Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Federated Learning-based Misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles

L. Jai Vinita [*], V. Vetriselvi

Department of Computer Science and Engineering, College of Engineering Guindy, Chennai, 600025, TamilNadu, India

ARTICLE INFO

Keywords:
Internet of Vehicles
6G
Vehicular fog computing
Software-defined fog computing networks
Emergency message scenarios
Federated learning

ABSTRACT

With the 6G-enabled Internet of Vehicles (IoV), the Intelligent Transportation System (ITS) uses new communication technologies and smart data analysis to make transportation systems more innovative and effective. After the emergence of 6G, focusing on emergency message dissemination scenarios in IoVs, software-defined vehicular fog computing (SDVF) networks can be instrumental in enhancing vehicle-to-everything communication for sharing the road, traffic, and accident information with low latency. Nevertheless, the emergency messages are susceptible to intrusion attacks, especially Sybil attacks that could produce numerous bogus clients or collaborate with compromised devices. Thus, we propose the Federated Learning Entrusted Misbehaviour Detection System (FLEMDS) with vehicle selection to support the 6G-enabled vehicles in combating Sybil attackers. As a result, Sybil attack detection is carried out locally in the vehicles employing the federated learning on-vehicle AI technique. The FLEMDS employs a three-level model weight aggregation process at three locations to improve detection accuracy. To minimize the learning and detection latency, federated learning and software-defined vehicular fog computing are combined in the FLEMDS. We employ a fuzzy logic-based FL-vehicle selection (FLBFLVS) technique in the Road-Side Units (RSUs) and Base Stations (BSs) to choose suitable FL-vehicles as clients for the participation of local training in the FL process. The experimental results substantiate that the FLEMDS with FLBFLVS capitulates with a higher detection accuracy of 87% for a minimal number of global aggregations. Furthermore, FLEMDS with FLBFLVS is compared with the state-of-the-art FL-based frameworks. The outcomes show that the proposed schemes yield a faster convergence rate as well as a decrease in the time consumption of computation and communication.

1. Introduction

The fast growth of Internet technology has made it possible for people to use 5G technology in their everyday lives. 5G technology is being developed because people want to use a lot of data and the number of cell phones and cars is growing quickly. As 5G technology gets more mature, countries all over the world have started to look into 6G technology [1]. When 6G comes out, it will give mobile communication networks a lot of new ways to improve. With 6G connectivity, huge mobile networks can do complicated calculations quickly, and the user experience will start to get closer to the real response time [2]. The future of automotive and intelligent transportation systems is the 6G-enabled Internet of Vehicles (IoVs). Communication between moving vehicles is essential for maintaining road safety on the 6G-enabled IoVs. The IoVs that support 6G encompasses Vehicle-to-Everything (V2X) communications. They are Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Sensors (V2S), Vehicle-to-Road-Side Units (V2R), and Vehicle-to-Pedestrians (V2P) communications. V2X communications increase comfort, send warning signals, and improve road safety (including media). Dash cameras are a type of onboard sensor that can record evidence and provide information about the environment around the vehicle. Video data from moving vehicles or communication infrastructure could be used to help people drive, manage traffic, find parking spots, and let ambulances know about accidents. The round-trip delay between vehicles and the cloud is a big reason for the delay. For applications that need to process high-quality video streams in real-time, sending data to the cloud for analytics is not a good idea. Only a small number of vehicles have the right amount of processing power to do advanced data analytics locally.

1.1. Fog computing and software-defined networks

Cisco Systems [3] developed fog computing to address the above issues. Fog computing refers to the processing and analysis of time-critical data at the network's edge level. The Internet of Vehicles

[*] Corresponding author.
E-mail addresses: jai.vinita@gmail.com (L. Jai Vinita), kalvivetri@gmail.com (V. Vetriselvi).

https://doi.org/10.1016/j.adhoc.2023.103153
Received 31 October 2022; Received in revised form 17 February 2023; Accepted 13 March 2023
Available online 21 March 2023

uses V2V and V2I interaction to quickly transmit recovery information to ambulances and other vehicles. Ge et al. [4] present the 5G software-defined vehicular network architecture to meet performance requirements such as minimal dissemination latency and maximum network capacity while taking into account varying vehicle densities. The goal is to bring together technologies such as 5G connectivity, cloud computing, and software-defined networks. The construction of heterogeneous networks using software-defined networking (SDN) and fog computing is the focus of the 6G vehicular networks. The IoVs are exceptionally dynamic and complicated, with strict conditions on extremely low latency, excessive reliability, and enormous connections [5]. By conquering the restrictions of 5G technology, 6G would be a leading promoter of the advancement towards a genuine Intelligent Transportation System and the recognition of the Smart City concept. 6G is essential for IoV to meet the stricter requirements for vehicle communications that 5G only partially met. More importantly, security and privacy should be a key priority because vulnerabilities can lead to disastrous effects. As a result, serious concerns are raised about collecting data from people and sensors.

1.2. Basic architecture

As shown in Fig. 1 our basic architecture holds the SDVF architecture [6]. The SDVF architecture is comprised of several fog clusters and a central SDN controller (SDNC) that makes cycling decisions. The predominant component of the SDVF architecture is the SDNC. A fog cluster is a group of multiple vehicles, RSUs, base stations (BS), and one RSU controller (RSUC). The design of the SDVF architecture includes a distributed control plane that is made up of multiple controllers. Path estimation and video packet tracking are effectively made possible by the combination of VFC with a global centralized SDN controller and local hierarchical multiple RSU controllers. The SDNC manages the entire IoV network. Instead of transmitting specific flow rules, SDNC transmits an abstract policy rule. RSUCs determine how specific policy rules behave within a fog cluster based on their local intelligence. Fog nodes can be vehicles, RSUs, or 5G cellular BSs (enodeB). Fog nodes are nodes made up of storage and computation components. According to Xiao et al. [7], installing fog computing nodes on specific connected and autonomous vehicles (vehicular fog nodes or fog vehicles), including smart buses and taxis, leverages their mobility to offer on-demand fog computing for cost-effectiveness. It is the situation with vehicles travelling at the furthest lowest speed, such as cars stuck in high traffic, airport shuttles, or automobiles on college campuses. To reduce data propagation delays between the monitoring area and the cloud, fog nodes analyse raw video streams.

1.3. Motivation

Physical security compromises are more likely in fog nodes than in cloud servers, which are often physically secure. A malevolent vehicle that pretends to be genuine is called a rogue vehicle. Rogue vehicles can create fake identities or spoof many real IDs by changing the network topology. These attack vehicles use numerous bogus identities to simultaneously halt IoV networks. A rogue node that assigns innumerable false identities is a Sybil attacker [8]. A promising technology is required for misbehaviour detection that uses data from messages to identify vehicles that may be acting maliciously within the system.

According to Al-Otaibi et al. [9], managing rogue vehicles that transmit false information or threaten users' privacy is a critical security issue in IoVs. The information given by other vehicles must enable drivers to make crucial decisions. A rogue node may purposefully introduce misleading data into the network with malicious intent, or unreliable sensors may result in catastrophic network damage. Extreme circumstances may even cause the network to become immobilized. The rogue node can fake the vehicle speed values as well as its computed traffic flow and density values in safety beacon messages. There is a requirement for a misbehaving detection mechanism to detect such bogus network flows. Machine learning (ML) and deep learning (DL)-based misbehaving detection systems (MDS) on IoVs have gotten a lot of attention because of their success in achieving high classification accuracy [10,11]. However, existing IoV misbehaving detection schemes based on machine learning require massive amounts of data to complete model updates. To counter new threats, ML-based misbehaving detection models must be updated. Also, storing and sending information to a central server could put privacy and security at risk. 6G will be an intelligent self-learning network that uses artificial intelligence and deep learning to deal with vehicular network and network management complexity [12]. It is of crucial importance to handle the security and privacy challenges of 6G.

1.4. Federated Learning

The term Federated Learning (FL) was first coined in 2016 by McMahan et al. [13] which brings unique advantages to the security and privacy issues in 6G. Federated Learning [14] offers a technique for cooperative training that enables various entities to create a common machine-learning model. Each device that takes part in federated learning has access to private training data that other clients and the server cannot see. This protects the privacy and security of the data. Only model updates are shared at a central aggregation server, which is usually hosted by one of the parties or by a cloud service provider. A method for identifying passive mobile attacks in 5G VFC was brought forth by Boualouache et al. [15]. This method employs FL to enable secure and confidential collaborative learning and to develop a robust global ML model to detect passive attackers. By putting FL servers at the edge, the system lets clients talk to each other quickly and uses semi-supervised learning to let the data label itself. The aggregation server is a central part that could be a single point of failure if something goes wrong. The FL aggregation process can be carried out at multiple levels. Hayawi et al. [16] suggested a federated learning framework to address the issues of network congestion and capacity. They developed an algorithm for a flexible aggregation node selection process, which chooses the most suitable global aggregation node depending on how busy it is and how long it takes to communicate with it. RSU acts as a fog node that performs local aggregation. Based on what computers can do and how much energy is left over, a method for choosing clients for each communication round was suggested. Though the authors claim that it assists in streamlining the learning process and cutting out delays, it lacks in considering the quality of the communication links with the clients.

1.5. Problem statement

Many research papers have only discussed the issues with vehicle selection strategies in FL-based IoVs. Similarly, only a few papers in the research literature have developed and addressed the issues of FL-based systems for detecting bad behaviour in 6G-enabled IoVs. Before, not enough factors were taken into account to pick the best and most appropriate vehicles for local training of datasets for the use of FL to improve MDS approaches in IoVs with 6G.

1.6. Solution outline

To the best of our knowledge, this is the first effort to analyze the computational capabilities and quality of the communication links of the clients to develop a single framework that focuses on misbehaviour detection (FLEMDS) along with the best vehicle selection strategy (FLBFLVS) in 6G-enabled IoVs. The FLEMDS framework is unique because it uses three-layered federated learning, a sophisticated type of deep learning network, to tell the difference between legitimate network flows and those that are part of a Sybil attack. The framework is made up of a number of distributed SDN controllers that collaborate

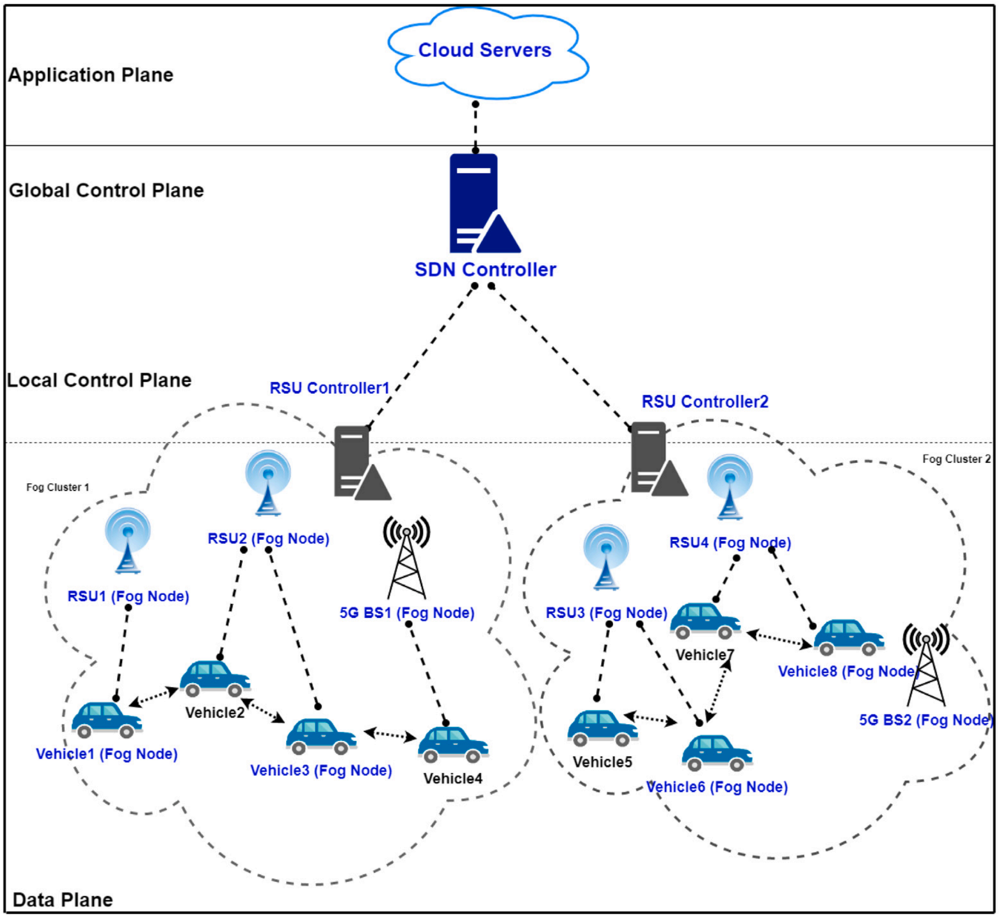


Fig. 1. Architecture of SDVF.

to train an effective misbehaving detection system for the entire vehicular network. The number of model aggregations and synchronization rounds that an FL operation goes through depends on the batch size that is employed. Before model aggregation, each client (FL-vehicle) trains on its entire local dataset. A model update generated by a node for each round is frequently intended to be temporary, but the aggregator must durably store it until model aggregation is finished [17]. The aggregator nodes selected in FLEMDS are capable of storage and computing. The SDNC, RSUCs, RSUs, and base stations are the aggregator nodes in FLEMDS. By default, the aggregator side discards the model updates once the aggregated model is permanently stored, while each node may maintain model updates according to its local data retention policy. Furthermore, the resources of a few fog vehicles may be partially or completely utilized. To fully take part in federated learning with fog vehicles, it is important to choose suitable vehicular fog nodes. Over time, vehicles change their minds about whether or not to take part in the FL process until they reach an equilibrium state. The equilibrium state might not last for very long. So, it makes more sense to choose some connected fog vehicles that take part in the federated learning process. There could be a large number of uncovered fog vehicles with high computational capabilities. Because of these reasons, it is important to choose fog vehicles as FL vehicles and think about uncovered fog vehicles. The FLEMDS framework uses the FLBFLVS method, which only lets a set of fog vehicles train locally as clients. The novelty of FLEMDS architecture also lies in the FLBFLVS approach, which uses the fuzzy-logic concept to select only the fog vehicles that take part in local training for the FL process. FL-vehicles are chosen using the FLBFLVS approach based on factors such as RSSI_{LTE}, $\text{RSSI}_{\text{IEEE802.11p}}$, residual energy, available memory, and current data records.

1.6.1. Contributions

The paper makes four major contributions:

1. We point out the main problems with the various learning-based misbehaving detection techniques and the existing FL-based frameworks in vehicular networks. We specifically point out the difficulties in employing the distributed and centralized solutions currently in use without considering the effectiveness of communication and the uneven characteristics of vehicle data. We also summarized the existing approaches to vehicle selection in federated learning.
2. Additionally, we discuss the significance of using the fuzzy logic technique [18] for choosing the FL-vehicles from a huge pool of vehicular fog nodes to deliver a fast global update. The FL-vehicles' contributions to the overall model are different because they have different sets of features and training, which makes this more difficult.
3. We suggest a security architecture for SDVF that is built around a speedy traffic accident rescue scenario and has three layers of federated learning to spot Sybil attacks. So, the identification can be done in the car while taking into account the learning environment's unbalanced vehicle data. After three levels of aggregation have been set up, the global weights are sent from SDNC to RSUs through RSUCs. When in range, the FL-vehicles interact with the RSUs, or BSs, which act as fog nodes to gather and incorporate the most recent updates from the global model weights into their local models. The final global model is generated for edge vehicles. Additionally, in the case of our FL proposal, only model weights are provided to the global model, successfully protecting the privacy of the vehicle data in the SDVF.

Table 1
Summary of acronyms.

Acronym	Definition
6G	Sixth-Generation
5G	Fifth-generation
ITS	Intelligent Transportation Systems
IoV	Internet of Vehicles
VFC	Vehicular Fog Computing
SDVF	Software-defined Vehicular Fog Computing
FL	Federated Learning
MDS	Misbehaving Detection System
FLEMDS	Federated Learning Entrusted Misbehaviour Detection System
RSU	Road-Side Unit
BS	Base Station
enodeB	Evolved Node B
FLBFLVS	Fuzzy Logic-Based Federated Learning Vehicle Selection
FL-vehicle	Federated Learning-vehicle
V2X	Vehicle-to-Everything
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2S	Vehicle-to-Sensors
V2R	Vehicle-to-Road-Side Units
V2P	Vehicle-to-Pedestrians
SDN	Software-defined Networks
SDNC	Software-defined Network Controller
RSUC	Road-Side Unit Controller
PKI	Public Key Infrastructure
PCA	Pseudonym Certificate Authority
ML	Machine Learning
DL	Deep Learning
MDS	Misbehaving Detection Systems

4. Lastly, we compare our federated learning entrusted misbehaving detection (FLEMDS) method to the state-of-the-art FL-based frameworks and test it against a simulated threat model in the SDVF architecture. We also estimate the FLBFLVS procedure based on the fuzzy logic method.

1.7. Paper organization

The following is the organization of the paper: Section 2 discusses the issues in various learning-based misbehaving detection techniques, the existing federated learning frameworks on the Internet of Vehicles, and also the current vehicle selection strategies for FL. Section 3 presents a threat model in a speedy traffic accident rescue scenario concerning SDVF networks. Section 4 describes the proposed System. The experimental setup and comparative results are explicated in Section 5. Section 6 contains the conclusion and future work. For ease of reading, the definitions of acronyms used in this paper are summarized in Table 1.

2. Related work

In this section, we first look at the problems with different learning-based misbehaviour detection systems for next-generation 6G-enabled vehicle networks. In existing federated learning frameworks for vehicles, an investigation of the complications and consequences is carried out.

2.1. Learning-based misbehaving detection systems on the internet of vehicles.

To issue digital keys to the OBUs of vehicles and RSUs in accordance with ETSI and IEEE standards, a vehicular Public Key Infrastructure (PKI) system is in place [19]. To sign messages sent back and forth, digital keys are used. This verifies the identity of the sender and ensures that the message is real. These signatures serve as a means of identifying a specific vehicle. As a result, it would be simple to follow a vehicle that uses the same key to sign all of its messages. Pseudonym authentication is handled by the Pseudonym Certificate Authority (PCA). The pseudonym authentication process creates a pseudonym for the vehicle ID. The pseudonym can include the valid open key, the length of time the key is good for, and the digital signature of PCA. The vehicle ID can be saved to track the ID of a vehicle that uses a pseudonym. A Sybil attack can happen when a misbehaving vehicle uses multiple valid pseudonym certificates at the same time. So that users' privacy is protected, the vehicular PKI gives out many certificates with fake names for the same vehicle. A vehicle can therefore frequently change its signature to avoid being tracked. An attack is made possible by the pseudonymization of cars, in which a single vehicle pretends to be several different on-road entities at once. This form of malicious activity is known as a Sybil attack. Even though the mobile PKI protects the network from attacks from the outside, it can still be attacked by rogue nodes on the inside. A node with valid keys could sign the messages while sending false information.

Misbehaviour Detection System (MDS) is one of the most important and basic security measures for stopping security attacks. It has been used extensively in conventional cyber networks for ages. To enhance the accuracy of a centralized detector in the IoTs, Li et al. [20] built a cooperative misbehaving detection framework. At the moment, considering factors such as the placement location and learning levels of IDS, the misbehaving detection systems on IoVs fall into the following categories.

2.1.1. Distributed learning-based misbehaving detection system on individual clients

In VANETs, each node, like a vehicle or RSU, has its own IDS module that looks for local attempts to break in. To protect VANETs from false information assaults, Zaidi et al. [21] presented an intrusion detection algorithm based on communication intervals and vehicle density. It operates without expensive gear like Lidar, radar, or cameras and is independent of any infrastructure (such as RSU). Zhang et al. [22] suggested a machine learning-based MDS using differential privacy while maintaining privacy.

2.1.2. Centralized learning-based misbehaving detection system on RSUs or cluster heads

The IDS implemented on the RSU is considered a substitute for the vehicles. A hierarchical IDS relying on BUSNet was developed by Tian et al. [22], in which buses serve as clustering nodes to collect routing packets together with the dissemination of data among cars. The RSU learns the regular behaviour model using the collected flows after receiving the actual network flows from the bus nodes. The RSU traces the whole vehicular network using a well-trained model that can recognize network packets and data packets and quickly sound an alarm if it sees anything out of the ordinary. As a central RSU monitors the complete network, the method experiences SPOF (single-point-of-failure). Additionally, since it cannot expand properly as the system gets bigger. For such solutions to work, the centralized RSU needs to have enormous storage, computing, and transmitting resources so that it may efficiently manage vast volumes of traffic. Cluster head nodes in this classification are in charge of spotting intrusion attempts. Wahab et al. [23] suggest an intelligent detection model that boosts the rate of detection and obtains the least overhead even with the high mobility of VANET. Within each cluster, multi-point relay vehicles (specialist nodes that are in charge of packet forwarding) work together to collect training data, which is then processed in real time. Subba et al. [24] created an IDS with a classifier with a lightweight component drawn on game theory with two players to detect malicious cars and decrease communication costs. To cut down on IDS traffic, it simulates the exchange of messages by making the cluster head and a bad vehicle play a game. This is done by using a monitoring method based on the

Nash equilibrium. This method slows down the network less than both centralized and decentralized IDS, and it can keep fighting attackers even when RSU is not available. But this method can only get limited information about flow from the cluster head, so it cannot be used to train a detection system to keep track of the whole network. For this type of solution to work, the cluster head must be a vehicle with a lot of computing power and a stable network bandwidth so that connections do not drop out.

2.1.3. Collaborative learning-based misbehaving detection systems

Both individual vehicles and centralized elements like RSUs and cluster heads are equipped with collaborative learning-based IDS. Sedjelmaci et al. [25] created a powerful clustering method and suggested a compact intrusion detection system for VANET to handle high node mobility and quick changes in topology. The intrusion detection of three agents makes up the entire network: the cluster runs the local IDS; the cluster head runs the global IDS; the RSU runs the global decision system. Three detection agents must work together to identify malicious vehicles so they can be detected with high accuracy. Aloqaily et al. [26] used deep belief-decision trees to create an attack detection technique for detecting attacks on automobiles. Nevertheless, each intrusion detection requires participation from the cars, cluster heads, and RSUs, which adds a significant burden to the system. So that network latency does not get in the way, these frequent exchanges and active engagement need constant and high bandwidth. Multiple SDN controllers are installed on the base stations, and Shu et al. [27] suggest a collaborative IDS drawn on distributed SDN as a defense against security attacks in vehicular ad-hoc networks. A generative adversarial network (GAN) is employed to construct a collaborative IDS. It lets multiple SDN controllers work together to train an IDS model for the whole vehicular network, not just a local network, to fix the problem of biased flow caused by deploying multiple SDNs in a decentralized way. For software-defined connected vehicles, Kim et al. [28] developed an MDS based on machine learning where the vehicles examine the incoming traffic and send some chosen traffic flows to the SDN controller. The SDN controller uses the SVM classification method to train a multi-class classifier based on these data flows. Vehicles get the parameters of the trained model so they can use them to find other vehicles that are acting up. For vehicular networks, Mirzaee et al. [29] suggested a two-layer IDS technique based on machine learning. The suggested method calls for the Edge IDS at the supervisory layer and the on-vehicle IDS as the main node to work together to find threats.

2.2. Existing federated learning frameworks on the internet of vehicles

Many pieces of writing have shown that federated learning is used in IoVs to reach many goals. Its goals include keeping data private, making routing decisions, putting in place systems to find people who are acting strangely, and a few other things. An FL-based framework was put forth by Mowla et al. [30] to enable on-device identification of jamming attacks in FANET using the UAV clients with the federated learning model. The selection of client groups for a constrained global model is carried out by employing a client group prioritization method based on the Dempster–Shafer theory. Zhou et al. [31] suggested a two-layer FL-based model for intelligent object detection that produces accurate and efficient learning outcomes in 6G Internet-of-Vehicles. A two-layer FL framework was made and put into place to improve the traditional cloud computing architecture in vehicular networks that support 6G. An embedded TFL-CNN was built by leveraging convolutional neural networks for training to employ local input and just share the parameters. A technique called multi-layered heterogeneous model selection and aggregation was used to take into account both the local and global contexts of each car and RSU. This made the training process much more effective. The intelligent object detection process was then created using a context-aware learning approach. Magdum et al. demonstrated a cooperative V2X-based FL system that combines LTE and 802.11p interfaces [32]. The proposed V2X-based FL system asserts to be more effective than the system that simply uses a 4G LTE interface for V2X interaction since it makes use of radios already present in vehicles.

Xiao et al. [33] suggest a completely decentralized FL-enabled framework for investigating an undirected topology with time variation matching to the dynamic multi-agent system. The authors claim that the system can quickly converge and is partially immune to the effects of random data collection and dynamic networks that change over time. Zhang et al. [34] present a secure cloud–edge-end collaboration PIoT (BASE-PIoT) architecture based on blockchain and AI. With three common blockchains working with PIoT, the system claims to guarantee data security and intelligent computation offloading, as well as versatile resource allocation, secure data sharing, and diffserv guarantee. They also suggest a federated deep actor-critic work offloading method powered by blockchain to solve the problem of safe and low-latency compute offloading.

Since vehicles generate a tremendous amount of data, the traditional machine learning-based IDS approaches are not the best options. The fact is that data needs significant computing, network, and private resources. Traditional vehicular networks cannot have fully distributed IDS frameworks because cars do not have enough computing power. Vehicle networks' need for low latency and limited connectivity also makes it hard to use an edge computing model. But the federated learning paradigm is useful because it lets devices be trained by the machine learning models that use the least amount of network resources and do not share data with a central server. In most federated learning MDS architectures, a single global aggregator is used, with the possibility of a single point of failure. And also, they are unable to scale to handle large amounts of data and adapt to the vehicle's behaviour. The performance of a single aggregator drops as the number of vehicles increases. Also, in a fair way to choose clients, the local training might not be enough for clients with limited computing resources and bad wireless conditions. It leads to disproportionately long upload times. Existing federated learning methodologies often create a single separate local model for every vehicle to train the local sensitive data, then employ a centralized model globally to interchange the trained weights of connected vehicles that take part in the FL process. Due to the different properties of the local data samples and the different types of vehicles involved in the process, the traditional FL method cannot make all vehicles perform well in the same way.

2.3. Vehicle selection strategies for federated learning in the internet of vehicles

The vehicles taking part in federated learning are not all the same, they may have different types of datasets with different sizes, qualities, and distributions. This is called the non-independent and identically distributed (non-IID) problem. If you select vehicles randomly, you might end up with vehicles with few resources or clients with less data. This could make it harder to reach the goal of a certain level of accuracy and lead to a large number of learning rounds. There are several approaches to federated learning vehicle selection, including biased client selection [35], unbiased client selection [36,37], synchronous aggregation [38], and asynchronous aggregation [39–42]. The performance of FL, like training time, model upload latency, and model convergence, is affected by the number of vehicles chosen and the local data of those vehicles. The authors present the link between the number of selected clients and the FL training process [36]. Then, incorporate federated learning with client selection (FedCS), in which the server chooses as many clients as possible in each communication round to speed up global model convergence in resource-constrained edge networks. Yoshida et al. [43] proposed a selection approach based on a multi-armed bandit model that takes into account different computing and communication resources. The authors suggest a later-is-better principle-based client selection approach, in which chosen clients with minimal bandwidth and energy are updated in each learning round [44]. Analogous findings are presented in [45], where a stochastic selection scheme based on the efficiency and fairness of the participant is established to achieve a trade-off between convergence time and accuracy. In order to select more clients for the FL training process, the authors of [46] take into account CPU frequency, memory, and energy. Lai et al. [47] use multiple predefined criteria to select the device with the best computational capability and sample quality in order to improve the accuracy of the FL model. Wang et al. [48] investigate the change in the model before and after training, in which the local dataset is assumed to be non-IID. Then reinforcement learning is used by taking into account how accurate the model is and how many communication rounds there have been. The authors of [49] calculate the reputation of each vehicle by testing the accuracy of the local model to build a client selection architecture. Maintaining the reputation of every FL user, though, is challenging. In fact, selecting more clients means injecting the model with more samples. Evidently, the research discussed above only takes into account the number of chosen clients, and it does not adequately address the clients' capacities for FL training.

The authors state that in synchronous FL, there is only one launch end and one aggregation end for the single global model [42]. Each round begins at the same time for each client. When all models are in the environment, federated aggregation is run without setting a goal for learning rounds. In response to frequent model updates and bottlenecks, the authors of [50] implement semi-asynchronous FL. Then, they theoretically assess how partial clients may affect convergence performance. The authors in [38] design a traditional synchronous FL protocol that includes choice of the client, node layout, and reporting. Even with bottlenecks, this protocol is suitable for the majority of FL scenarios. The bottlenecks are remedied effectively in [43] by incorporating the limit in learning rounds.

Previous research studies did not investigate the computational capabilities and communication link qualities of the vehicles when participating in FL local training. To solve these problems in 5G and lay the groundwork for 6G, an FL-entrusted misbehaving detection system (FLEMDS) has been proposed. The FL process looks at many aggregators from different network domains that have a lot of computing power. So, there is a guarantee that data from vehicles will be shared and gathered correctly so that updates from vehicles in all parts of the vehicular network will be better. FLEMDS with FLBFLVS only lets vehicular fog nodes that can compute and store data be chosen as clients for the training process. This makes sure that update and upload times are kept to a minimum.

3. Threat model in speedy traffic accident rescue scenario

When 5G or 6G vehicular networks are combined with software-defined networks (SDN) and fog computing, they can achieve ultra-low transmission delay, high throughput, and the ability to support the dynamic nature of vehicular network functions for a low cost. We are considering 6G SDVF networks because our application, the emergency Scenario, is all about sending out safety beacons with very low latency. The basic SDVF architecture [6] incorporates the scenario for speedy traffic accident rescue and uses fog nodes to broadcast accident-related video required by hospitals to dispatch ambulances for both congestion avoidance and speedy traffic accident rescue. The SDVF network sends video packets from the surveillance area to the hospital by dividing roads into different road segments with unique segment IDs. The information that vehicles on the road share with the RSUs is then forwarded by the RSUs to the RSU controller of the fog cluster. After obtaining real-time topology from RSUC, SDNC finds the shortest path to the target. The suggested system also takes advantage of the fact that video packets can be routed via RSUC from one fog cluster to another or in a multi-hop fashion, i.e., V2V, V2R, and V2B communications. Maintaining up-to-date routes to neighbouring RSUs will eventually be important compared to other mobile nodes because vehicles demand a high level of familiarity with RSUs. The SDNC and the RSUC are the most important parts of the SDVF architecture for video streaming between fog nodes and the hospital. Fog nodes must be distributed around the network, and global–local controllers must be in charge of managing them. Services that are dispersed over numerous fog nodes must be managed. SDNC figures out the best way for the ambulances to get to the accident scene as quickly as possible. The video is looked at locally in cars, RSUs, or cell phone base stations called fog nodes.

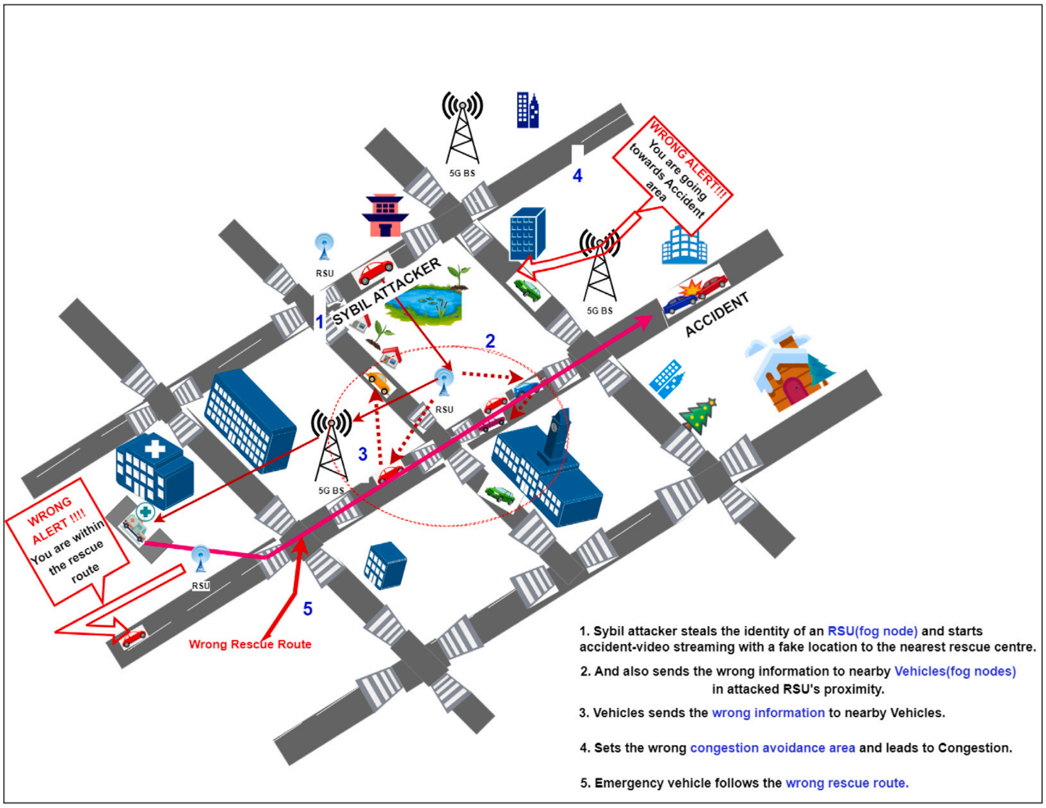
In vehicular networks, vehicles regularly broadcast basic safety messages (BSM), also known as safety beacons, for accurate positioning, localization of new neighbouring vehicles, or emergency purposes by claiming their identities. A Sybil node can effortlessly allege multitudinous identities without being detected. Identity authentication is not a good way to keep Sybil attacks from happening in IoVs. The SDVF design for the rescue scenario has added two variants of the Sybil attack known as emergency message alteration and fake emergency message generation in which either the position of the vehicles is modified or a new emergency message is generated by claiming the identities. Fig. 2 depicts a Sybil attacker discovering the identity of a fog node (for example, an RSU). Then, it sends a fake location-based accident video to the hospital. As a compromised fog node, the Sybil attacker first looks for hospitals near the fog cluster. If the RSUC of the current fog cluster cannot find the hospital nearby, it sends the video feeds and emergency alerts to the next fog cluster. The main job of a cellular BS (also a fog node) is to provide wide coverage. If a BS in the fog cluster detects a hospital nearby after receiving a video stream, it will transfer the streams to the hospital. The hospital also analyzes the received video feeds for accident-related data, like the severity of the accident, the number of people who were affected in the accident, the number of vehicles jumbled in the collision, etc. Eventually, the hospital sends an ambulance to the accident site based on the claimed position in the safety beacon that it has received. The wrong alerts act as false notifications, such as You are on the rescue route or You are nearing an accident location, and are sent to vehicles moving towards the accident location through V2V communications. Altogether, an erroneous congestion area is formed by the fog nodes. As drivers get closer to the made-up accident site, the wrong virtual congestion avoidance region is made based on where the accident happened (shown by the red square). The incorrect rescue route is chosen, as shown by the pink line from the hospital to the scene of the accident. The goal of the vehicles' emergency rescue plan is to change the route so that ambulances can get to the scene of an accident. But the route is wrongly predicted by the Sybil attacker.

4. Proposed system: Federated Learning Entrusted Misbehaving Detection System (FLEMDS) for SDVF networks

This section explains the step-by-step process of creating a federated learning misbehaving detection system model for SDVF architecture using the FLBFLVS algorithm. The subsection discusses Sybil attack detection, which will be implemented at the edge vehicles using the FLEMDS algorithm. We also precisely analyze the correctness of the verification. We show how a Sybil attack can be found in the local data of a vehicle using the threat model.

4.1. System model

The Federated Learning-entrusted misbehaving detection system for SDVF networks is shown in Fig. 3. It performs three-level flexible and adaptive model aggregation that detects Sybil attacks. The FLEMDS architecture is scalable for vehicle participation and topology. Three-level model aggregation aims to enhance the detection accuracy of the MDS system. The procedure for the FLEMDS comprises six steps. Refer to Table 2 for the notation symbols and their definitions used throughout the article.



1. Sybil attacker steals the identity of an RSU(fog node) and starts accident-video streaming with a fake location to the nearest rescue centre.
2. And also sends the wrong information to nearby Vehicles(fog nodes) in attacked RSU's proximity.
3. Vehicles sends the wrong information to nearby Vehicles.
4. Sets the wrong congestion avoidance area and leads to Congestion.
5. Emergency vehicle follows the wrong rescue route.

Fig. 2. Threat model in a speedy traffic accident rescue scenario considering a software-defined vehicular fog computing architecture.

Table 2
Summary of notations.

Notation symbols	Definition
K	Total number of model updates
$W_{initial}$	Initial detection model weights
c	A specific fog cluster
D_f	Local Sybil attack dataset instances
W_f	Global model weights
W_{t+1}	weights at time t+1
L_f	Global loss function
l_f	Local loss function
u_f	Number of local updates
n_f	Number of local attack dataset instances
e_f	Residual energy
m_f	Available memory
P_f	Available CPU capacity
J_f	Number of data records
LA1	Number of level-1 local aggregators
LA2	Number of level-2 local aggregators
q_f	Number of FL-vehicles
n_f	Number of local attack dataset instances
$w1_f$	Level-1 local weights
$w2_f$	Level-2 local weights
p	Pseudonym
pdb	Pseudonym database

4.1.1. Parameter estimation and the creation of an initial global model

The chosen FL-vehicle acts as a client in the federated learning architecture for 6G-enabled IoVs. The clients do parameter estimation by using vehicle features like pseudonym, sender-ID, receiver-ID, speed, claimed transmission time, reception time, position, acceleration, and heading. To train a model for finding bad behaviour, we added an attack-ID feature with two different values. This feature is used to differentiate between two types of network flows, such as normal flows and Sybil attack flows. The initial detection model, $W_{initial}$ is trained globally at the SDN controller using a few instances of the attack dataset. The updated weights W_f are then distributed to the fog clusters participating in the model training phase of the federated learning process. The weights W_f are the parameters that can be learned and are changed when the model is being trained. A shared weight update from the FL-vehicles that is sent to the global model also makes it possible to use a federated averaging method.

4.1.2. Fog cluster selection for training

Consider that there are V_f fog vehicles, R RSUs, and B base stations as fog nodes which are disseminated to *N* fog clusters. Due to the fog clustering phenomenon, the fog vehicles with the proper computational and storage capabilities get the opportunity to take part in local training, which minimizes the communication overhead between the vehicles and the SDN controller. Additionally, the local Sybil attack dataset owned by the fog vehicles in a fog cluster c is denoted by D_f.

At the initialization of a round t, the SDNC initiates the fog clusters N. Suppose K is the total number of model updates in round t, and the SDNC sends W_f as the latest model weight to RSUCs (fog cluster heads). While the new update begins, RSUCs forward it to their corresponding fog clusters. The fog clusters can be selected in a fixed or random manner. At the starting point, W_f equals $W_{initial}$. At a later point, W_f assigns a value of W_{t+1} soon after the SDNC has been updated. The SDNC adds a new fog cluster from *N* that has not participated in each round or update.

4.1.3. Fuzzy logic-based FL-vehicle selection (FLBFLVS)

Each FL round must include a certain ratio of fog vehicles that are chosen inside a fog cluster, and other connected vehicles are dropped based on certain criteria. We assume that a pseudonym authentication procedure is followed to verify the issued certificates of fog vehicles to prove they are legitimate [51]. The main reason behind the selection of vehicle clients for local training is the following: most vehicles are unable to participate in the learning process due to their lower computational capabilities and memory. If they were likely to be forced,

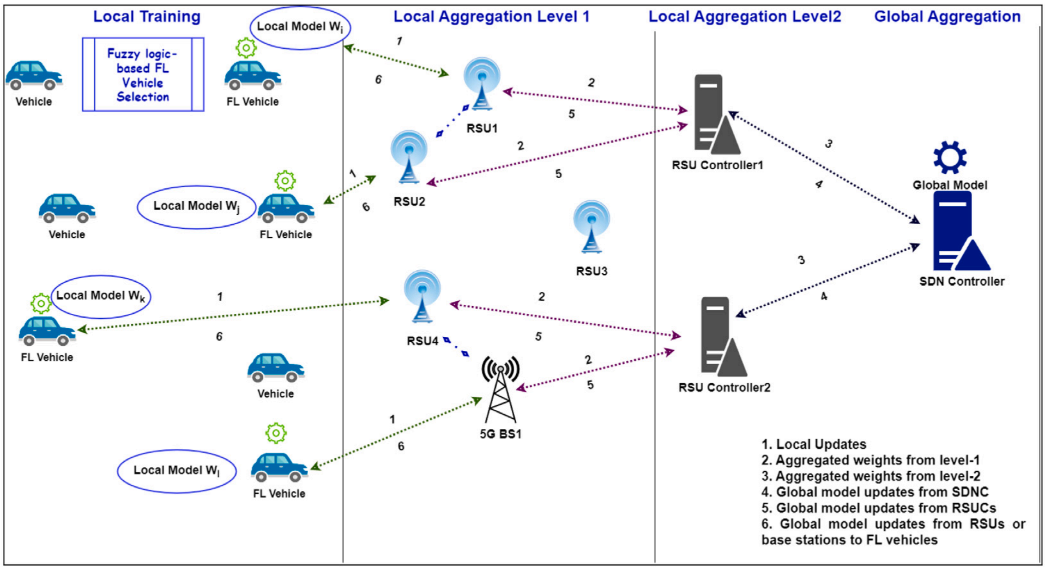

Local Training
Local Aggregation Level 1
Local Aggregation Level2
Global Aggregation
Fuzzy logic-based FL Vehicle Selection
Vehicle
FL Vehicle
Local Model W$_i$
Local Model W$_j$
Local Model W$_k$
Local Model W$_l$
RSU1
RSU2
RSU3
RSU4
5G BS1
RSU Controller1
RSU Controller2
Global Model
SDN Controller
1
2
3
4
5
6
1. Local Updates
2. Aggregated weights from level-1
3. Aggregated weights from level-2
4. Global model updates from SDNC
5. Global model updates from RSUCs
6. Global model updates from RSUs or base stations to FL vehicles

Fig. 3. Federated learning entrusted misbehaviour detection system (FLEMDS) model.

the learning process would be slowed. Thus, an efficient fuzzy logic-based FL-vehicle selection (FLBFLVS) technique is executed at RSUs and base stations based on factors such as accessible memory, data records, residual energy, and link strength of LTE and IEEE 802.11p connections. An Elliptic Curve Cryptography (ECC) is used to guarantee communication privacy and source authentication which is provided by the current IEEE 802.11 standard IEEE 1609.2 /cite. The fuzzy logic technique [18] is utilized for the selection of suitable FL-vehicles to be involved in the process of federated learning. FL fog vehicles are selected periodically. The selected nodes accumulate the required parameters through V2I (RSUs, BSs) and V2V communications. The FLBFLVS selection algorithm calculates the convenience of vehicles to decide whether to include them in the selected vehicle list or not. Algorithm 1 outlines the algorithm of the FLBFLVS procedure.

A mathematical technique that employs linguistic variables is termed fuzzy logic. There are three steps in the fuzzy process, namely fuzzification, fuzzy inference systems, and defuzzification. Fuzzification is the process of converting the input values into linguistic values using membership functions. A set of IF-THEN rules known as a fuzzy inference system is used to translate inputs into outputs. Using fuzzy operations, the various inputs are combined in the initial section of rules using an IF statement with the combination of OR and AND. The second section, which is the linguistic value of the output, represents the outcome of the IF statement. The opposite of fuzzification is defuzzification. The fuzzy inference system's linguistic output is transformed into an accurate numerical value.

The accuracy of the data gathered may be impacted by changes in network circumstances such as interference and weather. Fuzzy logic is employed in this situation because it is a reliable method for resolving issues with imperfect information. We investigate the case of slow-moving cars and buses at airports and university campuses. We use LTE link strength and IEEE802.11p link strength, residual energy, available memory, and the current number of data records as metrics. The link strength is measured based on the received signal strength indicator (RSSI). The variables $\mathrm{RSSI}_{\mathrm{LTE}}$ and $\mathrm{RSSI}_{\mathrm{IEEE802.11p}}$ have linguistic values of poor, average, and good. The variables residual energy, available memory, and current data records have low, medium, and high linguistic values. Vehicular fogs should have high LTE and IEEE 802.11p link strength (RSSI) values to guarantee communication with RSUs and base stations. The range of linguistic variables $\mathrm{RSSI}_{\mathrm{LTE}}$ $\mathrm{RSSI}_{802.11\mathrm{p}}$, residual energy, available memory, and data records are presented in Fig. 4. A classic fuzzy set function is between the values of 0 and 1, denoting the degree of the membership function of the input or output linguistic variables in a particular set. Table 3 depicts the rules for the five parameters that act as inputs to fuzzy variables to produce fuzzy production rules. The output linguistic variable, convenience, signifies the chances of the fog vehicles becoming FL-vehicles. The higher the value of the convenience variable, the greater the chances of the fog vehicles becoming FL-vehicles. This structure was used to select the FL-vehicle in a fog cluster at each RSU or base station. The most suitable fog vehicular nodes are selected as FL vehicles.

Table 3
FLBFLVS parameter rules.

Rules	RSSI (LTE)	RSSI (802.11p)	RE	M	DR	Convenience (output)
1	Good	Good	High	High	Low	High
2	Good	Good	High	High	Medium	High
3	Good	Good	High	High	Medium	Medium
4	Average	Average	High	High	High	Medium
5	Poor	Poor	Low	Low	Low	Low
6	Poor	Poor	Low	High	High	Low

The format of the rules is demonstrated by the following instance:

If ($\mathrm{RSSI}_{\mathrm{LTE}}$ is good) OR ($\mathrm{RSSI}_{802.11\mathrm{p}}$ is good) AND (residual energy (RE) is high) AND (memory (M) is high) AND (data record (DR) is low) then (convenience is high).

4.1.4. Local training and upload

FL-vehicles undergo local training using local Sybil attack data D_f and transmit the local updates(weights) to RSUs in their proximity. The $\gamma \in \mathrm{D}_\mathrm{f}$ has two parts $(\mathrm{X}^\gamma, \mathrm{Y}^\gamma)$, X^γ represents an input vector whereas Y^γ represents an output vector.

A local model is trained by the selected FL-vehicle using a finite sum objective of the following form:

$$\min \sum L_\mathrm{f}(w) \tag{1}$$

In Eq. (1), $\mathrm{L}_\mathrm{f}(\mathrm{w})$ is a global loss function across each FL-vehicle that is to be minimized. $\mathrm{l}_\mathrm{f}(\mathrm{w})$ is a local loss function across FL-vehicle's training over Sybil attack dataset instances D_f.

$$L_\mathrm{f}(w) = \frac{1}{n_\mathrm{f}} \sum_{i \in D_\mathrm{f}} l_\mathrm{i}(w) \tag{2}$$

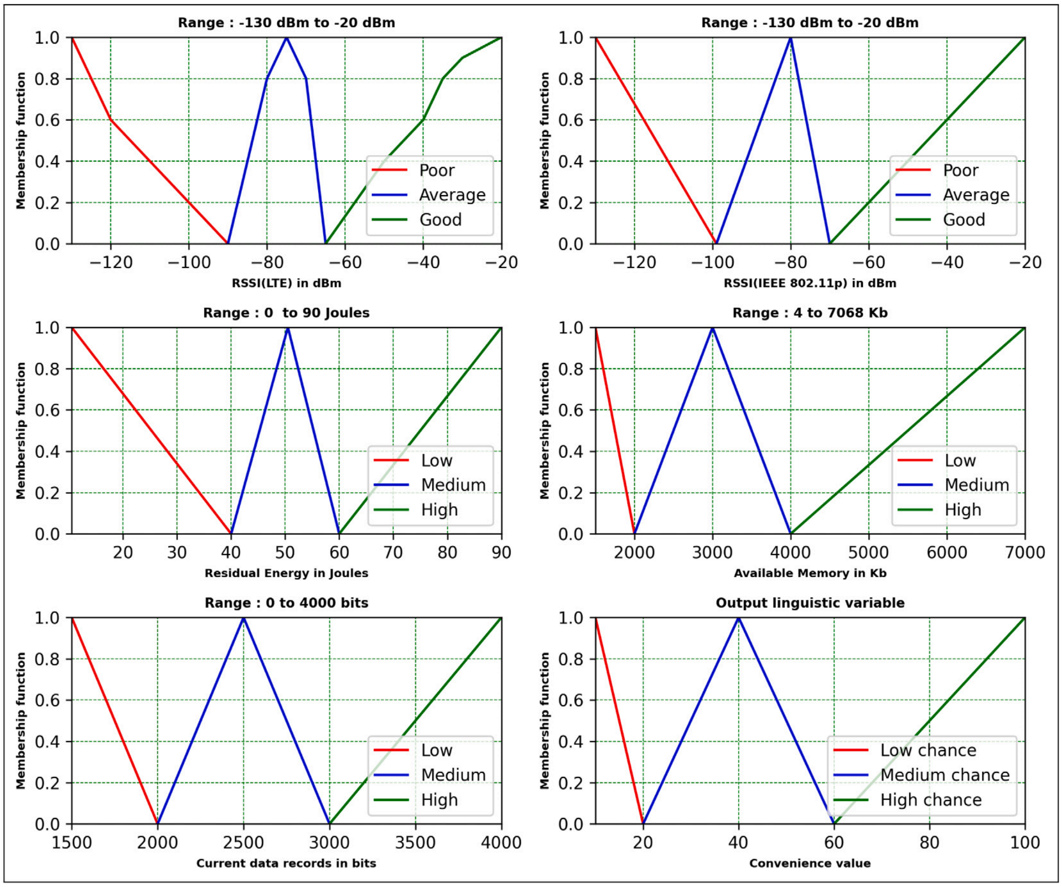


Fig. 4. Range of linguistic input and output variables (a) $\mathrm{RSSI}_{\mathrm{LTE}}$ (b) $\mathrm{RSSI}_{\mathrm{IEEE802.11p}}$ (c) Residual energy (d) Available memory (e) Current data records (f) Convenience.

Algorithm 1 FLBFLVS procedure within each fog cluster

```
Input : List of vehicular fog nodes $V_f$ in a cluster with the metrics data LTE link quality IEEE 802.11p link quality, residual energy, available memory, and data records, $V_f = \{v_1, v_2, v_3, .....v_f\}$
Output : Set of suitable FL vehicles $B_f = \{b_1, b_2, b_3, ......b_f\}$
function FLBFLVS
    Initialization :
    $t \leftarrow 0$                                  ▷ time slot
    $N_f$                                             ▷ number of fog vehicles at t
    At each period $T$
    $T \leftarrow t + T$                              ▷ Initial FL selected vehicle list is empty
    $B_f \leftarrow NULL$
    for (i =0 to $N_f$) do
        Each fog vehicle $V_f$ collects its inputs
                ▷ $\mathrm{RSSI}_{\mathrm{LTE}}$, $\mathrm{RSSI}_{802.11\mathrm{p}}$, residual energy RE, available memory M, and current data records DR.
        $convenience_i = fuzzy\_function(RSSI_{LTE}, RSSI_{802.11p}, RE, M, DR)$
                ▷ Each vehicle computes its convenience value.
        if ($convenience_i$ is high) then             ▷ $v_i$ becomes an FL-vehicle(client)
            $B_f \leftarrow v_i$
        end if
    end for
    return $B_f$
end function
```

where,

$$l_i(w) = f(x_i, y_i; w) \tag{3}$$

In Eq. (3), $l_i(w)$ is a function with the ith feature x_i associated with label y_i and model weights w.

Local weights are updated on each selected FL-vehicle in B_f.

$$w_f(t) = w_f(t) - \alpha \Delta L_f(w_f(t, bs_f)) \tag{4}$$

bs_f refers to batch size and α refers to the learning rate that should be greater or equal to zero.

The number of local updates u_f as follows:

$$u_{\mathrm{f}} = \frac{E_{\mathrm{f}} n_{\mathrm{f}}}{bs_{\mathrm{f}}} \frac{e_{\mathrm{f}} m_{\mathrm{f}} P_{\mathrm{f}}}{J_{\mathrm{f}}} \tag{5}$$

The notations E_f, n_f, BS_f, e_f, m_f, P_f, and J_f represent the conditions of fog vehicles, E_f denotes the training iterations, n_f denotes the number of local attack dataset instances, bs_f denotes mini-batch size across fth FL-vehicle, e_f denotes residual energy, m_f denotes available memory, P_f denotes available CPU capacity, and J_f denotes the number of data records. After u_f local updates, all the FL-vehicles under RSU or BS (level 1 local aggregators) la1 ∈ LA1 send the updated parameters to the local aggregators at level 1 to calculate the local aggregation.

4.1.5. Local aggregation level-1

To achieve weight aggregation, RSUs or base stations serve as middle brokers to gather and aggregate data from all connected vehicles within their coverage area. It includes learning parameters as well as contextual information like vehicle locations and navigational instructions. RSUs then interact with the RSUC to upload aggregated model parameters and associated contextual data. After the global aggregation, the RSUC provides the updated model weights to RSUs, which were received from SDNC. RSUs and BSs subsequently send these updated model weights to specific vehicles. As referred to in Fig. 3, the RSU1 and RSU2 perform the process of local aggregation at level-1. RSU1 and RSU2 send the aggregated model weights to RSUC1. Similarly, selected FL-vehicles under RSU3 and 5G BS perform local aggregation at level-1, and the aggregated model weights are sent to the RSUC2.

The calculation of level-1 local weights aggregation is as follows:

$$w1_{\mathrm{f}}(t) = \frac{\sum_{f=1}^{q_{\mathrm{f}}} n_{\mathrm{f}} w_{\mathrm{f}}(t)}{q_{\mathrm{f}}} \tag{6}$$

where q_f denotes the number of FL-vehicles and n_f represents the number of local attack data-set instances.

4.1.6. Local aggregation level-2

At level 1, RSUs and BSs act as fog nodes. Since the fog nodes have high computational power, RSU controllers and SDN controllers do not infer the model quality. At level-2, RSU controllers act as aggregators which retrieve aggregated model weights from the RSUs and base stations from the level-1 to perform the level-2 aggregation process. As demonstrated in Fig. 3, the RSUC1 and RSUC2 perform the second level of aggregations and send the aggregated weights to SDNC.

The level-2 local aggregation weights $w2_f(t)$ are calculated at RSUCs by taking the average of local aggregation parameters $w1_f(t)$ from multiple level-1 local aggregators la1 ∈ LA1, which is defined as follows:

$$w2_{\mathrm{f}}(t) = \frac{\sum_{f=1}^{la1} w1_{\mathrm{f}}(t)}{LA1} \tag{7}$$

4.1.7. Global aggregation

The SDN controller is both a global controller and an aggregator for the SDVF network as a whole. It performs the global aggregation process at the third level by getting the local aggregated weights as part of level-2 from RSUC1 and RSUC2. The proposed three-layer federated learning system FLEMDS assures data privacy, effective knowledge-sharing, and improved Sybil attack detection accuracy only by requiring the transfer of model weights. Once the FLEMDS reaches the ϵ number of selections and two levels of local aggregations, RSUCs forward the aggregated weights for global aggregation to SDNC. If the SDNC controller fails, one of the best RSUCs will take over as the secondary global aggregator. Thus, it overcomes the challenge of a single failure in traditional FL techniques. For simplicity, we have set RSUC1 as the default secondary global aggregator. As part of our work in the future, we want to set up a way to choose the best RSUC to replace the SDNC.

The global aggregation weight W_f is calculated at SDNC by taking the average of local aggregation parameters $w2_f(t)$ from various level-2 local aggregators la2 ∈ LA2 within each fog cluster n ∈ N, which is defined as follows:

$$W_{\mathrm{f}} = \sum_{n=1}^{N} \left[\sum_{f=1}^{la2} \frac{w2_{\mathrm{f}}(t)}{LA2} \right] \tag{8}$$

We consider u_f local updates and $\epsilon 1 = LA1$ number of level-1 local aggregations and $\epsilon 2 = LA2$ number of level-2 local aggregations before accomplishing one global aggregation. Only a few global aggregations are executed instead of executing at each round which helps to minimize the communication overhead only after ϵ_i number of local aggregations at level-1 and level-2 from R number of communication rounds, where i represents the level number $i \in 1,2$. It also helps to reduce the latency by minimizing the number of communications to reach the SDNC for global aggregation, which is at the cloud level.

4.1.8. Back-propagation of the global weights

Once after completing the global aggregation, the global aggregator (SDNC) updates RSUCs, RSUs (or BSs), and FL-vehicles with the end global model. Before the next FL-vehicle selection and local training process, the model is synchronized with the global weights. The synchronization ensures that all of the chosen FL-vehicles receive the previously learned weights from the global model. It will be helpful when a particular vehicle has not participated in the prior FL-vehicle selection process. Once the global model reaches the convergence level, the FL process is stopped. The final global model is deployed in all the vehicles to detect Sybil attacking nodes at the edge level.

4.2. Sybil attack detection

To detect the fake safety beacon sent by the Sybil attacker, the global model is trained using the local attack dataset. The data instances are differentiated between normal and Sybil attack flow by making the usage of the concept known as pseudonyms linkage proposed in [52]. The objective of pseudonyms linkage is to precisely link all the pseudonyms that originate from the same vehicle. According to the machine learning linking option, it determines whether the two reported pseudonyms are linked to the same vehicle utilizing the formerly computed features. A few reported pseudonyms are flagged as linked pseudonyms are considered as an attack. The linked pseudonyms are stored in the pseudonym-database pdb. The pseudonym in the safety message is examined and if it belongs to the linked pseudonyms, then it is detected as abnormal network flow. Otherwise, it is detected as a normal network flow.

The federated learning entrusted intrusion detection system procedure is explained in Algorithm 2, which calls the FLBFLVS function, which is used to select the suitable fog node. The function receives metric data from fog vehicles. The complete step-wise explanation of the FLEMDS algorithm is given in Section 4.1. After model convergence at R=500, the final global model of FLEMDS is procured and it is deployed on the vehicles. The fog vehicles with the newly learned FLEMDS model weights can be in one of two modes: the training mode or the testing mode. If the fog vehicles are in the training mode, then they can make predictions about the network flows and provide local updates back to the RSUs, BSs, RSUCs, and then to SDNC for contributing to the global FLEMDS W_f model. If the fog vehicles are in the testing mode where only network flow predictions on pre-trained attack data instances are performed. The testing mode saves communication overhead as the FLEMDS weights are not required to be sent to the RSUs for the local aggregation. The testing mode is used when the fog vehicle is not suitable for the selection of participation in local training. Whenever the new safety messages reach the vehicles at the edge level through V2V or V2I communications, FLEMDS is utilized to verify whether the message belongs to normal flow or abnormal flow.

Algorithm 2 FLEMDS algorithm for each round

```
1: $Input : Initial\ parameter\ set\ (W_{initial})\ consists\ of\ pseudonym\ p,\ sender\text{-}ID\ sid,\ receiver\text{-}ID\ rid,\ position\ pos,\ speed\ sp,\ heading\ hd,\ acceleration\ ac,\ attack\text{-}ID\ aid, pseudonym - database\ pdb,\ F,\ C,\ N,\ LA1,\ LA2,\ \epsilon,\ R$
2: $Output : W_f(t)\ detects\ the\ normal\ and\ abnormal\ network\ flow$
3: function FLEMDS
4:     $Initialization\ :\ Initial\ global\ model\ W_{initial}$
5:     $SDNC\ declares\ N\ fog\ clusters\ for\ each\ round$
                                                                        ▷ Runs under each cluster
6:     for (each level-1 local aggregator la1 ∈ LA1 in parallel) do
7:             FLBFLVS()         ▷ Fuzzy logic-based federated learning vehicle selection function is called
8:         $Initialise\ w_f(t)\ at\ each\ selected FL\ vehicle\ f \in F$
9:         for (each selected FL-vehicle f ∈ F in parallel) do
10:            $calculate\ local\ updates\ w_f(t),\ u_f\ times\ using\ equations$ (4) $and$ (5)
11:            $calculate\ local\ loss\ l_f,\ using\ equation$ (3)
12:        end for
13:        $calculate\ Global\ loss\ L_f(w),\ using\ equation$ (2)
14:        $calculate\ level - 1\ local\ aggregation\ w1_f(t)\ using\ equation$ (6)
15:    end for
16:    for (each level-2 aggregator la2 ∈ LA2 in parallel) do
17:        $calculate\ level - 2\ local\ aggregationw2_f(t)\ using\ equation$ (7)
18:    end for
19:    if (t is an integer multiple of $\epsilon 1\ and\ \epsilon 2$) then
20:        $calculate\ globalparameter\ W_f(t)\ using\ equation$ (8)
21:    end if
22: end function
                                                              ▷ Sybil Attack Detection-Testing Phase
23: Whenever a safety beacon reaches a vehicle
24: pseudonym of the source present in the beacon is verified
25: if (pseudonym p ∈ pseudonym-database pdb) then
26:     $abnormal flow$
27: else
28:     $normal flow$
29: end if
```

4.2.1. Complexity analysis

We analyze the time and space complexity of the three-layered FL-based Sybil attack detection algorithm in the worst-case scenarios, which are presented in Algorithm 2. We consider the function FLEMDS to assess the time complexity of the proposed Algorithm 2. In particular, Algorithm 2 requires $(q_f + LA1 + LA2 + N)$ number of additions and $(q_f + LA1 + LA2)$ number of multiplications for generating a global gradient or model (Eqs. (6)–(8)). As such, the time complexity of the algorithm is $O(n^2)$ since the number of local updates u_f and the number of iterations E_f is much smaller than n. The time complexity for Algorithm 2 is O $(n^2 + n^2 + n) = O(2n^2 + n)$. The time complexity of calculating the convenience value using fuzzy logic in Algorithm 1 is $O(n^2 + n^2 + n) = O(2n^2 + n)$ (Step 12). Compared with local training, the complexity for backpropagation is $O(n^3)$ at the vehicle side. The overhead of the proposed algorithms on the aggregators' side is minor and can be ignored. Space complexity is the amount of memory a model needs to run well. The distributed nature of the federated model means it uses less memory. In contrast, the feed-forward nature of the proposed federated model that uses extreme machine learning has made it easier to use less space. The space complexity of Algorithms 1 and 2 is 3.25 MB.

4.2.2. Correctness proof

We present correctness proof that the Algorithm 2 enables vehicles to detect a Sybil attack whenever the new safety messages reach the vehicles at the edge level through V2V or V2I communications. We consider a threat scenario where a Sybil attacker discovers the identity of a fog node (for example, an RSU). Then, it sends a fake location-based accident video to the hospital. As a compromised fog node, the Sybil attacker first looks for hospitals near the fog cluster. If the RSUC of the current fog cluster cannot find the hospital nearby, it sends the video feeds and emergency alerts to the next fog cluster. We focus on Sybil attack detection with our algorithm, where the vehicles can operate. The initial global model W_f at SDNC is trained with an instance of attack dataset, and the model is transmitted to the B_f=5 selected FL vehicles within a fog cluster through RSUC and RSU. And then, at the starting point, R = 1 round of communication was used to train models using the three-layered FL method as described in Section 4.1. In particular, p = [1, 2, 3, 4], or 5 FL-vehicles, were used for distributed training. For instance, each FL dataset D_f1, D_f2, D_f3, D_f4, and D_f5 had examples from 2 classes of Sybil attacks, but each vehicle's FL dataset was biased towards a different class by 50%. The global loss L_fw is calculated (step 13) from the local losses l_1w, l_2w, l_3w l_4w, and l_5w. The local weights w_f1, w_f2, w_f3, w_f4, and w_f5 are calculated (step 12) for 5 FL-vehicles. Following u_f local updates (step 10), all 5 vehicles within RSU1's vicinity send the local weights to RSU. Similarly, RSU2 receives the local weights from their FL-vehicles. RSU1 and RSU2 calculate the level-1 aggregation weights $w1_f1$, $w1_f2$ respectively, $w1_f3$ and $w1_f4$ (step 14). RSUC1 and RSUC2 at level-2 receive level-1 aggregated weights. They perform the level-2 local aggregation process using 4 weights to obtain $w2_f1$ and $w2_f2$ (step 17). SDNC, as the global aggregator, performs the global aggregation process to calculate the global model W_f (step 20). The global model W_f is disseminated to the FL-vehicles through RSUC1, RSUC2, RSU1, RSU2, RSU3, and BS1 after the first communication round. To form a new local model, the weights of 5 FL vehicles under RSU1, RSU2, and other vehicles under RSU3, and BS1 are updated with the global weights. The new, updated local model is sent to the new set of selected FL vehicles for the next communication round. The same process is repeated for R = 500 communication rounds for model convergence. Following model convergence, the final model will be deployed in the vehicles.

Table 4
Simulation parameters.

Parameters	Values
Open Street Map bound box	80.2331, 12.9918; 80.2676, 13.0165
Mobility Scenario	Urban/Highways (10 km)
Threat	Constant location attack (Sybil)
Number of attackers	1
Normal Traffic density	$\approx$ 200 vehicles/km^2
Rush-hour traffic density	$\approx$ 700 vehicles/km^2
Channel frequency	5.890e9
SDN Controller	1
Propagation model	Two ray
Vehicle communication range	300 m
RSU Communication range	1000 m
Antenna model	Omnidirectional
Bit rate	18 Mbit/s
PHY model	IEEE 802.11p
MAC model	EDCA

Now the vehicles are ready for the Sybil attack detection phase. Whenever the safety beacons reach the vehicles, they automatically decline the message if it belongs to the attack class (steps 23–29). A detailed explanation of the simulated attack dataset and the classification of the dataset is given in Section 5.1.

5. Performance evaluation

In this section, we confer the performance results that were carried out to analyze the FLBFLVS procedure and the FLEMDS model. The FLEMDS with FLBFLVS process is compared with FLEMDS without any vehicle selection method. The FL-vehicles are chosen randomly if no vehicle selection mechanism is used for any FL framework. Also, the FLEMDS is compared with two different aggregations algorithm-based MDS that are state-of-the-art architectures in FL. So far as we know, this is the first approach to figure out how well-equipped vehicles that can be used for effective local training affect the development of FL-enabled MDS for the 6G-enabled Internet of vehicles by employing FLBFLVS.

5.1. Experimental setup

To experiment with the FLEMDS and FLBFLVS, we use the simulated threat model for the SDVF network incorporated into the speedy rescue traffic accident scenario. The simulation parameters are presented in Table 4. We use the Veins simulation framework to run our project. This framework combines the network simulator OMNET++ and the traffic simulator SUMO. Veins comes with detailed IEEE 802.11p and LTE V2X standards [53]. In contrast to LTE-V2X, which uses the PC5 sidelink interface specified in 3GPP Release 14 and improved in 3GPP Release 15, DSRC uses IEEE 802.11p-based standards as the foundation for wireless communication. IEEE 802.11p and LTE-V2X use IEEE 1609 WAVE standards for network layer and security protocols. The simulated threat dataset is utilized to implement FLEMDS with the vehicle selection method, FLEMDS without the FL-vehicle selection method, and three state-of-the-art baseline frameworks. The baselines are used for comparing vehicle selection strategies and also for comparing misbehaviour detection accuracy. Baseline-1 denotes a greedy framework (FedCS) that employs a one-layer aggregation algorithm-based MDS with client selection [36], baseline-2 indicates a traditional one-layer (FedAvg) aggregation algorithm-based MDS with random client selection [13]. The baseline-3 indicates a two-layer (TFL-CNN) aggregation algorithm-based misbehaviour detection framework [31]. The development took place on the Google Colaboratory platform under Python 3.7 using PyTorch 1.8.1 and a Graphics Processing Unit (GPU). In all the frameworks, the vehicles are given local training with the simulated attack datasets.

Table 5
Experimental parameters for FL.

Parameters	Notations	Values
Received signal strength indicator of LTE link	$\mathrm{RSSI}_{\mathrm{LTE}}$	−130 to −20 dBm
Received signal strength indicator of IEEE 802.11p link	$\mathrm{RSSI}_{802.11\mathrm{p}}$	−130 to −20 dBm
Residual energy	RE	0 to 90 Joules
Memory	M	4 to 7068 Kb
Data Records	DR	0 to 4000 bits
Number of fog clusters	N	2
Number of fog vehicles	V_{f}	120
Number of selected fog vehicles for each round in each fog cluster	B_{f}	10
Number of RSUs (fog nodes)	R	4
Number of base stations (fog nodes)	B	2
Number of RSU controllers	RSUC	2
Number of SDN controllers	SDNC	1
Number of communication rounds	R	500
Local batch size	bs_{s}	10
Number of local iterations	E_{f}	50
Learning rate	α	0.01

5.1.1. Simulated attack dataset

In the simulation, each vehicular fog node is equipped with LTE and IEEE 802.11p interfaces. Multiple RSUs and base stations are placed in various domains. The RSUs and base stations find out the fog vehicles that are located within their communication proximity. It broadcasts multi-hop probe messages to all the vehicles and collects response messages from fog vehicles. The simulated dataset [6] consists of Sybil attack flows and normal traffic flows. We treat the normal type as a normal flow and the Sybil attack type as an abnormal flow. The dataset consists of the messages that are broadcast and received by the RSUs, eNodeBs, and vehicle On-Board Units (OBUs).

The simulated Sybil attack dataset consists of two safety beacon forging attacks, two vehicle densities (Regular traffic and Rush traffic), and one Sybil attacker. Each parameter set is repeated for randomization. Each simulated dataset includes a ground truth JSON file that incorporates all the messages from normal vehicles and Sybil vehicles, as well as the messages that each vehicle has received. The ground truth file in JSON format is converted into a.csv format. We use the.csv ground truth file for detection. The features such as pseudonym, timestamp, sender-ID, receiver-ID, receiving time, position, speed, heading, acceleration, pseudonym-database and attack-ID are used to generate the attack dataset. Among the 11 features, the two features pseudonym-database and attack-ID are manually included in the .csv files. The parameters are combined by considering 100 instances of normal and attack configuration, thus we have generated 11,004 rows of data instances. The simulated attack dataset requires pre-processing to improve the training and testing dataset to adequately evaluate the detection performance for next-generation SDVF networks under emergency message dissemination scenarios. The three steps are followed during the pre-processing of the dataset:

1. Delete the records with incorrect formatting and missing features.
2. To perform federated learning, divide the entire larger dataset into smaller datasets.
3. Distribute the smaller datasets to the selected vehicular fog nodes.

5.2. Comparative results

We first evaluate the time (latency) consumption to determine the impact of the FL-vehicle selection scheme FLBFLVS in the FLEMDS.

In Fig. 5, the proposed FLEMDS with FLBFLVS is compared with baseline methods by considering several communication rounds and time consumption as metrics. Then we assess the FLEMDS in terms of learning efficiency. During the aggregation process, the shifting of the model weight and the decline of loss are observed by investigating the

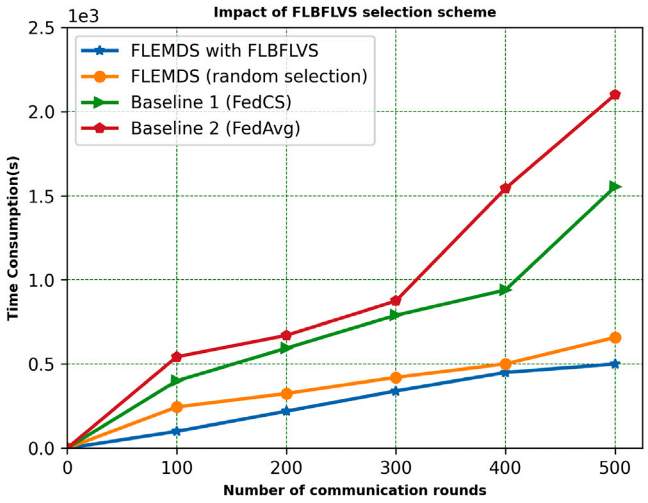


Fig. 5. Impact of FLBFLVS vehicle selection method.

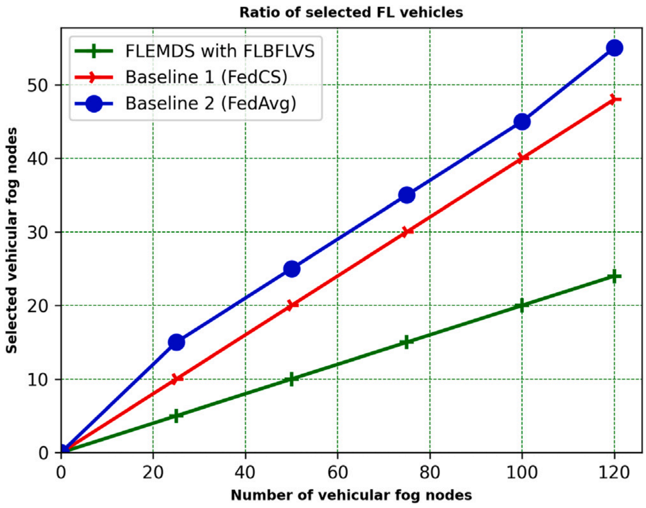


Fig. 7. Ratio of selected FL-vehicles.

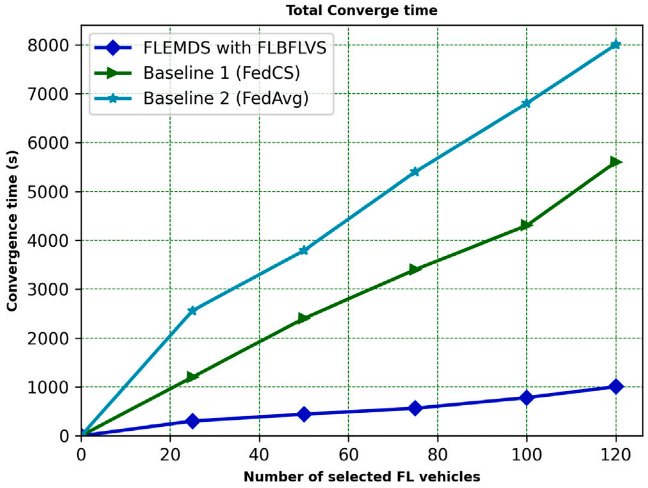


Fig. 6. Total convergence time.

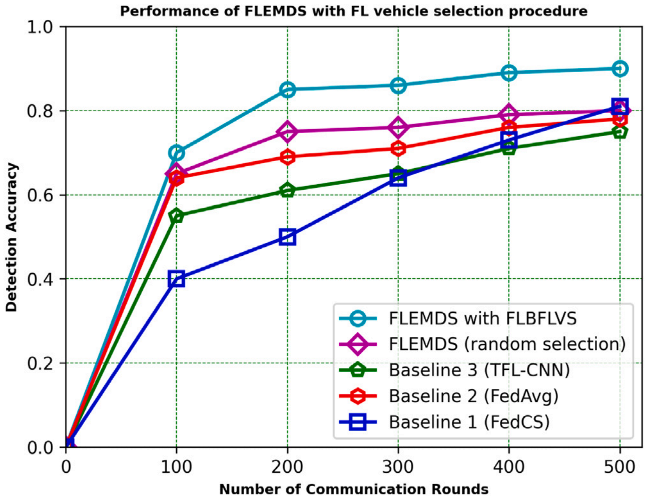


Fig. 8. Detection accuracy of FLEMDS with FLBFLVS procedure.

training process of the FLEMDS. The learning rate is set to 0.001 and the total number of communication rounds is set to 500. Based on the FLEMDS with FLBFLVS method, the average latency of each communication is less than other schemes. The reduction of time consumption in FLEMDS with the FLBFLVS method is relatively more than other schemes since FLEMDS tends to select vehicular fog nodes with good computational and storage capabilities as well as high qualities of LTE and IEEE 802.11p links. Table 5 lists the experimental parameters and their corresponding values used for the proposed work.

Around 120 vehicular fog nodes are used throughout the training process. It is noted that when the selected number of FL-vehicles from the fog vehicles increases, there is a rise in convergence time. The proposed FLEMDS converges faster than the other FL-baseline methods even with the minimal number of selected FL-vehicles which is illustrated in Fig. 6.

Fig. 7 shows the ratio of selected vehicular fog nodes as FL client vehicles for the FLBFLVS approach in FLEMDS and the random selection of FL client vehicles in FLEMDS. The estimated metric is interpreted as the ratio among the selected fog vehicles as FL clients versus the total number of fog vehicles. The proposed FLEMDS works with the FLBFLVS technique, which permits the reduction of the ratio compared to the random selection. The reason behind the performance is that only suitable and eligible fog vehicles are taken into consideration to participate in the federated learning. It is measured that while using the FLBFLVS approach, only 20% of vehicular fog nodes participated as FL clients in each round.

The performance of the final global model across the normal and abnormal types in the traffic flows is described by accuracy. It measures the proportion of accurate Sybil attack predictions to all predictions. In our system, it is the proportion of Sybil samples that were correctly identified in all of the samples. Aiming at a vehicular networking scheme for FLEMDS, the consequence of non-IID (non-independent and identically distributed) data is not contemplated on the performance of FLEMDS. The performance of FLEMDS with the fuzzy logic-based FL-vehicle selection procedure is demonstrated in Fig. 8. It can be found that compared to state-of-the-art schemes, the proposed solution can still achieve the best performance, and the detection accuracy is beyond that of the state-of-the-art FL-based frameworks. The computation time is calculated from the time taken by the RSU or base station to select the vehicular clients. It also includes the computation time for the vehicle clients to calculate their local loss values and the time taken by selected vehicles to perform local updates. This hints that FLEMDS does not arouse any further communication cost for FL-vehicle selection, instead it acquires a 2-times reduction in the number of communication rounds using 20% of vehicular clients compared to baseline frameworks and gets higher detection accuracy performance.

Table 6
Number of global aggregations (N_{GA})

Total number of communication rounds R	Baseline-2 framework (N_{GA})	Baseline-3 framework (N_{GA}) when $\epsilon = 10$	FLEMDS framework (N_{GA}) when $\epsilon 1 = 10$, $\epsilon 2 = 5$
500	500	50	10
400	400	40	8
300	300	30	6
200	200	20	4
100	100	10	2

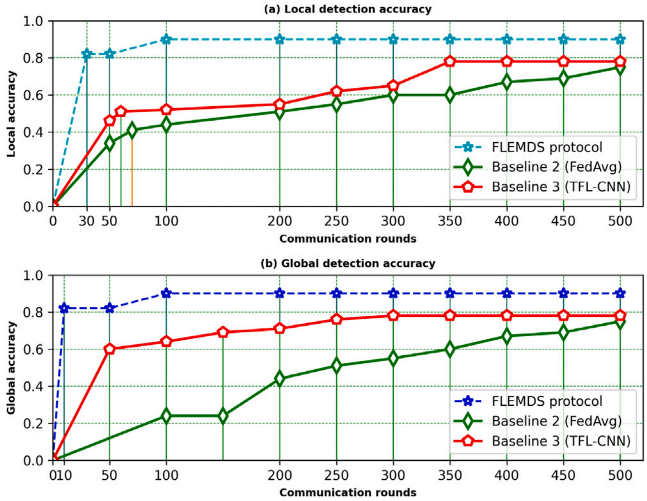


Fig. 9. Local accuracy versus Global accuracy.

To reach the target accuracy, the number of global aggregations executed for the proposed system and state-of-the-art baseline frameworks is shown in Table 6. Suppose N_{GA} denotes the number of global aggregations then the baseline-2 framework requires N_{GA}=R global aggregations after client training whereas the baseline-3 framework requires $N_{GA} = \frac{R}{\epsilon}$ global aggregations. The metric ϵ indicates the number of local aggregations. The FLEMDS requires only $N_{GA} = \frac{R}{(\epsilon 1 \times \epsilon 2)}$ number of global aggregations. The metrics $\epsilon 1$ and $\epsilon 2$ denote the number of level-1 local aggregations and level-2 local aggregations, respectively. For instance, if the total number of communication rounds R is 500, then the baseline-2 framework performs 500 global aggregations, the baseline-3 framework performs 50 global aggregations, and the FLEMDS requires only 10 global aggregations to reach 87% accuracy. The differentiation between local detection accuracy and global detection accuracy is shown in Fig. 9 by comparing three frameworks. Figs. 9(a) and (b) exhibit that the global detection accuracy for FLEMDS outperforms the state-of-the-art baseline-2 and baseline-3 frameworks.

6. Conclusion and future work

To support 6G-enabled IoVs, a secure and privacy-preserving federated learning entrusted misbehaviour detection framework is designed for the SDVF architecture that is adapted to a speedy traffic accident rescue scenario. A Sybil threat model is introduced in a speedy traffic accident rescue scenario and that is simulated in [6]. The simulated attack dataset is extracted, pre-processed, and utilized for the implementation of the FLEMDS framework. The FLEMDS performs three levels of aggregation at three locations, such as the RSU or base station, the RSU controller, and the SDN controller, respectively, that serve as a security and privacy foundation for 6G. To undergo local training in FLEMDS, vehicular fog nodes opt to be client vehicles. To achieve faster convergence, FL client vehicles are selected based on a fuzzy logic approach known as FLBFLVS. It is shown from the results that the detection accuracy of FLEMDS is also improved with the FLBFLVS procedure. It is also proved through the experiments that FLEMDS reaches higher detection accuracy in a smaller number of global aggregations when compared to state-of-the-art FL-based baseline frameworks.

We intended to extend our work by optimizing the FL-vehicle selection problem. We also intend to employ a secured aggregation algorithm to protect the model weights that are propagated across the network. As part of the extended work, we plan to investigate a fog clustering mechanism to be implemented by the SDN controller.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] K. David, J. Elmirghani, H. Haas, X.-H. You, Defining 6G: Challenges and opportunities [From the guest editors], IEEE Veh. Technol. Mag. 14 (3) (2019) 14–16, http://dx.doi.org/10.1109/MVT.2019.2922512.

[2] A.H. Sodhro, M. S. Obaidat, S. Pirbhulal, G.H. Sodhro, N. Zahid, A. Rawat, A novel energy optimization approach for artificial intelligence-enabled massive Internet of Things, in: 2019 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS, 2019, pp. 1–6, http://dx.doi.org/10.23919/SPECTS.2019.8823317.

[3] IoT, from cloud to fog computing, 2015, https://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing. (Accessed 03 March 2015).

[4] X. Ge, Z. Li, S. Li, 5G software defined vehicular networks, IEEE Commun. Mag. 55 (7) (2017) 87–93, http://dx.doi.org/10.1109/MCOM.2017.1601144.

[5] D.P. Moya Osorio, I. Ahmad, J.D.V. Sánchez, A. Gurtov, J. Scholliers, M. Kutila, P. Porambage, Towards 6G-enabled internet of vehicles: Security and privacy, IEEE Open J. Commun. Soc. 3 (2022) 82–105, http://dx.doi.org/10.1109/OJCOMS.2022.3143098.

[6] L. Jai Vinita, V. Vetriselvi, Impact of sybil attack on software-defined vehicular fog computing (SDVF) for an emergency vehicle scenario, in: G. Ranganathan, X. Fernando, Á. Rocha (Eds.), Inventive Communication and Computational Technologies, Springer Nature Singapore, Singapore, 2023, pp. 809–825.

[7] Y. Xiao, C. Zhu, Vehicular fog computing: Vision and challenges, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, 2017, pp. 6–9, http://dx.doi.org/10.1109/PERCOMW.2017.7917508.

[8] L. Jai Vinita, V. Vetriselvi, A survey on security aspects of internet of vehicles, in: Emerging Technologies and Applications for a Smart and Sustainable World, Vol. 1, Bentham Sciences, 2022, pp. 41–67, http://dx.doi.org/10.2174/9789815036244122010006 (Chapter 3).

[9] B. Al-Otaibi, N. Al-Nabhan, Y. Tian, Privacy-preserving vehicular rogue node detection scheme for fog computing., Sensors (Basel, Switzerland) 19 (2019) 1–18, http://dx.doi.org/10.3390/s19040965.

[10] A.R. Gad, A.A. Nashat, T.M. Barkat, Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset, IEEE Access 9 (October) (2021) 142206–142217, http://dx.doi.org/10.1109/ACCESS.2021.3120626.

[11] J. Grover, V. Laxmi, M.S. Gaur, Sybil attack detection in VANET using neighbouring vehicles, Int. J. Secur. Netw. 9 (4) (2014) 222–233, http://dx.doi.org/10.1504/IJSN.2014.066178.

[12] C.D. Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, M. Liyanage, Survey on 6G frontiers: Trends, applications, requirements, technologies and future research, IEEE Open J. Commun. Soc. 2 (2021) 836–886, http://dx.doi.org/10.1109/OJCOMS.2021.3071496.

[13] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP, Vol. 54, 2016, http://dx.doi.org/10.48550/ARXIV.1602.05629, URL https://arxiv.org/abs/1602.05629.

[14] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R.G.L. D'Oliveira, H. Eichner, S.E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P.B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S.U. Stich, Z. Sun, A.T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F.X. Yu, H. Yu, S. Zhao, Advances and open problems in federated learning, 2019, http://dx.doi.org/10.48550/ARXIV.1912.04977, URL https://arxiv.org/abs/1912.04977.

[15] A. Boualouache, T. Engel, Federated learning-based scheme for detecting passive mobile attackers in 5G vehicular edge computing, Ann. Telecommun. Telecommun. 77 (2022) 201–220, http://dx.doi.org/10.1007/s12243-021-00871-x.

[16] Z. Trabelsi, K. Hayawi, A.W. Malik, T. Qayyum, M. Ali, Fog enabled federated learning framework for vehicular AdHoc networks (Vanets), 2022, Available at SSRN 4054171.

[17] K.R. Jayaram, V. Muthusamy, G. Thomas, A. Verma, M. Purcell, Adaptive aggregation for federated learning, 2022, http://dx.doi.org/10.48550/ARXIV.2203.12163, URL https://arxiv.org/abs/2203.12163.

[18] W.I.S.W. Din, S. Yahya, R. Jailani, M.N. Taib, A.I.M. Yassin, R. Razali, Fuzzy logic for cluster head selection in wireless sensor network, AIP Conf. Proc. 1774 (1) (2016) 050006, http://dx.doi.org/10.1063/1.4965093, arXiv:https://aip.scitation.org/doi/pdf/10.1063/1.4965093, URL https://aip.scitation.org/doi/abs/10.1063/1.4965093.

[19] ETSI (European Telecommunications Standards Institute), ETSI EN 302 637-2 - Part 2: Specification of Cooperative Awareness Basic Service V1.4.1 (2019-04), Vol. 1, ETSI, 2019, pp. 1–22.

[20] W. Li, W. Meng, M.H. Au, Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments, J. Netw. Comput. Appl. 161 (2020) 102631, http://dx.doi.org/10.1016/j.jnca.2020.102631, URL https://www.sciencedirect.com/science/article/pii/S1084804520301053.

[21] K. Zaidi, M.B. Milojevic, V. Rakocevic, A. Nallanathan, M. Rajarajan, Host-based intrusion detection for VANETs: A statistical approach to rogue node detection, IEEE Trans. Veh. Technol. 65 (8) (2016) 6703–6714, http://dx.doi.org/10.1109/TVT.2015.2480244.

[22] T. Zhang, Q. Zhu, Distributed privacy-preserving collaborative intrusion detection systems for VANETs, IEEE Trans. Signal Inf. Process. Over Netw. 4 (1) (2018) 148–161, http://dx.doi.org/10.1109/TSIPN.2018.2801622.

[23] O.A. Wahab, A. Mourad, H. Otrok, J. Bentahar, CEAP SVM-based intelligent detection model for clustered vehicular ad hoc networks, Expert Syst. Appl. 50 (2016) 40–54, http://dx.doi.org/10.1016/j.eswa.2015.12.006, URL https://www.sciencedirect.com/science/article/pii/S0957417415008088.

[24] B. Subba, S. Biswas, S. Karmakar, A game theory based multi layered intrusion detection framework for VANET, Future Gener. Comput. Syst. 82 (2018) 12–28, http://dx.doi.org/10.1016/j.future.2017.12.008, URL https://www.sciencedirect.com/science/article/pii/S0167739X17314486.

[25] H. Sedjelmaci, S.M. Senouci, An accurate and efficient collaborative intrusion detection framework to secure vehicular networks, Comput. Electr. Eng. 43 (2015) 33–47, http://dx.doi.org/10.1016/j.compeleceng.2015.02.018, URL https://www.sciencedirect.com/science/article/pii/S004579061500066X.

[26] M. Aloqaily, S. Otoum, I.A. Ridhawi, Y. Jararweh, An intrusion detection system for connected vehicles in smart cities, Ad Hoc Netw. 90 (2019) 101842, http://dx.doi.org/10.1016/j.adhoc.2019.02.001, URL https://www.sciencedirect.com/science/article/pii/S1570870519301131, Recent advances on security and privacy in Intelligent Transportation Systems.

[27] J. Shu, L. Zhou, W. Zhang, X. Du, M. Guizani, Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4519–4530, http://dx.doi.org/10.1109/TITS.2020.3027390.

[28] M. Kim, I. Jang, S. Choo, J. Koo, S. Pack, Collaborative security attack detection in software-defined vehicular networks, in: 2017 19th Asia-Pacific Network Operations and Management Symposium, APNOMS, 2017, pp. 19–24, http://dx.doi.org/10.1109/APNOMS.2017.8094172.

[29] P.H. Mirzaee, M. Shojafar, H. Bagheri, T.H. Chan, H. Cruickshank, R. Tafazolli, A two-layer collaborative vehicle-edge intrusion detection system for vehicular communications, in: 2021 IEEE 94th Vehicular Technology Conference, VTC2021-Fall, 2021, pp. 1–6, http://dx.doi.org/10.1109/VTC2021-Fall52928.2021.9625388.

[30] N.I. Mowla, N.H. Tran, I. Doh, K. Chae, Federated learning-based cognitive detection of jamming attack in flying ad-hoc network, IEEE Access 8 (2020) 4338–4350, http://dx.doi.org/10.1109/ACCESS.2019.2962873.

[31] X. Zhou, W. Liang, J. She, Z. Yan, K.I.-K. Wang, Two-layer federated learning with heterogeneous model aggregation for 6G supported internet of vehicles, IEEE Trans. Veh. Technol. 70 (6) (2021) 5308–5317, http://dx.doi.org/10.1109/TVT.2021.3077893.

[32] S.S. Magdum, A.A. Franklin, T.B. Reddy, D.S. Pawar, SafeNav: A cooperative V2X system using cellular and 802.11p based radios opportunistically for safe navigation, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC, 2020, pp. 1–6.

[33] Y. Xiao, Y. Ye, S. Huang, L. Hao, Z. Ma, M. Xiao, S. Mumtaz, O.A. Dobre, Fully decentralized federated learning-based on-board mission for UAV swarm system, IEEE Commun. Lett. 25 (10) (2021) 3296–3300, http://dx.doi.org/10.1109/LCOMM.2021.3095362.

[34] S. Zhang, Z. Wang, Z. Zhou, Y. Wang, H. Zhang, G. Zhang, H. Ding, S. Mumtaz, M. Guizani, Blockchain and federated deep reinforcement learning based secure cloud-edge-end collaboration in power IoT, IEEE Wirel. Commun. 29 (2) (2022) 84–91, http://dx.doi.org/10.1109/MWC.010.2100491.

[35] Y.J. Cho, J. Wang, G. Joshi, Client selection in federated learning: Convergence analysis and power-of-choice selection strategies, 2021, URL https://openreview.net/forum?id=PYAFKBc8GL4.

[36] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC, IEEE, 2019, http://dx.doi.org/10.1109/icc.2019.8761315.

[37] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, 2018, http://dx.doi.org/10.48550/ARXIV.1812.06127, URL https://arxiv.org/abs/1812.06127.

[38] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H.B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, J. Roselander, Towards federated learning at scale: System design, 2019, http://dx.doi.org/10.48550/ARXIV.1902.01046, URL https://arxiv.org/abs/1902.01046.

[39] C. Xie, S. Koyejo, I. Gupta, Asynchronous federated optimization, 2019, http://dx.doi.org/10.48550/ARXIV.1903.03934, URL https://arxiv.org/abs/1903.03934.

[40] W. Wu, L. He, W. Lin, R. Mao, C. Maple, S. Jarvis, SAFA: A semi-asynchronous protocol for fast federated learning with low overhead, IEEE Trans. Comput. 70 (5) (2021) 655–668, http://dx.doi.org/10.1109/TC.2020.2994391.

[41] S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1205–1221, http://dx.doi.org/10.1109/JSAC.2019.2904348.

[42] P. Singh, M.K. Singh, R. Singh, N. Singh, Federated learning: Challenges, methods, and future directions, in: Federated Learning for IoT Applications, Springer International Publishing, Cham, 2022, pp. 199–214, http://dx.doi.org/10.1007/978-3-030-85559-8_13.

[43] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, MAB-based client selection for federated learning with uncertain resources in mobile networks, 2020, http://dx.doi.org/10.48550/ARXIV.2009.13879, URL https://arxiv.org/abs/2009.13879.

[44] J. Xu, H. Wang, Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective, IEEE Trans. Wireless Commun. 20 (2) (2021) 1188–1200, http://dx.doi.org/10.1109/TWC.2020.3031503.

[45] T. Huang, W. Lin, L. Shen, K. Li, A.Y. Zomaya, Stochastic client selection for federated learning with volatile clients, IEEE Internet Things J. 9 (20) (2022) 20055–20070, http://dx.doi.org/10.1109/JIOT.2022.3172113.

[46] S. Abdulrahman, H. Tout, A. Mourad, C. Talhi, FedMCCS: Multicriteria client selection model for optimal IoT federated learning, IEEE Internet Things J. 8 (6) (2021) 4723–4735, http://dx.doi.org/10.1109/JIOT.2020.3028742.

[47] F. Lai, X. Zhu, H.V. Madhyastha, M. Chowdhury, Oort: Efficient federated learning via guided participant selection, in: USENIX OSDI (2021), 2020, http://dx.doi.org/10.48550/ARXIV.2010.06081, URL https://arxiv.org/abs/2010.06081.

[48] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing federated learning on non-IID data with reinforcement learning, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 1698–1707, http://dx.doi.org/10.1109/INFOCOM41043.2020.9155494.

[49] Y. Wang, B. Kantarci, A novel reputation-aware client selection scheme for federated learning within mobile environments, in: 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, 2020, pp. 1–6, http://dx.doi.org/10.1109/CAMAD50429.2020.9209263.

[50] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, H. Huang, FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing, IEEE J. Sel. Areas Commun. 39 (12) (2021) 3654–3672, http://dx.doi.org/10.1109/JSAC.2021.3118435.

[51] T. Kim, B. Jin, S.-H. Cha, M.-S. Jun, Secure vehicle pseudonym certificate for smart car in internet of vehicles, Int. J. Control Autom. 10 (6) (2017) 35–48, http://dx.doi.org/10.14257/ijca.2017.10.6.05.

[52] J. Kamel, F. Haidar, I.B. Jemaa, A. Kaiser, B. Lonc, P. Urien, A misbehavior authority system for sybil attack detection in C-ITS, in: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2019, 2019, pp. 1117–1123, http://dx.doi.org/10.1109/UEMCON47517.2019.8993045.

[53] C. Sommer, D. Eckhoff, A. Brummer, D.S. Buse, F. Hagenauer, S. Joerer, M. Segata, Veins: The open source vehicular network simulation framework, in: Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem, Springer International Publishing, Cham, 2019, pp. 215–252, http://dx.doi.org/10.1007/978-3-030-12842-5_6.

L. Jai Vinita received a M.E. degree in Computer Science and Engineering from Anna University, Chennai in 2008. She is currently pursuing Ph.D. degree with the College of Engineering Guindy, Anna University, Chennai, India. Her research interest includes Vehicular ad-hoc networks, Internet of Vehicles and Cryptography and Network Security.

Vetriselvi V. received a Ph.D. degree in Computer Science and Engineering from Anna University, Chennai in 2008. She is currently working as a professor at the Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai. Her major area of research is Wireless Networks, Vehicular ad-hoc networks, Internet of Vehicles, Cryptography and Network Security.