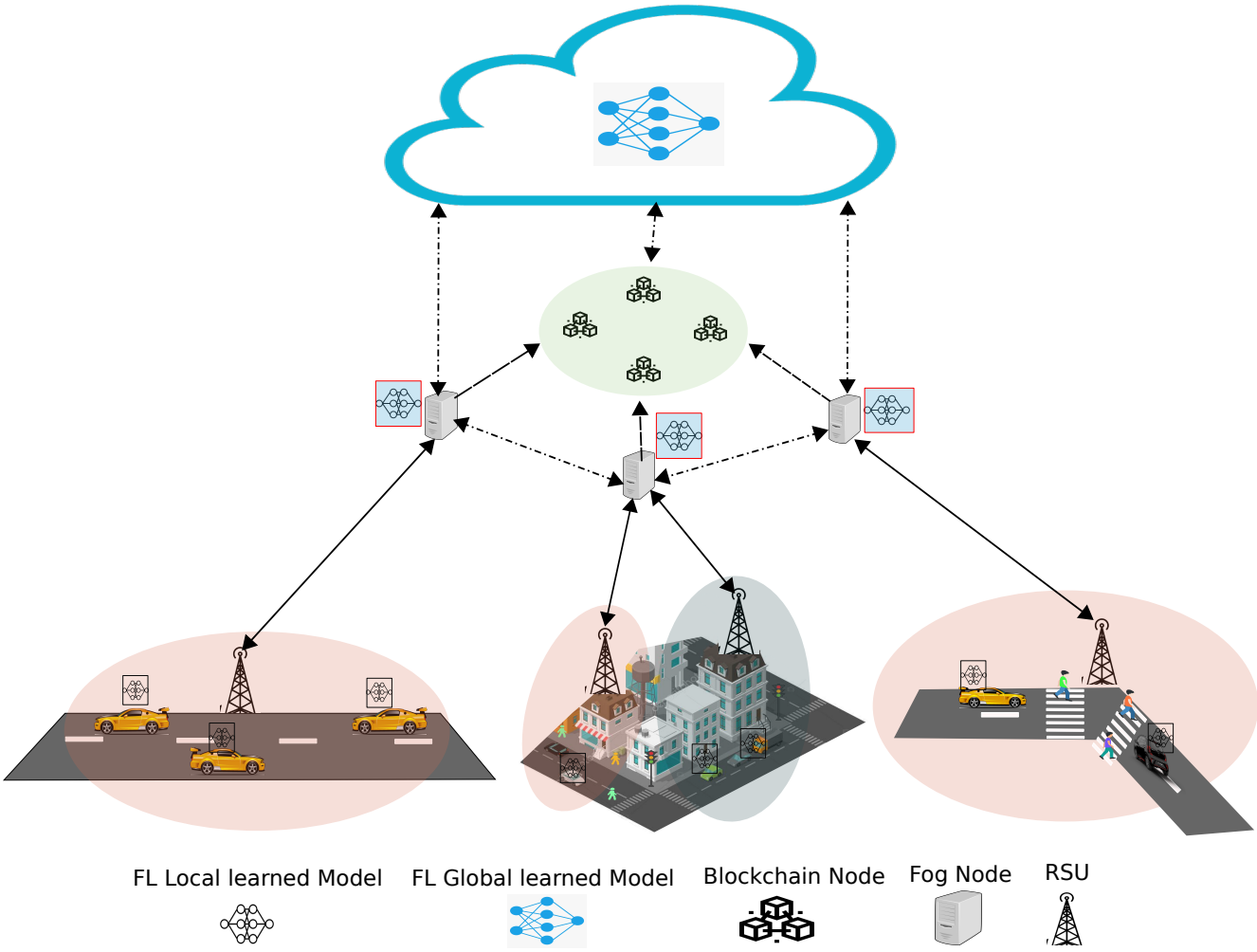

Figure 6: Federated learning System in IoV.

the network using our authentication scheme presented in 5. Moreover, each RSU $R_i \in R$ is connected to a fog node $F_j \in F$, where $i \in [1, Z]$ and $j \in [1, N]$ for $|F| = N$. Fog nodes in the set F receive training data or trained model parameters from vehicles V in RSU zones R. To update the global FL model, the fog nodes upload their learned local FL model parameters to the cloud.

In our proposed approach, the vehicles collect data to conduct a federated learning process, in the perception layer. The vehicles observe various driving routes and capture

varying amounts of environmental data as they move. In Fig. 7, we can see how the training data or model updates are sent to the fog node in case of a good network coverage. Fog nodes trained the local models and put the updates on the blockchain. The global model updates are sent back from the cloud to the blockchain, where they can be used for inference by the vehicles.

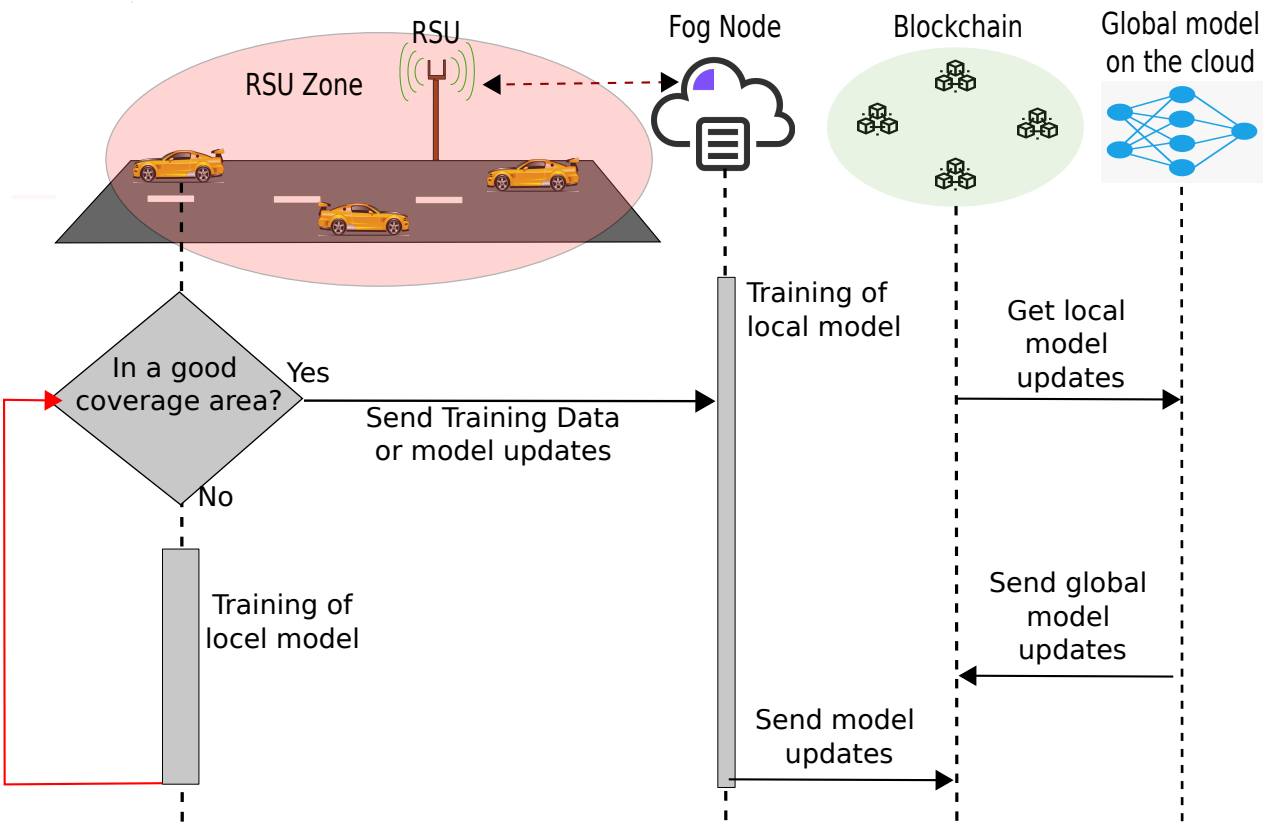

Figure 7: Sequence diagram of the workflow.

3.3.6 Trust: Using Blockchain for Model Parameters Publishing

After training their local models, the fog nodes build new blocks, which are added to the blockchain network. A consortium of blockchain nodes formed the blockchain network, which create a separate layer as shown in Fig.2 and Fig. 6. In this work, we employ the Algorand consensus mechanism, which is based on Proof of Stake (PoS) and Byzantine

fault tolerance (BFT) [84]. The BFT algorithm enables the Algorand protocol to commit transactions. The procedures below must be followed in order to establish a successful consensus.

1. A miner with more stakes has a better chance of becoming a leader. The number of stakes a miner possesses influences the likelihood of it being chosen as a leader.

2. After the leader has been chosen, it can generate new blocks. The blocks can then be verified by the participants. In practice, a new block is allowed when more than 2/3 of the members sign and agree on the leader's block.

3. To reach a consensus in blockchain, the participants broadcast the new block to their neighbors (i.e., using the gossip protocol [96]).

When the *FL-global-learned model* is published on the blockchain, vehicles that request access can use it for inference, as shown in Fig. 7. Subsequently, fog nodes publish their *FL-local-learned model* to the blockchain after training models locally. The leader and miners are in charge of verifying transactions, while the cloud calculates the averaged model parameters to generate a global model. A miner, in particular, verifies the digital signature of the model's hash published to the blockchain by a fog node. Furthermore, the miner confirms that the model parameters are originated from a valid fog node by determining whether the signature is authentic, at which point it is added to the transaction pool. The leader, who is chosen from among the miners, will then create a new block with the model parameters hash. Typically, the miners in the blockchain consortium choose the leader, and the miners compete to verify the digital signature of model parameters and get rewarded.

The Algorand protocol utilizes verifiable random functions (VRFs) to pick a group of miners as leader candidates, depending on the reward. For example, the leader candidate with the highest reward will be promoted to the position of leader. Since each minor's reward is weighted, and one award is equivalent to one coin, a minor with β coins is rated β. Moreover, α represents the expected number of participants (i.e. minors) and γ indicates the total quantity of coins divided among the participants. Then a participant m with β coins will require its secret key to make a hash and proof using VRF, as $hash/2^{hlen}$, where $hlen$ is the hash length. Mathematically,

$$hash/2^{hlen} \in \left[\sum_{k=0}^{\tau} \binom{\beta}{k} p^k (1-p)^{\beta-k}, \sum_{k=0}^{\tau+1} \binom{\beta}{k} p^k (1-p)^{\beta-k}\right] \tag{3.41}$$

where p is the probability of any coin being chosen, i.e., $p = \alpha/\gamma$, and τ represents the selected number of participants. As a result, we use E.q. 3.41 to find the interval that is used to determine the τ participant. After the leader uploads the model parameters, which are verified by the minors, the cloud derives the final FL global learned model from the blockchain network. The Global model is available afterward, to be used by the vehicles in the IoV network for inference.

3.4 Chapter Summary

In this chapter, we discussed how the proposed framework is built to achieve fast and secure communication in a high dynamic environment such as IoV. We discussed the system architecture that includes different layers connected together such as perception, fog and SDN, blockchain and the cloud. From a mathematical modeling perspective, we also explained how different parts of the framework (such as load balancing, secure commu-

nication, privacy preservation, and trustworthy communication) cooperate to achieve the goals of this work.

The importance of load balancing in fog-based IoV setting is to improve QoS, minimize end-to-end delays, and utilizes the available resources efficiently. At the SDN controller, we employ an RL-based algorithm to effectively assign tasks to the fog servers. Moreover, the vehicles and RSUs in the perception layer communicate with the fog servers over insecure channels. Therefore, authentication is undoubtedly needed. To address the problem of secure communication, we employ a lightweight and fog-based authentication key creation scheme in the IoV environment. For secure communication between RSUs and vehicles in an RSU zone, our proposed authentication scheme involves the creation of a mutual authentication session key. Moreover, our scheme uses a pseudonym mechanism to prevent any association with the actual information. Finally, in order to maintain privacy and ensure reliable communication, we use federated learning and blockchain approaches to train a global model without sharing the critical data from vehicles.

CHAPTER 4 LOAD BALANCING IN FOG COMPUTING-BASED IOV

In Section 3.3.1, we introduced how the available resources at the fog layer can be efficiently utilized. Moreover, we described our proposed approach for allocating fog servers to tasks, which is based on reinforcement learning. In this chapter, we demonstrate the effectiveness of our proposed method using simulations. Over the last few years, fog computing has developed as a contemporary approach for big data processing. Since the data is closer to the edge network, it can improve the efficiency and effectiveness of data processing. Despite the fact that fog computing has been shown to be effective in handling big data, it still faces some challenges, i.e., task capacity provisioning is a difficult problem to solve, especially when fog servers have heterogeneous characteristics and limited information between servers and tasks. Therefore, the majority of research have deployed fog resources for a given task model using machine learning-based methods. Our approach is based on Reinforcement Learning (RL) with a central management process such as SDN, where RL agent handles virtual resource allocation and task scheduling by optimizing a cost function.

4.1 Load Balancing: RL-Based Offloading Algorithm

In nature, the IoV environment is extremely dynamic, and it is thus difficult enough for the SDN controller to forecast p and R (transition probability and reward functions). Therefore, to predict the optimal policy for (s, a), we use the Q-learning-based algorithm 1 for effective task offloading. In addition, the current $s^{'}$ state, and the r reward, will be constantly learned and observed by the controller. The algorithm will update the Q-function, which is used to make the optimal decision for the new requested task (i.e., code offload-

ing), based on the updated information. The Q-function in Eq. 4.1 is stated as follows:

$$Q(s,a) = (1-\beta)Q(s,a) + \beta[R(s,a) + \alpha \max_{a}^{'} \in A_{s'} Q(s^{'}, a^{'})] \tag{4.1}$$

Where, β (i.e. $(0 < \beta < 1)$) is defined as the Q-learning rate. The reward function R is the current learning rate, which is the calculation learned from the IoV environment's underlying traffic.

Algorithm 1: IoV NETWORK LOAD BALANCING USING Q-LEARNING-BASED TASK OFFLOADING

Input: Latest (s, a)
Output: This algo returns the best fog node that is currently available.
Set i = 0 and Q(s, a) = 0
for $i \leq max_i$ **do**
    Select $a \in A$
    Continue with the offloading depending on the information gathered on a
    SDN controller perceive and learn $s^{'}$ and $r^{'}$
    The Q-function in Eq. 4.1 is updated
    s $\leftarrow s^{'}$
    i $\leftarrow$ i+1
end

4.2 Simulation Settings

For simulation, we have used the sumo (Simulation of Urban Mobility) for the IoV network. First, as illustrated in Fig. 8, we imported the road network of downtown Detroit (Michigan, US) into the sumo simulator. Next, we imported the sumo road map data and traffic configuration into the Omnet++ simulator to evaluate our Load balancing method. The density of vehicles was limited to 500. Our simulation is based on executing custom-made scenarios using the Omnet++ environment to simulate traffic data originating from vehicles to fog servers. We used an SDN controller, which obtains a global view of the IoV

network and implements Algorithm 1 for efficient traffic distribution among the fog nodes. The Flowchart in Fig.9 shows the simulation setup. We compare the results with a naive approach that always allocates the closest possible resources (fog node) to offload the vehicle's tasks. Therefore, in the naive approach, the load balancing script is not enabled, while we enable it in the fog nodes and the controller in our method.

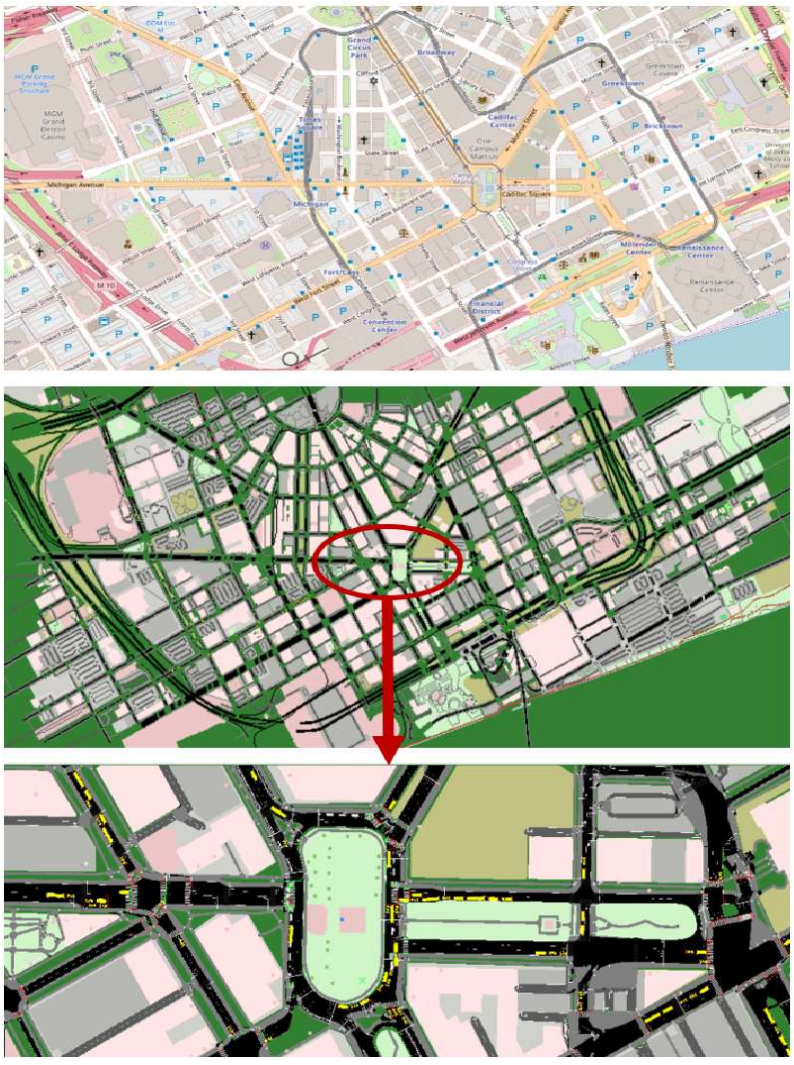

Figure 8: Downtown Detroit road map (OSM) imported in Sumo simulator. A zoomed view of a region shows traffic on the roads.

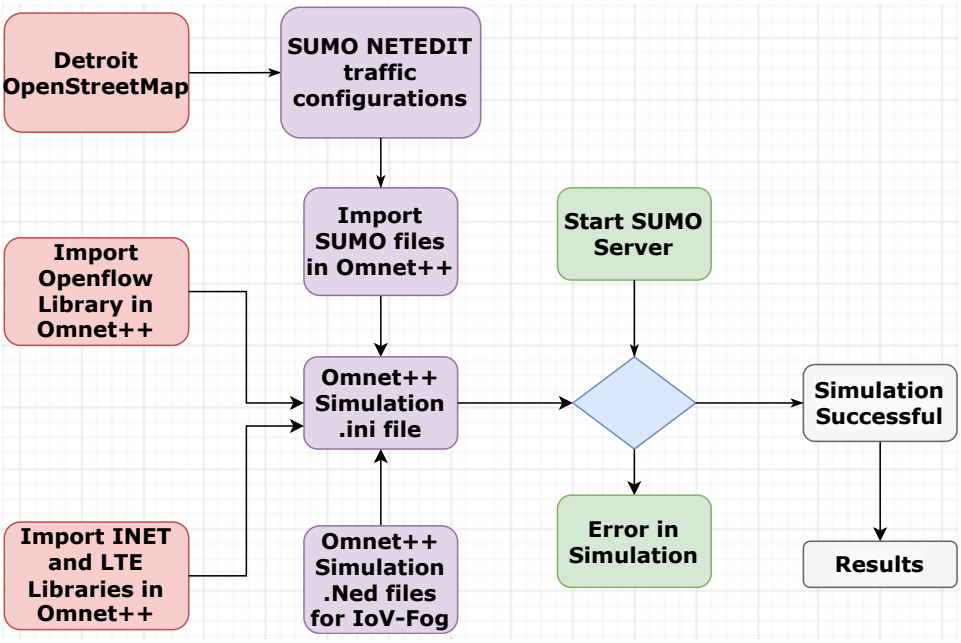

Figure 9: Flowchart showing the IoV-Fog simulation.

4.3 Fast Communication in IoV: Results and Discussion

To evaluate the performance, we have carried out simulations to measure how our method can efficiently allocate the available resources, minimize latency and congestion. We discuss the simulation results as follows.

4.3.1 Fog Nodes Utilization Vs. Number of Tasks

We simulate to measure the utilization of fog resources as shown in Fig. 10. We compare the performance of our method, when it allocates the fog resources to execute the tasks offloaded by the vehicles, with the naive approach. We can see that when the number of tasks from the vehicles increase, the fog nodes utilization in the case of our method was better than allocating the tasks to the nearest fog nodes in the case of the naive approach. The fog resources were less utilized in case of our method because tasks were efficiently allocated to the best available nodes.

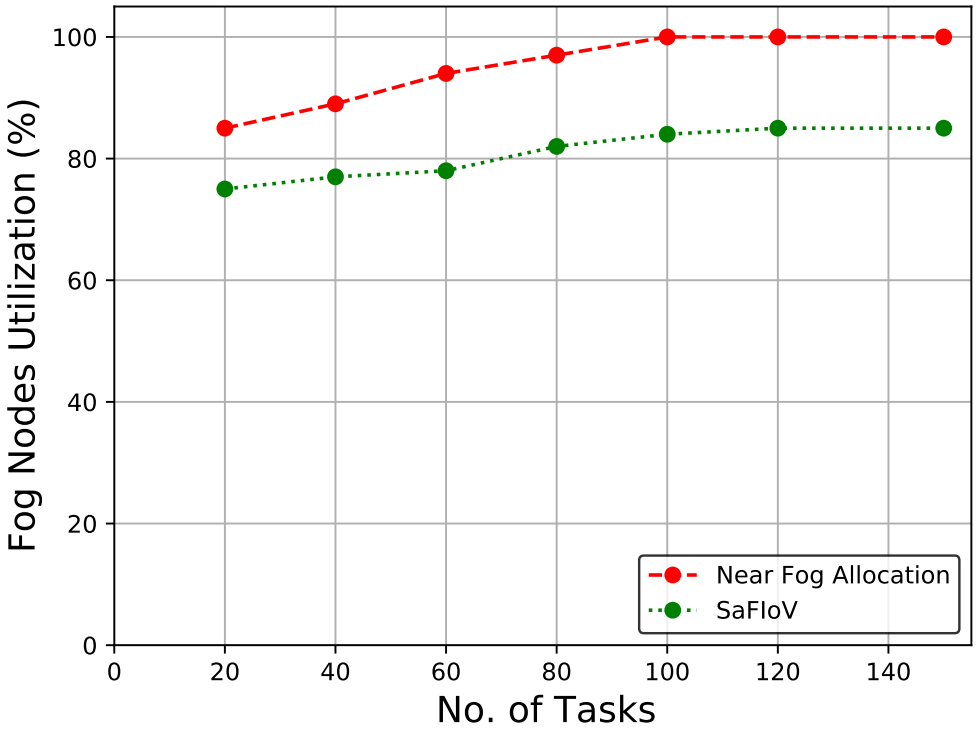

Figure 10: This plot compares results of our method with a naive approach that allocate tasks to the nearest fog nodes. As the number of tasks increase, the fog nodes utilization of both methods are measured.

4.3.2 Latency Vs. Number of Tasks

Our method is capable of handling the tasks and minimizing the end-to-end delays (latency) efficiently. As the number of tasks from the vehicles increases, our method efficiently allocates the tasks to the best possible nodes, considering the distance from the source as well as the load of the nodes. In Figure. 11, we can see a lower latency observed when our method was handling the tasks. The naive approach was sending the task to the nearest fog nodes, overloading the fog nodes, which resulted in high latency.

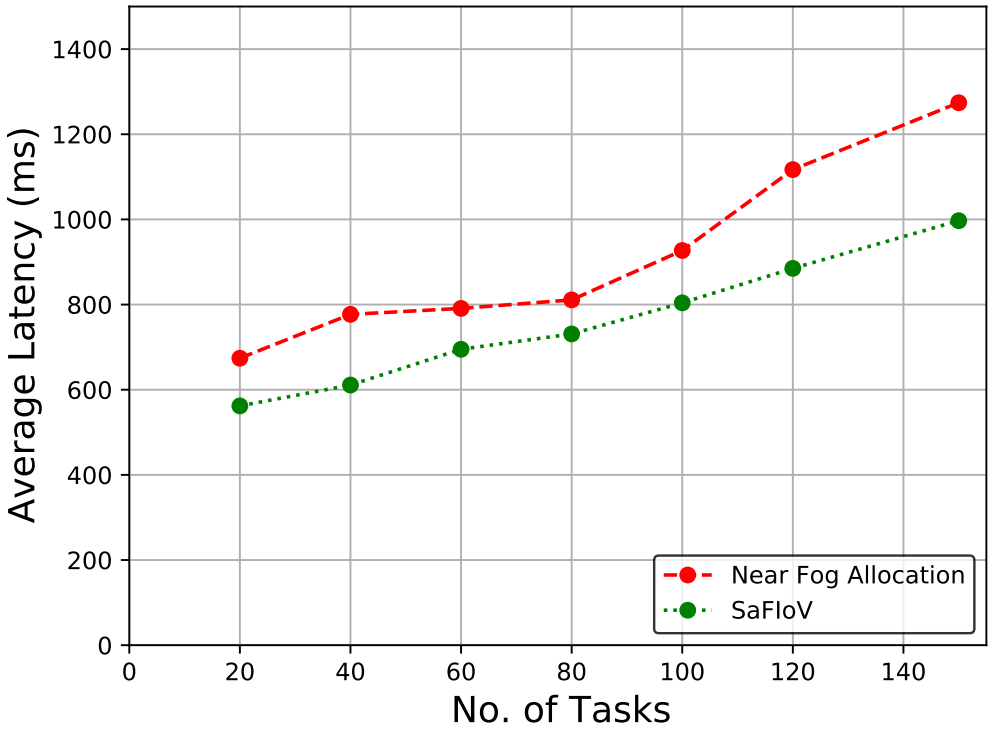

Figure 11: This plot compares results of our method with a naive approach that allocate tasks to the nearest fog nodes. As the number of tasks increase, the average latency of both methods are measured.

4.3.3 Congestion Vs. Number of Tasks

Network congestion in a highly dynamic environment such as IoV can significantly degrade the QoS. Therefore, it is an important parameter to consider for fast communication in the IoV network. As the traffic in the IoV network increases, which is due to an increasing number of tasks, the percentage of congested links also increases. This results in dropping packets or congested streams in the IoV network and, thus, adding more delay. We can see this in Fig. 12. Our method efficiently distributes the tasks among the fog nodes; therefore, the congested links (in percent) are lower than compared to the naive approach that does not use load balancing.

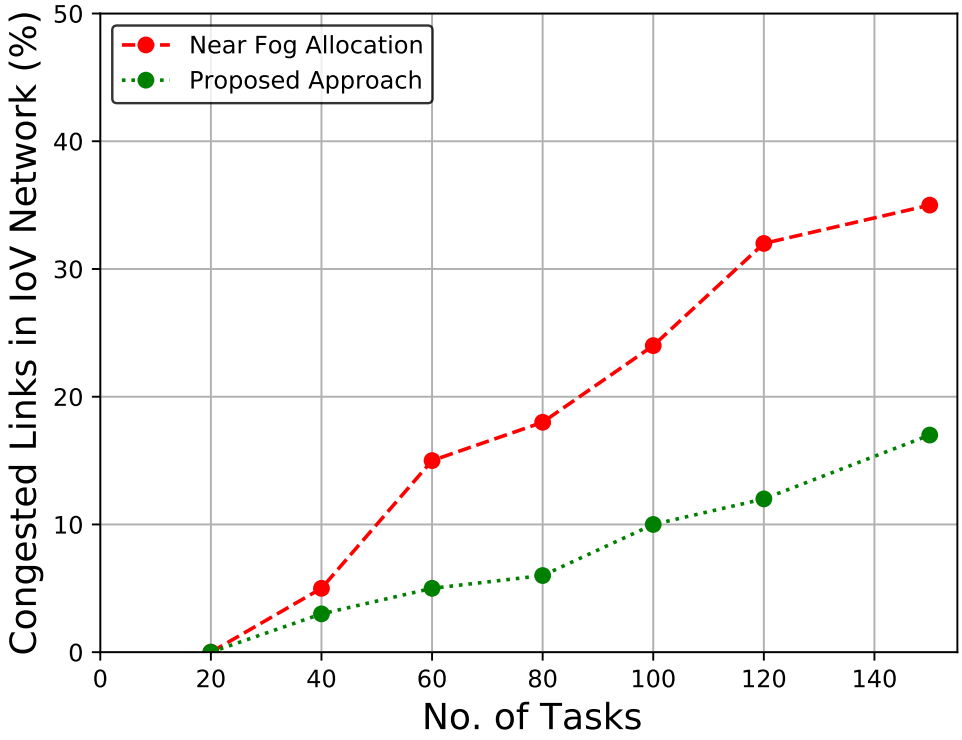

Figure 12: This plot compares results of our method with a naive approach that allocate tasks to the nearest fog nodes. As the number of tasks increase, congestion in the network for both methods are measured.

4.4 Chapter Contributions

In this chapter, we proposed a fast communication framework that improves QoS and minimize end-to-end delays. Also, the performance evaluation of our framework showed that the available resources in fog-based IoV networks are efficiently utilized. Our fast communication framework is based on SDN technology in which the controller employe an RL-based algorithm to effectively assign tasks to fog nodes in order to distribute traffic. Moreover, our experiments concluded that it can avoid congestion in the network and minimizes latency along with utilizing the resources efficiently. The key contributions of this work are stated as follow:

1. Our approach enables data offloading from vehicles to fog nodes, and from fog to fog nodes to achieve adaptive and efficient resource scaling, and also to minimize the end-to-end latency.

2. Our fast communication framework uses machine learning technology such as reinforcement learning to assign best fog resources to an incoming task from a vehicle.

CHAPTER 5 A SIMPLE AND LIGHTWEIGHT AUTHENTICATION SCHEME FOR THE INTERNET OF VEHICLES

Interactions between vehicles and RSUs in a typical IoV architecture are often done through a public channel. As a result, data-in-transit can be intercepted, altered, or erased. In other words, while designing IoV systems, we must consider resistance against attacks such as man-in-the-middle, privilege insider, impersonation, and known key. We must also prevent disclosing or leaking the vehicle's identify (e.g., during sending of information). As a result, authentication is the first and most crucial step in addressing security problems and preventing unwanted access. This work strengthens shared trust among IoV devices by allowing permitted interactions between vehicles and RSUs.

In this chapter, we assess our lightweight and mutual authentication system for IoV devices based on fog computing (i.e., RSUs, vehicles). A decentralized fog-based registration authority (RA) is employed in our method to register RSUs and vehicles. The RSU zone refers to each RSU that serves vehicles travelling within its coverage zone. Furthermore, RSUs and vehicles will select temporary virtual IDs (or pseudonyms) that will vary based on the randomizing strategy. Despite the fact that the creation date and validity period of pseudonyms are picked at random, they comply with PKI standards. Furthermore, the on-board unit (OBU) in vehicles would track when the pseudonyms used in communications needed to be altered. As a result of the randomization technique, eavesdroppers would face more challenges, increasing privacy and anonymity.

5.1 Informal Threat Assessment

In this section, we will go through some of the most typical cybersecurity risks that may be used to attack an IoV network. We'll talk about how our proposed model can handle

such threats.

5.1.1 Man-in-the-Middle Attack

A man-in-the-middle attack is a type of eavesdropping attack in which the attacker intercepts an ongoing conversation or data transmission. After inserting themselves in the *middle* of the conversation, the attackers pretend to be a genuine participant [97, 98]. All information, whether registration or authentication messages, must be kept private. Therefore, an eavesdropper (also known as a man-in-the-middle) would be impossible to identify or track communications between vehicles and RSUs.

Consider an adversary, A, who intercepts the authentication request message $msg_{request}$ and tries to send a genuine request message (i.e., the cryptographic information provided in Section 3.3.4). A may generate a nonce and a timestamp and compute from Eq.21 to Eq.26 for this purpose; however, A will fail to transmit the cryptographic information in the absence of long-term private information such as σ_j and n_j. Similarly, A cannot reproduce specific messages used in authentication. This proves that the system is not vulnerable to a man-in-the-middle attack.

5.1.2 Privilege Insider Attack

Insider privilege attack [99] is a sort of attack in which malicious insiders exploit their privileged account to attack the system. Insider assaults are undeniably the most harmful. A single breach can result in the erasure of databases, the misconfiguration of critical equipment, and the installation of malware on critical systems. Our proposed model's RSUs and vehicles are oblivious of each other's sensitive information, such as secret keys needed to derive cryptographic equations. As a result, instead of communicating their genuine identities, they generate virtual identities (i.e., VId_{R_i} and VId_{v_j}) that are secured

by a one-way hash function and communicated during the registration phase. By doing so, our method withstands this type of attack.

5.1.3 Impersonation Attack

In this type of attack, the adversary *A* attempts to enter the IoV network using fake credentials. It must first intercept the $msg_{request}$ message before it can use the services. The adversary will next attempt to solve the cryptographic equations; however, it will need the secret keys to do so, therefore its request will be rejected when the message's authenticity is verified. As a result, our proposed model also guards against user impersonation attacks.

5.1.4 Known key

Before the session is ended, a new secret key is transmitted to the server and encrypted with the server's public key. A randomly generated nonce is also used in our proposed method, adding an extra degree of protection. This guarantees that each session has a new secret, and hence a new session key.

5.2 Formal Threat Assessment

In this section, we first briefly describe the random oracle model [100], and then discuss the security of our protocol in terms of using private keys using this model. The foundations of the random oracle model are formed by the Real-Or-Random (ROR), which is a widely recognized model-based formal security analysis paradigm.

5.2.1 Random Oracle Model

The Random Oracle Model (ROR) is used to determine whether a security protocol/framework includes the required security feature. The adversary not only monitors all communications exchanged, but also communicates with the participants, according to

this model. A game and an adversary $\mathcal{A}$ are defined using this technique via a probabilistic polynomial-time (PPT) run against the proposed user authentication protocol in time t. There are three network participants: Fog Server, Vehicle, and RSU. The ROR model is useful and may be utilized by participants, even the adversary $\mathcal{A}$. The game starts when queries are accessible $\mathcal{A}$. Thus $\mathcal{A}$ takes a random bit k and asks multiple queries to the participant. The output of these queries should be consistent. After the game is over, $\mathcal{A}$ compares the estimated $k^{'}$ and wins the game when $k^{'} = k$.

Game 0 Toward the beginning of this game, the adversary $\mathcal{A}$ randomly picks the random bit b and then attempts to execute the actual attack. In this game, we have,

$$Adv_{\mathcal{A}}^{Auth}(t) = |2Adv_{\mathcal{A},Game_0}^{Auth} - 1| \tag{5.1}$$

Game 1 In this game, the adversary may listen in on all messages. However, without a private session key, eavesdropping has no effect. As a result, the chance of Game 1 is the same as the likelihood of Game 0.

$$Adv_{\mathcal{A},Game_1}^{Auth} = Adv_{\mathcal{A},Game_0}^{Auth} \tag{5.2}$$

Game 2 $\mathcal{A}$ uses hash and send queries to illustrate an active attack while simultaneously tricking the participant into believing the messages being delivered are real. Then, $\mathcal{A}$ runs Send queries several times until a collision is identified. Because each message is calculated using a secret and random number, no collision occurs.

Game 3 In this game, the adversary listens in on genuine participants' conversations and attempts to gain the actual session key using the info gathered. These session keys are generated with secret keys that the adversary is not aware of. Given the computational difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP), validation becomes difficult. Therefore, we get,

$$Adv^{Auth}_{\mathcal{A},Game_2} - Adv^{Auth}_{\mathcal{A},Game_3} \leq Adv^{ECDLP}_{\mathcal{A}} \tag{5.3}$$

Finally, the adversary $\mathcal{A}$ has to guess the bit b after the execution and simulation of all games and queries.

$$Adv^{Auth}_{\mathcal{A},Game_3} = \frac{1}{2}$$

By using equations 5.1, 5.2 and 5.3, we obtain,

$$\frac{1}{2}Adv^{Auth}_{\mathcal{A}} = |Adv^{Auth}_{\mathcal{A},Game_0} - \frac{1}{2}| = |Adv^{Auth}_{\mathcal{A},Game_1} - \frac{1}{2}| \tag{5.4}$$

5.3 Performance Evaluation

In this section, we assess the effectiveness of our proposed method. To simulate the proposed approach, we used the publicly available Omnet++ simulator with integrated tools such as Veins and Sumo. The simulation was run on an Ubuntu 18.04 LTS operating system with simulation tools installed, and the IEEE 802.11p wireless protocol was employed for communication between vehicles and RSUs. We considered vehicle mobility as well as network traffic congestion in our simulation analysis. Furthermore, the simulation

research included vehicle scalability in the RSU zone.

We established three scenarios to see how scalable our strategy is as the number of vehicles in RSU zones grows. A fog server manages IoV device registration and authentication in each RSU zone (i.e. RSU and vehicle). The network traffic and congestion will rise as the number of vehicles in each RSU zone increases.

1. *Scenario 1:* We have 5 RSUs and 50 vehicles in this scenario, which are grouped into 5 RSU zones. Each RSU zone has 10 vehicles that connect with the RSU and a fog server.

2. *Scenario 2:* In this scenario, we have 5 RSUs and 100 vehicles separated into 10 RSU zones. Each RSU zone has 20 vehicles that connect with the RSU and a fog server.

3. *Scenario 3:* In this scenario, we have 5 RSUs and 150 vehicles, which are divided into 5 RSU zones. Each RSU zone has 30 vehicles that connect with the RSU and a fog server.

Furthermore, the transmission cost of an authentication mechanism should be as minimal as possible. As network traffic grows, the chance of further packets being lost due to congestion grows as well. In each of the three scenarios, this might result in a delay in vehicle registration and authentication, a high rate of network throughput, and a high rate of packet loss. Furthermore, we contrast the use of inherited elliptic curve cryptography (ECC) (160-bit one-way cryptographic hash function) with the standard 1024-bit RSA [101]. ECC offers the same level of security as 1024-bit RSA while keeping communication overheads to a minimum. The results of the simulation study are discussed in the following subsections.

5.3.1 Network Throughput

It is defined as the number of bits sent per unit time and is an important statistic for measuring the efficiency of a protocol. The proposed scheme's network throughput (in bits per second (bps)) for the three scenarios is depicted in Figure 13. As the number of vehicles increased from scenario 1 to scenario 2, and then from scenario 2 to scenario 3, more messages were exchanged, increasing throughput. We can observe that our solution (with inherited ECC hash algorithm) performed better in terms of throughput.

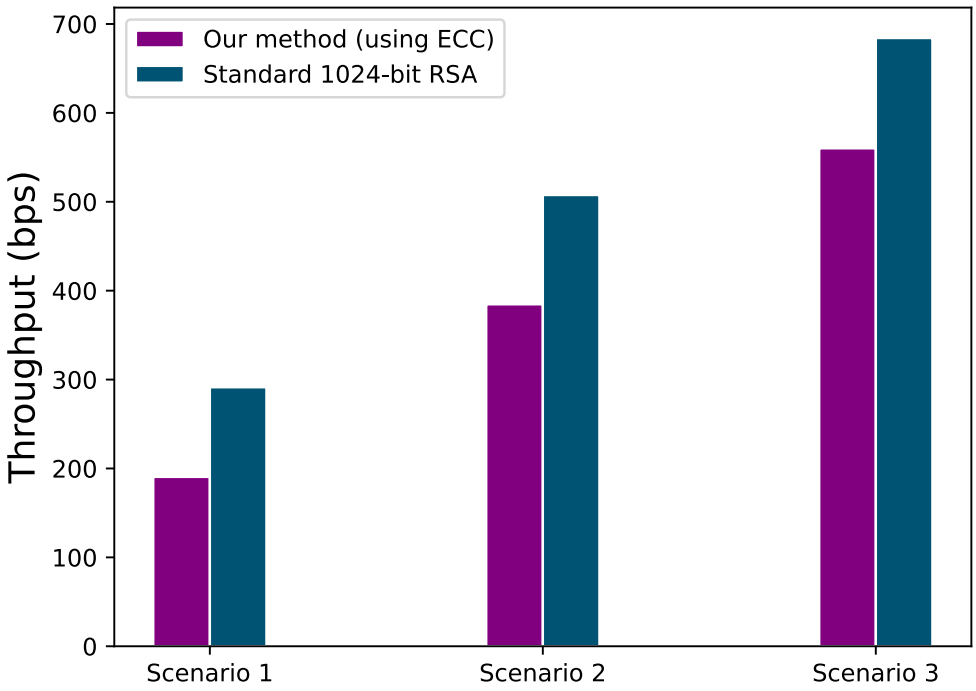

Figure 13: The plots show the simulation results of three different scenarios. The bars compare the results of using ECC-based one-way cryptographic function with standard 1024-bit RSA when implemented in our method. The results are compared while measuring the network throughput of both methods.

5.3.2 End-to-End Delay

It is defined as the average time it takes for communications to reach a specified destination from a given source. The time necessary for authentication followed by the establishment of session keys between communication parties by sending and receiving messages, is the end-to-end delay in an authentication protocol, such as the proposed method. For such key generation and authentication methods, the end-to-end delay should be as short as possible. We showed the end-to-end delay for the proposed system in Fig 14 for the three scenarios. Similar to network throughput, as the number of vehicles increases, so does the value of end-to-end latency since the protocol requires more messages to be sent between RSU/fog and vehicles. Using our proposed scheme performed better with ECC cryptographic hash function.

5.3.3 Rate of Packet Loss

It is also a critical network performance metric, defined as the amount of packet losses per unit time. In a reliable authentication technique, the rate of packet loss should be maintained to a minimal. The packet loss rates for the proposed method are depicted in Figure 15 for the three scenarios. Increased vehicle traffic will almost likely result in higher packet loss rates, as congestion causes more messages to be lost during transmission. Because ECC has less communication cost, the packet loss rate is lower than when employing 1024-bit RSA.

5.3.4 Results Discussion

The proposed PKI-based authentication technique was simulated using three different scenarios, each with 5 RSU zones and 10, 20, and 30 vehicles in the RSU zones, respec-

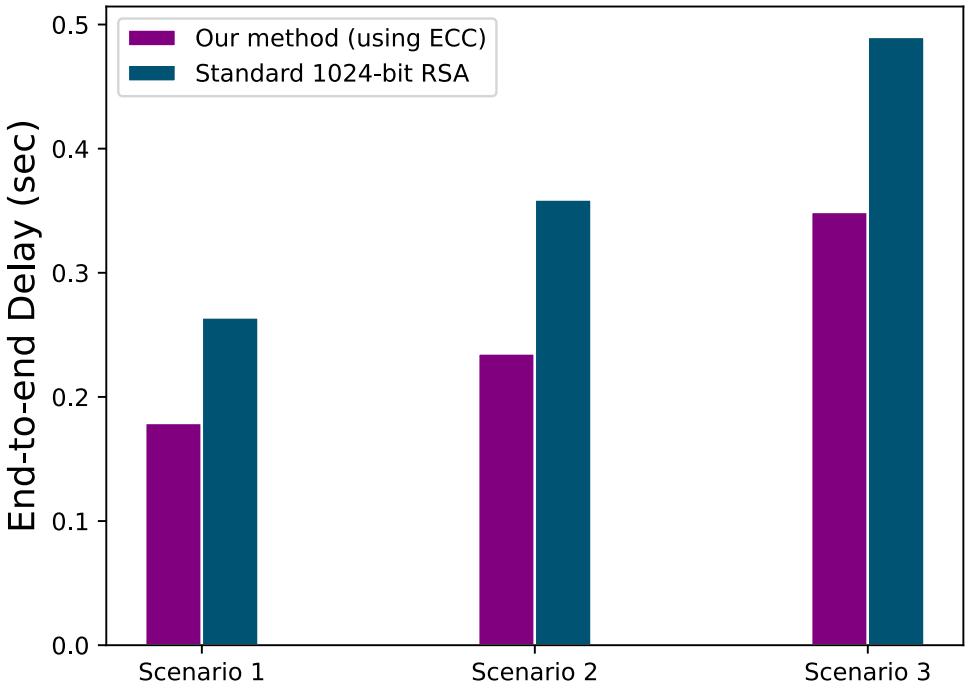

Figure 14: The plots show the simulation results of three different scenarios. The bars compare the results of using ECC-based one-way cryptographic function with standard 1024-bit RSA when implemented in our method. The results are compared while measuring the end-to-end delay of both methods.

tively. The end-to-end delay increased as more messages were sent while simulating with a larger number of vehicles. When the network becomes congested, the rate of packet loss increases as well. Because the communication overhead was kept low, our approach proved effective. For elliptic curve cryptography (ECC) and virtual identification, we employ 160-bits. We also take 128 bits for the random number, nonce, and secret key, as well as 32 bits for the timestamp. The bits in the Γ_1 and Γ_2 groups are 320 and 512, respectively. As a consequence of the reduced amount of message exchanges during the registration and authentication phases, our system would perform better in a large-scale network.

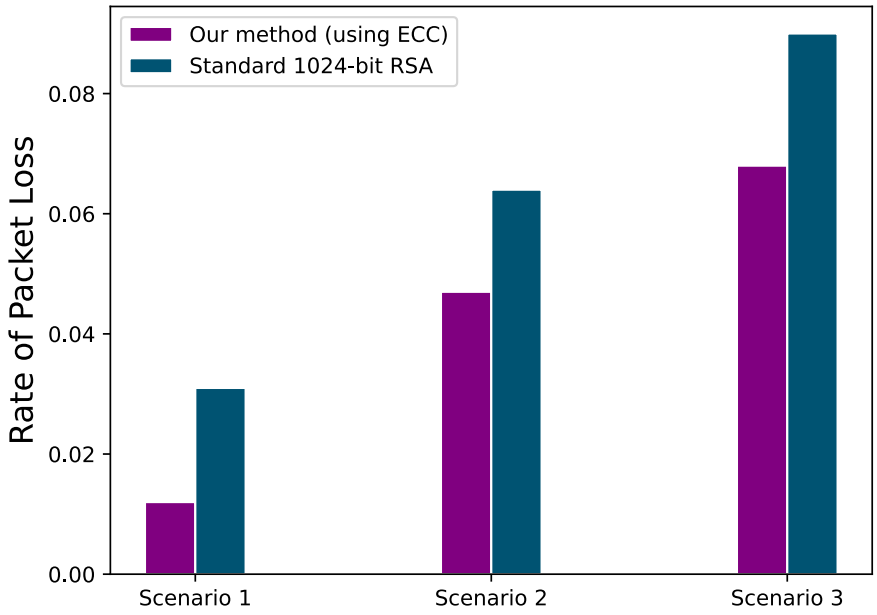

Figure 15: The plots show the simulation results of three different scenarios. The bars compare the results of using ECC-based one-way cryptographic function with standard 1024-bit RSA when implemented in our method. The results are compared while measuring the rate of packet loss of both methods.

5.4 Chapter Contributions

IoV devices work over an insecure channel, and thus, authentication is undoubtedly needed. To address problems during secure communication between IoV devices, in this chapter, we presented a novel lightweight and fog-based authentication key creation technique in the IoV environment. For safe communication between RSU and vehicles in an RSU zone, our proposed authentication scheme involves the creation of a mutual authentication session key. Scalability, authentication, secrecy, and integrity were all recognized as desirable security criteria in the proposed architecture. Moreover, our method uses a pseudonym mechanism to prevent any association with the actual information. Finally, the

network performance of our PKI-based authentication was evaluated using the Omnet++ simulation tool. the main contribution of the chapters were:

1. Our proposed method leverages fog computing in conjunction with IoV to enhance transportation safety while also securing the flow of data generated by vehicles.

2. Our proposed method makes use of lightweight algorithms to offer efficient and secure authentication for IoV devices.

3. The proposed method uses a pseudonym mechanism to prevent any association with the actual information. This mechanism prevents attackers from figuring out the real identities of IoV devices (i.e., RSUs and vehicles).

CHAPTER 6 A PRIVACY PRESERVING-BASED AND TRUST-BASED IOV-FOG ENVIRONMENT

In this chapter, we discuss the privacy preservation technique along with ensuring trust between the cloud and the block sub-system. We employ these techniques in the IoV-Fog environment using Federated Learning (FL) and blockchain technologies. In IoV, the attached OBUs in the vehicles may effectively assist edge or fog nodes in collecting locally sensed data and extracting local knowledge. Using machine-learning techniques on resource-intensive fog nodes, vehicles can learn not only about the road condition but also their surroundings [102]. Therefore, we use FL to train machine learning models on both vehicles and fog nodes. Our approach aims to offload part of the training process to the fog nodes because vehicles in the IoV environment have limited computational power and storage capacity. We'll start with our framework's FL method, which is as follows.

6.1 Federated Learning

In FL, it is not required that the resource-full fog nodes upload their training data to the cloud, but it only trains locally and uploads the updated model parameters cooperatively to enhance the global model. Therefore, FL ensures and protects the privacy of vehicles' private data. Moreover, we presented a lightweight authentication and registration scheme for safe communication between roadside units (RSU) and vehicles in a fog-based IoV environment in Chapter 5. Therefore, private data from the vehicles may be safely transmitted and utilized to train a machine learning model on a fog node leveraging FL. In the situation that the network is unavailable, the models may be trained on the vehicles.

Furthermore, single point of failure and privacy leakage is an issue with traditional centralized systems that rely on a trusted third party [103]. Therefore, we choose honest (i.e.,

reliable) vehicles based on reputation evaluation, which guarantees knowledge reliability. The reputation method takes into account several characteristics, including the degree of honesty, accuracy, and communication punctuality. As a result, not only is network usage reduced by relying solely on honest vehicle data (or model) uploads, but the global trained model in FL also provides significantly more accurate predictions.

Table 2: Notations and their description

Notation	Description
$\|V\| = X$	X number of vehicles in set V
Z	A set of RSU zones
F	A set of Fog nodes
D	Available data to all vehicles
η	Local model learning rate
m	Minors (participants)
γ	All coins divided among the participants
β	Coins
FF	Favorable feedback
AF	Adverse feedback
HI_{FF}	Honesty Impact due to favorable feedback
HI_{AF}	Honesty Impact due to adverse feedback
AI	Accuracy Impact
Rep_{ν}	Reputation of a vehicle ν

6.1.1 Federated Learning in the Perception and Fog Layers

Traditional machine learning models are trained on data that is stored in centralized data centers. In FL, data is distributed over many nodes to train machine learning models. Since IoV data is sensitive to privacy, FL uses a decentralized approach to conduct machine learning in order to protect data privacy while still maintaining the accuracy and performance of the machine learning model. The workflow for employing FL in IoV is that each vehicle $v_i \in V$ uses its own data set $e_k \in D$ to train a local machine learning model

FL_{local_l} called *FL-local-learned model*. The local training on vehicles is only allowed when the network is unavailable or signal strength is insufficient to transmit training data to a fog node $F_j \in F$ on time. When the network is available, vehicles also communicate their trained FL_{local} models to fog nodes through RSU, as shown in Fig. 7. Subsequently, the fog nodes update their local models and upload them to the cloud-based centralized model for aggregating through blockchain. As a result, the centralized cloud trains a *FL-global-learned model*, whereas, in the perception and fog layers, FL prevents a single point of failure. Moreover, the notations we used for modeling are listed in Table 2. The main goal of using FL is to minimize the following objective function, mathematically [104]:

$$\min_{w \in R^d} f\left(w\right),$$

$$s.t. \quad f\left(w\right) := \frac{1}{D} \sum_{k=1}^{X} f_k\left(w\right),$$

where D is the total amount of data, divided by the number of vehicles $|V| = X$, in the IoV system. Using the trained model with parameters w, $f_k(w)$ is the loss function on seen examples, i.e., $l\left(x_i; y_i; w\right)$. We also know that each vehicle uses its own local data set $e_k \in D$ in FL. The objective function may be stated as:

$$\min_{w \in R^d} f\left(w\right) = \frac{1}{D} \sum_{i=1}^{X} \sum_{j \in e_k} f_j(w) \tag{6.1}$$

Furthermore, we assume that the IoV data available for each vehicle is often non-

representative of the whole distribution because it only reflects the vehicle observation. Furthermore, since FL is a decentralized architecture, the Federated Aggregation algorithm is used to optimize models in vehicles and the cloud. Each vehicle performs a stochastic gradient descent step (i.e., at time t) on the local data available. Mathematically, each vehicle updates the model parameters as:

$$w_{t+1}^{j} \leftarrow w_t - \eta \psi_j,$$

$$s.t. \quad \psi_j = \nabla f_j(w_t),$$

where η is the model learning rate (hyperparameter). Therefore, each vehicle repeat the process of model updates for κ times. The server combines all gradients and adjusts the global model per round ρ as follows:

$$w_{\rho+1} \leftarrow w_\rho - \eta \frac{1}{D} \sum_{i=1}^{X} \sum_{j \in e_k} \nabla f_j(w) \tag{6.2}$$

We can see that the performance of the global model is improved by factors such as κ iterations of applying gradient descent for each vehicle $v \in V$ for *FL-local-learned model*, and ρ rounds to update *FL-global-learned model*.

6.2 The Reputation Scheme

The fog nodes receive data or model parameters from the vehicles to update their *FL-local-learned model*. Fog nodes serve RSU zones, and they are often hesitant of receiving model updates (or training data) from vehicles that are known for spreading misleading data. For example, the authors in [105] use a poison attack to evaluate the quality of a local model that may normally be aggregated to the global model. When a vehicle shares *favorable feedback* (FF) with a fog node, *FL-global-learned model's* accuracy ultimately improves, while the vehicle's reputation improves at the same time. *adverse feedback* (AF) on the other hand, will result in a loss of reputation. A vehicle with a bad reputation is not allowed to participate in further training of the *FL-local-learned model*. The reputation model in this study takes into account the interaction of vehicles with fog nodes. We'll presume that ν and f refer to a vehicle and a fog node, respectively. A fog node f determines reputation based on the following characteristics while communicating with a vehicle ν.

6.2.1 Honesty Impact (HI)

To achieve a high HI between vehicles and fog nodes, there must be more interactions and hence more FF. Moreover, HI is divided into two categories, which is based on the frequency of interactions (i.e., FF and AF) to share model parameters or training data, and is computed as:

$$HI_{FF_{\nu \leftrightarrow f}} = \frac{T_{FF_{\nu \leftrightarrow f}}}{T_{\nu \leftrightarrow f}}, \tag{6.3}$$

$$HI_{AF_{\nu \leftrightarrow f}} = \frac{T_{AF_{\nu \leftrightarrow f}}}{T_{\nu \leftrightarrow f}}, \tag{6.4}$$

where $T_{\nu \leftrightarrow f}$ represents the total number of interactions between a vehicle ν and a fog node f. In addition, T_{FF} and T_{AF} are the favorable and adverse interactions, respectively.

6.2.2 Accuracy Impact (AI)

Another element that influences reputation is the extent to which model accuracy improves. Let's say the global model loss before and after the k^{th} interaction between a vehicle ν and a fog node f is represented as $L_{k-1_{\nu \leftrightarrow f}}$ and $L_{k_{\nu \leftrightarrow f}}$, respectively. Next, the AI can be calculated as:

$$AI_{k_{\nu \leftrightarrow f}} = log_2 \left[1 + \frac{-\left(L_{k_{\nu \leftrightarrow f}} - L_{k-1_{\nu \leftrightarrow f}}\right)}{L_{k-1_{\nu \leftrightarrow f}}} \right] \tag{6.5}$$

The updated model is beneficial when the global model loss decreases after the k^th interaction, i.e., $L_{k_{\nu \leftrightarrow f}} < L_{k-1_{\nu \leftrightarrow f}}$. As a result, $AI_{k_{\nu \leftrightarrow f}} > 0$, with the value of $AI_{k_{\nu \leftrightarrow f}}$ increases as the loss decreases. If the loss increases, however, then $AI_{k_{\nu \leftrightarrow f}} < 0$. Furthermore, $AI_{k_{\nu \leftrightarrow f}} = 0$ when the uploaded parameters have no influence on the global model. Therefore, we can categorize the contribution as favorable or adverse. Mathematically,

$$AI_{FF_{k_{\nu \leftrightarrow f}}} = log_2 \left[1 + \frac{-\left(L_{FF_{k_{\nu \leftrightarrow f}}} - L_{FF_{k-1_{\nu \leftrightarrow f}}}\right)}{L_{FF_{k-1_{\nu \leftrightarrow f}}}} \right], \tag{6.6}$$

$$AI_{AF_{k_{\nu \leftrightarrow f}}} = log_2 \left[1 + \frac{-\left(L_{AF_{k_{\nu \leftrightarrow f}}} - L_{AF_{k-1_{\nu \leftrightarrow f}}}\right)}{L_{AF_{k-1_{\nu \leftrightarrow f}}}} \right], \tag{6.7}$$

where $AI_{FF_{k_{\nu \leftrightarrow f}}} > 0$ and $AI_{AF_{k_{\nu \leftrightarrow f}}} < 0$.

6.2.3 Reputation of a Vehicle

The reputation of a vehicle is established by its interaction with a fog node in terms of the degree of honesty and accuracy level, taking into account the aforementioned factors (HI and AI). Mathematically, the reputation of a vehicle ν is:

$$Rep_{\nu} = HI_{FF_{\nu \leftrightarrow f}} AI_{FF_{k_{\nu \leftrightarrow f}}} + HI_{AF_{\nu \leftrightarrow f}} AI_{AF_{k_{\nu \leftrightarrow f}}} \tag{6.8}$$

In E.q 6.8, we can see that two characteristics (AI and HI) are combined to determine a vehicle's reputation. Furthermore, to calculate the reputation values of all vehicles in the IoV network, we consider the maximum reputation as, $\underset{\text{max}}{Rep_{\nu \leftrightarrow f}}$, such that $\nu \in [1, V], f \in [1, F]$. Also, we define a reputation threshold as, $\epsilon \cdot \underset{\text{max}}{Rep_{\nu \leftrightarrow f}}$, such that $\epsilon \in [0, 1)$. The reputation vehicles that are above the present threshold are trusted to take part in the training of *FL-local-learned model*. The value of ϵ can be dynamically adjusted to meet model training requirements. The higher the value of ϵ, the higher the requirement for model accuracy, and the more rounds are required to make the global model's loss fulfill specified criteria.

6.3 Performance Evaluation

This section describes the performance evaluation of our approach, which is based on FL, blockchain and reputation scheme. We start with evaluating the reputation-based scheme (described in Section 6.2) in terms of ML performance using FL, and then we examine the security and efficiency of the blockchain.

6.3.1 Experimental Setup

We utilized the *MNIST* dataset to test the reputation scheme, which has $60,000$ training and $10,000$ testing images of handwritten digits [106]. The dataset is divided into 100 Tranches, each of which is assigned to 100 vehicles. The vehicles configured to deliver the training data to the fog nodes when the signals were good. They were also opt to train the *FL-Local-Learned model* themselves if the network signal strength is poor. For the ML, a CNN model was used as the *FL-Local-Learned model*. The blockchain consortium's leader adds local models to the chain, which the cloud collects and analyzes to create a *FL-Global-Learned model*. The performance of the global model on the test set is then evaluated by the cloud, as shown in Fig. 16.

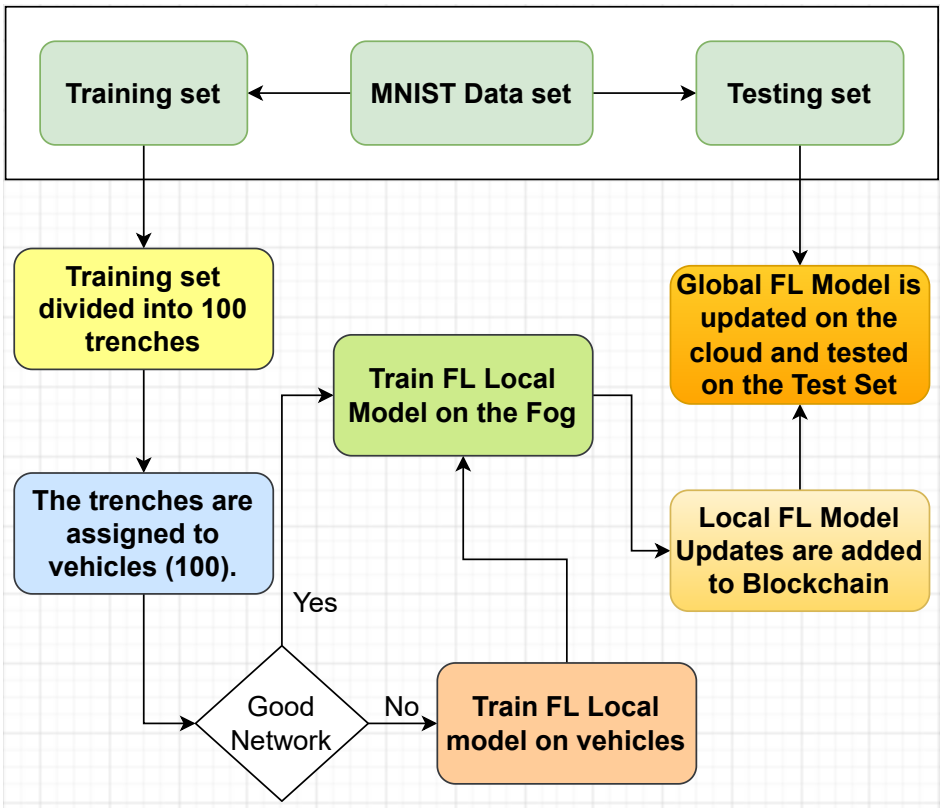

Figure 16: Flowchart showing the simulation steps using Federated Learning and Blockchain.

6.3.2 The Effectiveness of the Reputation Scheme Under FL

We evaluate the accuracy of the global model, with respect to 50 FL rounds to update the model parameters, with and without the reputation scheme. The accuracy of the global model, under a fixed threshold ($\epsilon = 0.45$), is used to compare the performance of our approach with and without the reputation scheme. This threshold is the optimal number determined after a series of experiments to obtain a compromise between the accuracy and convergence speed. Also, by setting $\epsilon = 0$, the FL process continues without taking the reputation into account. The model incorporating reputation evaluation obtained greater accuracy, as shown in Figure 17. When there are more reputable vehicles participating, high accuracy is obtained. This is owing to the fact that when there are more reputable vehicles participating, the model is thoroughly learned. However, when there were 30 vehicles, the model took more time to convergence, which was keeping the model accuracy rate lower.

6.3.3 The Effectiveness of Rewarding Vehicles

The reputation of a vehicle is directly affected by how they are rewarded. In the experiments, we take into account the following parameters. We suppose that every vehicle's initial reward value is 0.45, which is the "ϵ" threshold value. Equation 6.8 is used to determine the initial reward value, in which the two factors (AI and HI) are given equal weights and calculated as 1. Furthermore, each global epoch updates the initial reward value of the vehicles. In this experiment, we compare the results of vehicles achieving rewards in four distinct scenarios as follows:

1. Vehicle without using the reputation scheme.

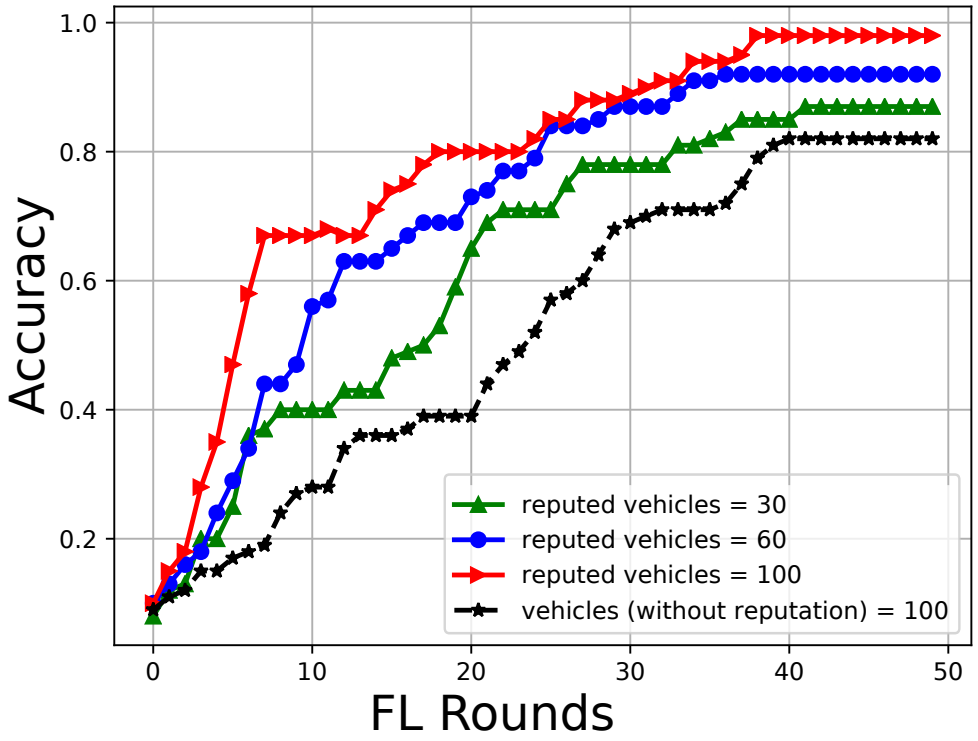

Figure 17: The global model's accuracy with varied numbers of vehicles under reputation and non-reputation schemes.

2. Vehicle always using the reputation scheme with no malicious activity.

3. A vehicle using reputation scheme, but performing a poisoning attack at epoch 1.

4. A vehicle using reputation scheme, but performing a poisoning attack at epoch 0 and 3.

The vehicles that do not use the reputation scheme will always have the initial reward of 0.45 in each global epoch, as shown in Fig. 18. We can observe that if a vehicle is not rewarded, its value remains constant. When the reputation scheme is used, however, the reward value of honest vehicles whose model updates are approved increases as the number of global epochs increases. If a vehicle's update is deemed a threat (i.e., poisoning attack), it will not be aggregated to the global model and will be assigned a reward value

equal to 0. This will take some time for the vehicle to build its reputation, which is getting more awards (depending on its honesty), in order for its updates to be accepted for the global model.

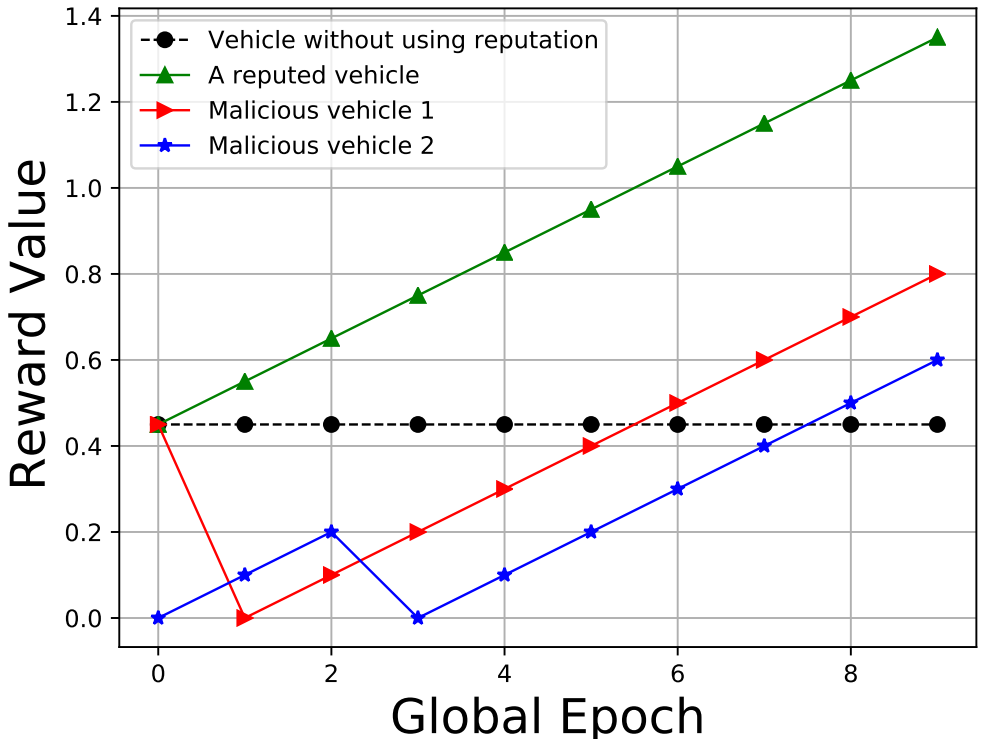

Figure 18: Rewarding reputed and malicious vehicles.

6.3.4 The Effectiveness of Blockchain

We used blockchain to maintain trust between fog nodes, the cloud, and the vehicles as they interact and exchange model updates. To evaluate the efficiency of blockchain, we created a scenario that compares the performance of the blockchain we constructed with transactions that do not use blockchain.

Table 3 shows the simulation results for packet overhead. Blockchain's encryption and hashing impose a small packet overhead. However, our implementation of blockchain, in

which miners compete for the leader for all types of transactions, has lowered total packet overhead and ensured trust.

In addition, the time overhead on the blockchain is calculated from the time a transaction is received by the miner/leader until the requester receives a response. We can see in Table 3 that the time it takes to conduct a transaction using our approach is comparable to without utilizing blockchain. As a result, employing approach to guarantee trust between fog nodes and the cloud can be fast with more efficient than the system without using blockchain.

Table 3: Packet overhead with or without using blockchain

Flow of Packet	Packet overhead without BC (Bytes)	Transaction Duration without BC (sec)	Packet overhead with BC (Bytes)	Transaction Duration with BC (sec)	Packet overhead with our method (Bytes)	Transaction Duration with our method (sec)
From a fog node to blockchain	8	3	18	5	12	3
From blockchain to the cloud	8	2	42	6	29	2
From the cloud to the blockchain	8	3	18	5	12	3

6.4 Chapter Contributions

In this Chapter, we presented FL and blockchain-based method for IoV environment using fog computing and reputation scheme. Our method adapt to the large-scale vehicular networks while providing a secure and reliable sharing of knowledge. In the perception layer of the IoV, vehicles data is securely shared to the fog nodes for local model training in

case of good network connection. When the network is unavailable, local models are used to train on vehicles. A reputation scheme ensures that only honest vehicles are allowed to share the training data, therefore, minimizing the risk of data set poisoning attack. The blockchain layer is responsible to ensure the cloud and vehicles trust each other when the updates are shared with the cloud and the global learned model is used by vehicles for inference. Our experimental study shows the effectiveness of the FL and reputation methods. Furthermore, experiments shows that the time it takes to perform a transaction using blockchain in our method is fast enough for IoV environment. As a result, the blockchain is efficient in terms of both processing time and ensuring trust. The key contributions of this chapter were:

1. Given the high dynamic environment and local data available to vehicles in IoV, our method explores the learning of global model using FL in conjunction with fog computing. The benefits of both fog computing and FL are utilized to overcome the latency and learning of global model.

2. Our method is based on a reputation scheme that rewards honest vehicles, which improves the accuracy of the global learned model.

3. Our method utilizes blockchain technology to ensure trust when local learned model updates are shared with the cloud.

CHAPTER 7 CONCLUSION

The Intelligent Transportation System (ITS) is a decentralized system that connects several services to create unified solutions, such as sophisticated traffic control systems. The advantages of ITS over traditional centralized systems are safety, availability, and dependability. The Internet-of-Things (IoT), on the other hand, helps to improve resource efficiency in computing systems by connecting to virtual resources in the cloud. The integration of ITS with IoT results in the Internet-of-Vehicles (IoV), which enables real-time data exchange to create a safe and reliable transportation system. In addition, the benefits of IoV include decreased costs owing to the use of resource-efficient cloud computing, as well as greater safety due to accurate and timely traffic reports. However, with the IoV, relying too much on cloud computing (i.e., real-time applications) can lead to issues like poor quality-of-service (QoS) and extended end-to-end delays (also known as latency). As a result, vehicular fog computing (VFC), which merges the IoV perception layer with a fog computing layer, can be utilized to reduce latency.

Fog computing provides compute and storage services at the network's edge. It connects the capabilities of cloud computing with those of IoT devices. This thesis proposed a framework that allows information processing, machine learning, and registration of vehicles in the fog nodes. However, the resources available in the fog computing layer are limited, which requires an effective strategy to allocate them to tasks originating from the vehicles. In other words. a load balancer was employed in our framework, which uses underlying technology such as SDN. We used an RL-based algorithm at the SDN controller to efficiently allocate tasks to fog nodes in order to distribute traffic. While fog nodes can

provide real-time computing and networking services to IoV vehicles, SDN-based switches that are connected to the SDN controller are responsible for delivering real-time network status reports to the controller. The controller keeps track of the network topology from a logical and high-level perspective. This is necessary for load balancing to be carried out in order to minimize latency and make optimal use of the fog nodes' resources. In Section 4.2, we carried out extensive experiments that demonstrated that our framework is capable of avoiding network congestion, minimizing latency, and efficiently utilizing resources.

Furthermore, in this thesis, we demonstrated how fog computing combined with IoV improves traffic safety while also securing the flow of data created by vehicles. We introduced a lightweight method for IoV device authentication that is both efficient and secure. Authenticating the IoV devices is required as they work over an insecure channel. The verified and authorized device may then connect with other trustworthy entities in its proximity via secure communication channels, ensuring that even if a hostile entity infiltrates the channel, they will not be able to decode and steal data from other devices in the system. Therefore, we introduces a reliable, cost-effective, and distributed authentication system for IoV networks using fog computing. The cloud, fog servers, and RSU zone are the three main components in the proposed system architecture. Vehicles are end-devices that are equipped with a number of sensors that create data and send it to RSUs. The three main components are used to register IoV devices securely and mutually authenticate them before they may communicate. Furthermore, we showed that our proposed authentication scheme can withstand threats such as Man-in-the-middle Attack, Privilege Insider Attack, Impersonation Attack, and Known key attack. Also, we carried-out and discussed formal threat assessment of our authentication scheme using Random Oracle Model (ROR). Our

simulation results concluded that in a large scale network, our scheme would show better results, due to the reduced number of message exchanges during the registration and authentication phases.

In addition, this thesis proposed a method for training machine learning models locally on vehicles and fog servers using federated learning (FL). The intuition behind using FL is to preserve the privacy of critical vehicle data such as location, driver's identity, and so on. The FL approach for model training, on the other hand, is sensitive to model poisoning attacks and the risks of data leakage on a fog computing server hosted by a third party. We employed blockchain to address this, which has gained widespread use due to its secure, anonymous, and decentralized trust features. Furthermore, we employ a reputation-based technique to further ensure the credibility of information acquired as a result of model training. The reputation approach considers a number of factors, including degree of honesty, accuracy, and communication timeliness. As a result of relying entirely on honest vehicle data (or model) uploads, not only is network use minimized, but the global trained model in FL also gives substantially more accurate predictions. Nonetheless, the overall compute cost of completing the blockchain consensus process is reduced. Our fog computing and reputation scheme-based FL and blockchain-based approach for IoV environments adapts to large-scale vehicular networks while offering safe and trustworthy knowledge sharing. Vehicle data is safely exchanged with fog nodes for local model training in the IoV's perception layer. Our experimental study shows the effectiveness of the FL and reputation methods.

REFERENCES

[1] Javier Barrachina, Piedad Garrido, Manuel Fogue, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Caova: A car accident ontology for vanets. In *2012 IEEE wireless communications and networking conference (WCNC)*, pages 1864–1869. Ieee, 2012.

[2] Xiangming Wen, Jenhui Chen, Zhiqun Hu, and Zhaoming Lu. A p-opportunistic channel access scheme for interference mitigation between v2v and v2i communications. *IEEE Internet of Things Journal*, 7(5):3706–3718, 2020.

[3] Sebastian Kühlmorgen, Hongsheng Lu, Andreas Festag, John Kenney, Sebastian Gemsheim, and Gerhard Fettweis. Evaluation of congestion-enabled forwarding with mixed data traffic in vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):233–247, 2019.

[4] Dongyao Jia, Kejie Lu, and Jianping Wang. A disturbance-adaptive design for vanet-enabled vehicle platoon. *IEEE Transactions on Vehicular Technology*, 63(2):527–539, 2014.

[5] Marcelo Yannuzzi, Rodolfo Milito, René Serral-Gracià, Diego Montero, and Mario Nemirovsky. Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing. In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 325–329. IEEE, 2014.

[6] Omprakash Kaiwartya, Abdul Hanan Abdullah, Yue Cao, Ayman Altameem, Mukesh Prasad, Chin-Teng Lin, and Xiulei Liu. Internet of vehicles: Motivation, layered

architecture, network model, challenges, and future aspects. *IEEE Access*, 4:5356–5373, 2016.

[7] Mario Gerla, Eun-Kyu Lee, Giovanni Pau, and Uichin Lee. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In *2014 IEEE world forum on internet of things (WF-IoT)*, pages 241–246. IEEE, 2014.

[8] Salim Bitam and Abdelhamid Mellouk. Its-cloud: Cloud computing for intelligent transportation system. In *2012 IEEE global communications conference (GLOBECOM)*, pages 2054–2059. IEEE, 2012.

[9] Xiaojun Yang, Luan Zeng, Feng Luo, and Shaoxuan Wang. Cloud hierarchical analysis. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, 7(12):2468–2477, 2010.

[10] Chao Zhu, Giancarlo Pastor, Yu Xiao, Yong Li, and Antti Ylae-Jaeaeski. Fog following me: Latency and quality balanced task allocation in vehicular fog computing. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2018.

[11] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16, 2012.

[12] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C. Lin, and X. Liu. Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. *IEEE Access*, 4:5356–5373, 2016.

[13] Ahmed Jawad Kadhim and Seyed Amin Hosseini Seno. Maximizing the utilization of fog computing in internet of vehicle using sdn. *IEEE Communications Letters*,

23(1):140–143, 2018.

[14] Chengchao Liang, F Richard Yu, and Xi Zhang. Information-centric network function virtualization over 5g mobile wireless networks. *IEEE network*, 29(3):68–74, 2015.

[15] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[16] Maria Azees, Pandi Vijayakumar, and Lazarus Jegatha Deborah. Comprehensive survey on security services in vehicular ad-hoc networks. *IET Intelligent Transport Systems*, 10(6):379–388, 2016.

[17] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials*, 17(1):228–255, 2014.

[18] David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[19] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

[20] Marc Pilkington. Blockchain technology: principles and applications. In *Research handbook on digital transformations*. Edward Elgar Publishing, 2016.

[21] Tigang Jiang, Hua Fang, and Honggang Wang. Blockchain-based internet of vehicles: Distributed network architecture and performance analysis. *IEEE Internet of Things Journal*, 6(3):4640–4649, 2018.

[22] Oscar Novo. Blockchain meets iot: An architecture for scalable access management in iot. *IEEE Internet of Things Journal*, 5(2):1184–1195, 2018.

[23] VP Anuradha and D Sumathi. A survey on resource allocation strategies in cloud computing. In *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pages 1–7. IEEE, 2014.

[24] Sudip Misra and Niloy Saha. Detour: Dynamic task offloading in software-defined fog for iot applications. *IEEE Journal on Selected Areas in Communications*, 37(5):1159–1166, 2019.

[25] Martin Casado, Michael J Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. Ethane: Taking control of the enterprise. *ACM SIGCOMM computer communication review*, 37(4):1–12, 2007.

[26] Hamid Farhady, HyunYong Lee, and Akihiro Nakao. Software-defined networking: A survey. *Computer Networks*, 81:79–95, 2015.

[27] Myung-Ki Shin, Ki-Hyuk Nam, and Hyoung-Jun Kim. Software-defined networking (sdn): A reference architecture and open apis. In *2012 International Conference on ICT Convergence (ICTC)*, pages 360–361. IEEE, 2012.

[28] Ian F Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, and Wu Chou. A roadmap for traffic engineering in sdn-openflow networks. *Computer Networks*, 71:1–30, 2014.

[29] Abdelhamied A Ateya, Anastasia Vybornova, Ruslan Kirichek, and Andrey Koucheryavy. Multilevel cloud based tactile internet system. In *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pages 105–110. IEEE, 2017.

[30] Wenyu Zhang, Zhenjiang Zhang, and Han-Chieh Chao. 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, 2017.

[31] Iman Azimi, Arman Anzanpour, Amir M Rahmani, Tapio Pahikkala, Marco Levorato, Pasi Liljeberg, and Nikil Dutt. Hich: Hierarchical fog-assisted computing architecture for healthcare iot. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):1–20, 2017.

[32] Cunqian Yu, Bin Lin, Ping Guo, Wei Zhang, Sen Li, and Rongxi He. Deployment and dimensioning of fog computing-based internet of vehicle infrastructure for autonomous driving. *IEEE Internet of Things Journal*, 6(1):149–160, 2018.

[33] Fang Fu, Yunpeng Kang, Zhicai Zhang, F Richard Yu, and Tuan Wu. Soft actor-critic drl for live transcoding and streaming in vehicular fog computing-enabled iov. *IEEE Internet of Things Journal*, 2020.

[34] Bin Cao, Zhiheng Sun, Jintong Zhang, and Yu Gu. Resource allocation in 5g iov architecture based on sdn and fog-cloud computing. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[35] Jéferson Campos Nobre, Allan M de Souza, Denis Rosário, Cristiano Both, Leandro A Villas, Eduardo Cerqueira, Torsten Braun, and Mario Gerla. Vehicular software-defined networking and fog computing: Integration and design principles. *Ad Hoc Networks*, 82:172–181, 2019.

[36] Zongjian He, Jiannong Cao, and Xuefeng Liu. Sdvn: Enabling rapid network innovation for heterogeneous vehicular communication. *IEEE network*, 30(4):10–15, 2016.

[37] Muhammad Arif, Guojun Wang, Tian Wang, and Tao Peng. Sdn-based secure vanets communication with fog computing. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 46–59. Springer, 2018.

[38] Ahmed Jawad Kadhim and Seyed Amin Hosseini Seno. Energy-efficient multicast routing protocol based on sdn and fog computing for vehicular networks. *Ad Hoc Networks*, 84:68–81, 2019.

[39] Mostafa Ghobaei-Arani, Alireza Souri, and Ali A Rahmanian. Resource management approaches in fog computing: a comprehensive review. *Journal of Grid Computing*, pages 1–42, 2019.

[40] Antonio Brogi and Stefano Forti. Qos-aware deployment of iot applications through the fog. *IEEE Internet of Things Journal*, 4(5):1185–1192, 2017.

[41] Minh-Quang Tran, Duy Tai Nguyen, Van An Le, Duc Hai Nguyen, and Tran Vu Pham. Task placement on fog computing made efficient for iot application provision. *Wireless Communications and Mobile Computing*, 2019, 2019.

[42] Haiying Shen and Liuhua Chen. A resource usage intensity aware load balancing method for virtual machine migration in cloud datacenters. *IEEE Transactions on Cloud Computing*, 8(1):17–31, 2017.

[43] Aliyu Lawal Aliyu, Peter Bull, and Ali Abdallah. A trust management framework for network applications within an sdn environment. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 93–98. IEEE, 2017.

[44] Bassey Isong, Tebogo Kgogo, Francis Lugayizi, and Bennett Kankuzi. Trust establishment framework between sdn controller and applications. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 101–107. IEEE, 2017.

[45] Sandra Scott-Hayward, Christopher Kane, and Sakir Sezer. Operationcheckpoint: Sdn application control. In *2014 IEEE 22nd International Conference on Network Protocols*, pages 618–623. IEEE, 2014.

[46] Muhammad Naveed Aman, Mohamed Haroon Basheer, and Biplab Sikdar. Two-factor authentication for iot with location information. *IEEE Internet of Things Journal*, 6(2):3335–3351, 2018.

[47] Parwinder Kaur Dhillon and Sheetal Kalra. Multi-factor user authentication scheme for iot-based healthcare services. *Journal of Reliable Intelligent Environments*, 4(3):141–160, 2018.

[48] Preeti Chandrakar and Hari Om. An efficient two-factor remote user authentication and session key agreement scheme using rabin cryptosystem. *Arabian Journal for Science and Engineering*, 43(2):661–673, 2018.

[49] Qi Jiang, Xin Zhang, Ning Zhang, Youliang Tian, Xindi Ma, and Jianfeng Ma. Two-factor authentication protocol using physical unclonable function for iov. In *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 195–200. IEEE, 2019.

[50] Qi Jiang, Muhammad Khurram Khan, Xiang Lu, Jianfeng Ma, and Debiao He. A privacy preserving three-factor authentication protocol for e-health clouds. *The Journal of Supercomputing*, 72(10):3826–3849, 2016.

[51] JoonYoung Lee, SungJin Yu, KiSung Park, YoHan Park, and YoungHo Park. Secure three-factor authentication protocol for multi-gateway iot environments. *Sensors*, 19(10):2358, 2019.

[52] Liang Kou, Yiqi Shi, Liguo Zhang, Duo Liu, and Qing Yang. A lightweight three-factor user authentication protocol for the information perception of iot. *CMC-Computers, Materials & Continua*, 58(2):545–565, 2019.

[53] Qi Jiang, Xin Zhang, Ning Zhang, Youliang Tian, Xindi Ma, and Jianfeng Ma. Three-factor authentication protocol using physical unclonable function for iov. *Computer Communications*, 173:45–55, 2021.

[54] Ashok Kumar Das, Anil Kumar Sutrala, Vanga Odelu, and Adrijit Goswami. A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks. *Wireless Personal Communications*, 94(3):1899–1933, 2017.

[55] Fan Wu, Lili Xu, Saru Kumari, and Xiong Li. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-to-Peer Networking and Applications*, 10(1):16–30, 2017.

[56] Marvin A Sirbu and JC-I Chuang. Distributed authentication in kerberos using public key cryptography. In *Proceedings of SNDSS'97: Internet Society 1997 Symposium on Network and Distributed System Security*, pages 134–141. IEEE, 1997.

[57] Eric Rescorla. Rfc2631: Diffie-hellman key agreement method, 1999.

[58] Hodjat Hamidi. An approach to develop the smart health using internet of things and authentication based on biometric technology. *Future generation computer systems*, 91:434–449, 2019.

[59] Zeeshan Ali, Anwar Ghani, Imran Khan, Shehzad Ashraf Chaudhry, SK Hafizul Islam, and Debasis Giri. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *Journal of Information Security and Applications*, 52:102502, 2020.

[60] Aamir Akbar and Peter R Lewis. Self-adaptive and self-aware mobile-cloud hybrid robotics. In *2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pages 262–267. IEEE, 2018.

[61] Aamir Akbar and Peter R Lewis. The importance of granularity in multiobjective optimization of mobile cloud hybrid applications. *Transactions on Emerging Telecommunications Technologies*, 30(8):e3526, 2019.

[62] Haoxing Li, Fenghua Li, Chenggen Song, and Yalong Yan. Towards smart card based mutual authentication schemes in cloud computing. *KSII Transactions on Internet and Information Systems (TIIS)*, 9(7):2719–2735, 2015.

[63] He Xu, Jie Ding, Peng Li, Feng Zhu, and Ruchuan Wang. A lightweight rfid mutual authentication protocol based on physical unclonable function. *Sensors*, 18(3):760, 2018.

[64] Lu Zhou, Xiong Li, Kuo-Hui Yeh, Chunhua Su, and Wayne Chiu. Lightweight iot-based authentication scheme in cloud computing circumstance. *Future generation computer systems*, 91:244–251, 2019.

[65] Anis Begum Shakeel Ahamed, Navaneetha Kanagaraj, and Maria Azees. Emba: An efficient anonymous mutual and batch authentication schemes for vanets. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1320–1326. IEEE, 2018.

[66] Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, and Athanasios V Vasilakos. Design of secure key management and user authentication scheme for fog computing services. *Future Generation Computer Systems*, 91:475–492, 2019.

[67] Muhammad Qamar Raza and Abbas Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50:1352–1372, 2015.

[68] Xiaokang Zhou, Xuesong Xu, Wei Liang, Zhi Zeng, and Zheng Yan. Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart iot. *IEEE Internet of Things Journal*, 8(16):12588–12596, 2021.

[69] Xiaokang Zhou, Wei Liang, Jinhua She, Zheng Yan, I Kevin, and Kai Wang. Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles. *IEEE Transactions on Vehicular Technology*, 70(6):5308–5317, 2021.

[70] Xianglin Bao, Cheng Su, Yan Xiong, Wenchao Huang, and Yifei Hu. Flchain: A blockchain for auditable federated learning with trust and incentive. In *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*, pages 151–159. IEEE, 2019.

[71] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2):72–80, 2020.

[72] Weishan Zhang, Qinghua Lu, Qiuyu Yu, Zhaotong Li, Yue Liu, Sin Kit Lo, Shiping Chen, Xiwei Xu, and Liming Zhu. Blockchain-based federated learning for device

failure detection in industrial iot. *IEEE Internet of Things Journal*, 8(7):5926–5937, 2020.

[73] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6):1279–1283, 2019.

[74] Ke Zhang, Supeng Leng, Xin Peng, Li Pan, Sabita Maharjan, and Yan Zhang. Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE internet of Things Journal*, 6(2):1987–1997, 2018.

[75] Lorena Cazorla, Cristina Alcaraz, and Javier Lopez. Cyber stealth attacks in critical information infrastructures. *IEEE Systems Journal*, 12(2):1778–1792, 2016.

[76] Cristina Alcaraz and Javier Lopez. Analysis of requirements for critical control systems. *International journal of critical infrastructure protection*, 5(3-4):137–145, 2012.

[77] Chi Harold Liu, Qiuxia Lin, and Shilin Wen. Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, 15(6):3516–3526, 2018.

[78] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[79] Jianbing Ni, Kuan Zhang, Yong Yu, Xiaodong Lin, and Xuemin Sherman Shen. Providing task allocation and secure deduplication for mobile crowdsensing via fog computing. *IEEE Transactions on Dependable and Secure Computing*, 17(3):581–594, 2018.

[80] Pengfei Hu, Huansheng Ning, Tie Qiu, Houbing Song, Yanna Wang, and Xuanxia Yao. Security and privacy preservation scheme of face identification and resolu-

tion framework using fog computing in internet of things. *IEEE Internet of Things Journal*, 4(5):1143–1155, 2017.

[81] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE symposium on security and privacy. S&P 2000*, pages 44–55. IEEE, 2000.

[82] Bruce Gu, Longxiang Gao, Xiaodong Wang, Youyang Qu, Jiong Jin, and Shui Yu. Privacy on the edge: Customizable privacy-preserving context sharing in hierarchical edge computing. *IEEE Transactions on Network Science and Engineering*, 7(4):2298–2309, 2019.

[83] Kai Liu, Xincao Xu, Mengliang Chen, Bingyi Liu, Libing Wu, and Victor CS Lee. A hierarchical architecture for the future internet of vehicles. *IEEE Communications Magazine*, 57(7):41–47, 2019.

[84] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68, 2017.

[85] Nathalie Mitton, Symeon Papavassiliou, Antonio Puliafito, and Kishor S Trivedi. Combining cloud and sensors in a smart city environment, 2012.

[86] Rob Kitchin. The real-time city? big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014.

[87] Md Whaiduzzaman, Anjum Naveed, and Abdullah Gani. Mobicore: Mobile device based cloudlet resource enhancement for optimal task response. *IEEE transactions on services computing*, 11(1):144–154, 2016.

[88] Meng Hsi Chen, Min Dong, and Ben Liang. Joint offloading decision and resource allocation for mobile cloud with computing access point. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3516–3520. IEEE, 2016.

[89] Junaid Shuja, Abdullah Gani, Kwangman Ko, Kyoungyoung So, Saad Mustafa, Sajjad A Madani, and Muhammad Khurram Khan. Simdom: A framework for simd instruction translation and offloading in heterogeneous mobile architectures. *Transactions on Emerging Telecommunications Technologies*, 29(4):e3174, 2018.

[90] Aamir Akbar and Peter R. Lewis. The importance of granularity in multiobjective optimization of mobile cloud hybrid applications. *Transactions on Emerging Telecommunications Technologies*, 30(8):e3526, 2019.

[91] Aamir Akbar, Peter R. Lewis, and Elizabeth Wanner. A self-aware and scalable solution for efficient mobile-cloud hybrid robotics. *Frontiers in Robotics and AI*, 7:102, 2020.

[92] Manisha Verma and Neelam Bhardwaj Arun Kumar Yadav. An architecture for load balancing techniques for fog computing environment. *International Journal of Computer Science and Communication*, 8(2):43–49, 2015.

[93] Jungyeon Baek and Georges Kaddoum. Heterogeneous task offloading and resource allocations via deep recurrent reinforcement learning in partial observable multi-fog networks. *IEEE Internet of Things Journal*, 2020.

[94] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[95] Neal Koblitz. Mathematics of computation. *Elliptic Curve Cryptosystems, 48n*, 1(77):l, 1987.

[96] Anne-Marie Kermarrec and Maarten Van Steen. Gossiping in distributed systems. *ACM SIGOPS operating systems review*, 41(5):2–7, 2007.

[97] Renzo E Navas, Hélène Le Bouder, Nora Cuppens, Frédéric Cuppens, and Georgios Z Papadopoulos. Do not trust your neighbors! a small iot platform illustrating a man-in-the-middle attack. In *International conference on ad-hoc networks and wireless*, pages 120–125. Springer, 2018.

[98] Harsha Vasudev and Debasis Das. An efficient authentication and secure vehicle-to-vehicle communications in an iov. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5. IEEE, 2019.

[99] Christian W Probst, René Rydhof Hansen, and Flemming Nielson. Where can an insider attack? In *International Workshop on Formal Aspects in Security and Trust*, pages 127–142. Springer, 2006.

[100] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.

[101] Elaine Barker, Elaine Barker, William Burr, William Polk, Miles Smid, et al. *Recommendation for key management: Part 1: General*. National Institute of Standards and Technology, Technology Administration, 2006.

[102] Haoye Chai, Supeng Leng, Yijin Chen, and Ke Zhang. A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[103] Jiawen Kang, Rong Yu, Xumin Huang, Sabita Maharjan, Yan Zhang, and Ekram Hossain. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics*, 13(6):3154–3164, 2017.

[104] Nicolas Kourtellis, Kleomenis Katevas, and Diego Perino. Flaas: Federated learning as a service. In *Proceedings of the 1st workshop on distributed machine learning*, pages 7–13, 2020.

[105] Muhammad Shayan, Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. Biscotti: A ledger for private and secure peer-to-peer machine learning. *ArXiv*, abs/1811.09904, 2018.

[106] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

ABSTRACT

SDN-ENABLED EFFICIENT RESOURCE UTILIZATION IN A SECURE, TRUSTWORTHY AND PRIVACY PRESERVING IOV-FOG ENVIRONMENT

by

JAMAL ALOTAIBI

August 2022

Advisor: Lubna Alazzawi

Major: Computer Engineering

Degree: Doctor of Philosophy

The development of intelligent transportation systems (ITS) is aided by the advent of Internet-of-Vehicles (IoV), which is a decentralized network that allows connected vehicles and vehicular ad hoc networks to share data (VANETs). However, today's IoV networks face a number of challenges, including effective resource utilization, security and privacy, trust, information irregularity, etc. In addition, IoV applications have a wide range of Quality-of-Service (QoS) requirements, making it difficult to develop an efficient solution to deal with big data in IoV. Furthermore, the solution should be scalable and extendable, as well as lightweight and cost-effective to maintain. By outsourcing computationally-intensive operations to nearby situated fog nodes, fog computing tackles the fundamental weakness of centralized data processing in cloud computing. Furthermore, as the number of vehicles using the IoV architecture expands, new challenges and requirements emerge, such as scalability, resource efficiency, and secure communication.

In this research work, we look at load balancing, secure communication, privacy preservation, and trustworthy communication in SDN-enabled and fog-based IoV networks. Us-

ing reinforcement learning (RL) approaches, our propose a framework that efficiently distribute tasks in the fog-to-fog and vehicles-to-fog layers. Furthermore, since vehicle data is private and sensitive, further vigilance is required. Authentication of communicating devices is one example of a data security approach. Authentication is used to safeguard data transferred through public channels. Many protocols have been created; nevertheless, typical authentication methods cannot be readily applied to situations that need minimal latency. They're also ineffectual for two reasons: first, they can't keep up with the expanding volume of data collected, and second, they're vulnerable to cyber-attacks. As a result, we attempt to propose a viable solution that is totally resilient and solves the aforementioned issues in this work. We created a lightweight, fog-based authentication mechanism to protect data from IoV devices during transmission. Our method provides low communication costs while meeting high security requirements. Finally, we evaluate and compare the performance of our technique in terms of network parameters including throughput, end-to-end latency, and packet loss rate.

In addition, when private data is exchanged among fog nodes, privacy concerns arise, limiting the usefulness of IoV systems. We propose a Federated Learning-based (FL) and Blockchain-based system for privacy preservation in IoV to address this challenge. Traditional machine learning algorithms are not well suited for distributed and highly dynamic systems like IoV since they train on data with local features. As a result, FL is used to train the global model while preserving the privacy. In addition, our strategy is built on a reputation scheme that evaluates the reliability of vehicles participating in the FL training process. Furthermore, our solution makes use of blockchain technology to ensure trust across numerous communication nodes. All transactions, for example, take place on the

blockchain when local learned model updates from vehicles and fog nodes are shared with the cloud to update the global learnt model. As a result of allowing reputable vehicles to update the global model, our proposed method improves the global model's accuracy, according to the results of our experimental study.

AUTOBIOGRAPHICAL STATEMENT

Jamal Alotaibi received the B.S. degree in Computer engineering from Qassim University, in 2012. He joined Wayne state university 2016 and received M. S. degree in Electrical and Computer Engineering. He worked in STC the telecommunication company as networks engineering. From 2012 to 2015, he is working a lecturer in collage of computer at Qassim University in Saudi Arabia. Since 2018, he is Ph.D. student in Electrical and Computer Engineering Department, College of Engineering, Wayne State University. Mr. Alotaibi research interests include Internet of Things (IoT) applications and security, computer networks, wireless and mobile networks, machine learning, and cybersecurity. Mr. Alotaibi awarded full scholarship from Qassim university in 2016 to do his master degree at Wayne State University, also in 2018 he got another scholarship to do his PhD at Wayne State University from Qassim University.