Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.084062

ARTICLE

Robust Federated Learning for Intrusion Detection in Autonomous Vehicles against Poisoning Attacks

Ulysses Lam[1,*], Jin-Hee Cho[2], Hyuk Lim[3], Terrence Moore[4], Frederica Free-Nelson[4], Hyunjae Kang[1] and Dan Dongseong Kim[1]

[1]School of Electrical Engineering and Computer Science, the University of Queensland, Brisbane, QLD, Australia
[2]Department of Computer Science, Virginia Tech, Falls Church, VA, USA
[3]School of Energy Engineering, Korea Institute of Energy Technology, Naju-Si, Republic of Korea
[4]US DEVCOM Army Research Laboratory, Adelphi, MD, USA

*Corresponding Author: Ulysses Lam. Email: u.lam@uq.edu.au

Received: 15 April 2026; Accepted: 04 June 2026; Published: 30 June 2026

ABSTRACT: Autonomous vehicles are potentially more vulnerable to cyber-attacks compared to traditional human-driven ones, as they employ electronic sensors to enable self-driving. Cybersecurity for autonomous vehicles will be crucial in the near future. However, intrusion detection systems (IDSes) for vehicles are still in the early stages. Many IDS models that claim to work for vehicles are actually built with traditional Internet datasets rather than those with real vehicle data, which is impractical in reality. In this paper, IDS models are developed with Federated Learning (FL) with the Car-Hacking and CAN-MIRGU datasets, which are obtained from real vehicles. The FL-based IDS models achieve high attack-detection performance, while each local client retains their privacy by sharing only local model weights rather than local datasets. Training of local models takes an extremely short time and is feasible in practice for vehicles with low computational resources. Furthermore, different poisoning scenarios are performed on local clients to demonstrate the high robustness of FL models. The FL-based IDS models are highly robust against poisoning attacks and maintain high detection accuracy as long as the majority of local clients are not compromised.

KEYWORDS: Federated learning; intrusion detection; Controller Area Network (CAN bus); Vehicular Ad-hoc Network (VANET); poisoning attack

1 Introduction

In recent years, the number of autonomous vehicles has grown rapidly around the world. In countries such as the United States and China, self-driving taxis have become a revolutionary means of transportation for passengers in major cities. Although autonomous vehicles have brought great convenience to transportation, statistical reports show that the accident rates of autonomous vehicles are generally higher than those of human-driven cars. In addition, since autonomous vehicles make driving decisions based on information obtained from various electronic devices, such as the Global Positioning System (GPS), Light Detection and Ranging (LiDAR), cameras, and radars, they can be more vulnerable to cyber-attacks compared to traditional human-driven vehicles. Therefore, as autonomous vehicles become the future trend of transportation, their safety issues will become the main concern of all road users at the same time.

To protect autonomous vehicles from cyber-attacks, a potential solution is to develop new Intrusion Detection Systems (IDSes) for vehicles. Firstly, IDS models should be built with datasets obtained from real

vehicle data so that they are more practical for detecting attacks in vehicular networks. Moreover, since autonomous vehicles have very limited computational resources, which differ from those of computers, Federated Learning (FL) can be an ideal solution, as local clients share the training process with smaller amounts of data. FL has been studied in various domains, but its effectiveness and robustness for vehicular intrusion detection remain insufficiently explored. Lastly, because IDS performance can be affected if local clients are compromised by poisoning attacks, the robustness of FL models is also an important measure.

This study evaluates the feasibility of deploying FL-based IDSes on Vehicular Ad-hoc Networks (VANET) with two contributions:

1. To develop reliable, high-detection IDS models by processing real-world vehicle datasets.
2. To test the robustness of FL models in different scenarios involving compromised clients and contaminated local data.

Compared to the preliminary version [1] (A preliminary version of this work was presented at MobiSec 2025), this paper extends the work by providing additional experiments on the Median aggregation algorithm.

2 Background

2.1 Vehicular Ad-Hoc Network

VANETs are decentralised communication networks that connect vehicles with each other and with roadside infrastructure. By enabling continuous data exchange across a large number of vehicles, VANET provides a rich source of traffic and security-related information that can be leveraged for collaborative learning [2]. This distributed nature makes VANET particularly suitable for FL, where vehicles act as clients that train local intrusion detection models on their own data and share only model updates with a central server. Such an approach preserves data privacy, reduces communication overhead, and allows intrusion detection systems to benefit from diverse and dynamic vehicular data without requiring direct data sharing.

In this context, VANET can be realised through two communication scenarios: Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V), and this paper focuses only on the V2I scenario as shown in Fig. 1. The V2I scenario consists of a central server, multiple Roadside Units (RSUs) and multiple vehicles. It enables communication between vehicles and RSUs, and subsequently between RSUs and the central server. Therefore, vehicles can share data with the central server and vice versa. V2I is often used for managing live traffic and sending emergency messages.

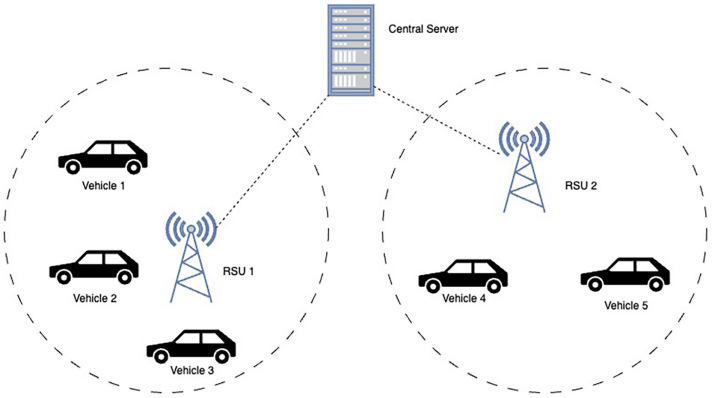

Figure 1: Vehicle-to-infrastructure in VANET.

2.2 Federated Learning

FL is a decentralised machine learning technique that enables multiple clients to collaboratively train a shared global model without exchanging raw data. Each client trains the model locally on its own dataset and transmits only the model updates to a central server, where aggregation is performed to refine the global model. This approach enhances data privacy, reduces communication costs, and allows learning from diverse, decentralised data sources [3].

Given these properties, integrating FL with VANET offers several advantages. First, since clients upload only model updates rather than raw data collected from vehicle sensors, FL enables vehicles to contribute to training without exposing sensitive information such as location (GPS data) or driving behaviour. Second, by avoiding direct data transmission, FL also reduces communication overhead between vehicles and the central server. Third, FL supports vehicular edge computing by leveraging the computational resources of edge nodes, thereby alleviating the training burden on the central server that would otherwise need to process data from numerous vehicles [4].

However, security issues have emerged due to FL's distributed and decentralised nature. One major threat is poisoning attacks [5,6], in which malicious participants corrupt the global model either by manipulating local training data (data poisoning) or by directly altering model updates (model poisoning). Data poisoning, such as label flipping, can bias the model toward targeted misclassifications, while model poisoning injects adversarial gradients to degrade overall performance. These threats highlight the need for further research on the impact of poisoning attacks, particularly in safety-critical domains such as vehicular intrusion detection.

3 Related Work

Recent studies have shown growing interest in applying FL to intrusion detection in vehicular networks. Chen et al. [7] proposed a VANET intrusion detection system leveraging federated learning (VAN-FED-IDS) for VANETs with CIC-IDS-2017 [8] datasets, where RSUs serve as FL clients to aggregate models trained with packet- and physics-based features, demonstrating that FL can preserve privacy while maintaining detection accuracy. Gurjar et al. [9] developed an FL-based misbehaviour classification system for VANETs using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) variants under different aggregation strategies, highlighting FL's scalability in dynamic traffic environments with the VeReMi extension dataset. Huang et al. [10] introduced FED-IoV for the Internet of Vehicles (IoV), applying MobileNet-Tiny within an FL framework and validating its effectiveness on both in-vehicle network(Car-Hacking [11]) and computer network (CIC-IDS-2017) intrusion datasets. Similarly, Althunayyan et al. [12] applied hierarchical FL to IoV using the Car-Hacking dataset, while Bhavsar et al. [13] proposed FL-IDS with NSL-KDD [14] and Car-Hacking datasets, which leverages vehicular edge devices in transportation Internet of Things (IoT) to reduce communication costs and distribute computation. Collectively, these works establish FL as a promising paradigm for intrusion detection in vehicular networks.

Beyond accuracy and efficiency, a few studies have addressed the security of FL in vehicular contexts. Abou El Houda et al. [15] integrated FL with blockchain to enhance the reliability and trustworthiness of FL in VANETs using the UNSW-NB15 dataset [16], while Ullah et al. [17] proposed SPBFL-IoV, a secure and privacy-preserving blockchain-based framework incorporating homomorphic encryption and filtering to mitigate poisoning attacks in general IoV environments with MNIST and CIFAR-10.

Table 1 shows a comparison of related work and the IDS models in this paper. Some of the IDS models were not built with vehicular datasets. Gurjar et al. [9] used the VeReMi extension dataset which was collected from simulators rather than a real vehicle. Other models were developed with both vehicular and computer

network datasets, such as Huang et al. [10] and Bhavsar et al. [13]. Finally, only Althunayyan et al. [12] has built their IDS with only real vehicular datasets. Conversely, the IDS models in this paper are built and tested with two real vehicular datasets: the Car-Hacking and CAN-MIRGU [18] datasets. Although these studies advanced privacy and robustness in FL, none specifically investigated poisoning attacks in the context of intrusion detection for vehicular networks using FL. As current research indicates no prior work in this area, this paper is the first to address this gap by systematically evaluating poisoning attacks against FL-based intrusion detection in vehicular environments.

Table 1: Comparison of related work and this paper.

Work	Dataset	Vehicular Data	Poisoning Attack
Chen et al. [7]	CIC-IDS-2017		
Gurjar et al. [9]	VeReMi extension	Simulation	
Huang et al. [10]	Car-Hacking & CIC-IDS-2017	Partly	
Althunayyan et al. [12]	Car-Hacking	✓	
Bhavsar et al. [13]	NSL-KDD & Car-Hacking	Partly	
Abou El Houda et al. [15]	UNSW-NB15		
Ullah et al. [17]	MNIST & CIFAR-10		
This paper	Car-Hacking & CAN-MIRGU	✓	✓

4 Data Processing

The Car-Hacking and CAN-MIRGU datasets are selected to build FL-based IDS models and compare their performance. The procedures of data processing are shown in Fig. 2. They are multi-class, with each attack type saved in an individual file. To simplify the detection, the data are converted into binary classes: attack and benign. Both datasets consist of Controller Area Network (CAN) bus messages, the de facto communication protocol for in-vehicle networks. A simple version of CAN message structure is shown in Table 2, including 3 major fields: CAN ID, Data Length Code (DLC) and Data. The CAN ID is a message identifier, where a lower ID has a higher priority in case of concurrent messages are sent through the CAN bus. The DLC indicates the number of bytes of data, ranging from 0 to 8. The data is the actual payload to be transmitted. In reality, a CAN message also contains other fields such as start of Frame (SOF), Cyclic Redundancy Code (CRC), Acknowledgment (ACK), *etc*. However, most vehicular datasets contain only the fields shown in Table 2.

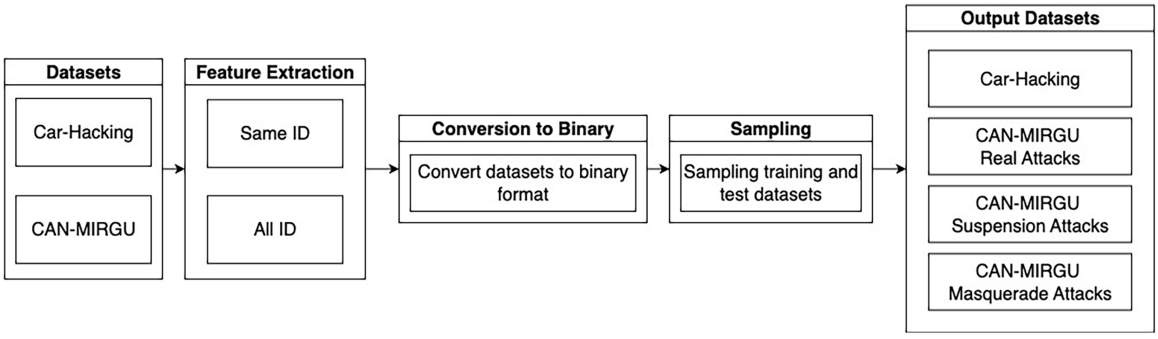

Figure 2: Data processing of datasets.

Table 2: Simple CAN message structure.

CAN ID	DLC	Data
11 bits	4 bits	0 to 64 bits

The Car-Hacking dataset [11] is published with real vehicle data by the Hacking and Countermeasure Research Lab (HCRL). It contains four attack types: Denial-of-Service (DoS) Attack, Fuzzy Attack, Spoofing the drive gear, and Spoofing the Revolutions Per Minute (RPM) gauge, as well as attack-free data. The data were collected from a Hyundai YF Sonata. The CAN messages were captured from the OBD-II port by a Raspberry Pi3 while attacks were injected. The dataset is unbalanced with numbers of benign and attack samples summarised in Table 3.

Table 3: Car-hacking dataset summary.

Attack Type	Num of Benign Messages	Num of Attacks	Benign to Attack Ratio
DoS Attack	3,078,250	587,521	5:1
Fuzzy Attack	3,347,013	491,847	7:1
Spoofing the drive gear	3,845,890	597,252	6:1
Spoofing the RPM gauge	3,966,805	654,897	6:1

The CAN-MIRGU dataset, published in 2024, comprises 36 attack types divided into three main groups: real attacks, suspension attacks, and masquerade attacks. The data were collected from an electric vehicle with full autonomous driving capabilities, but the make and model were not disclosed. A Kvaser Memorator 2xHS v2 was connected directly to the CAN gateway for data logging. Compared with the Car-Hacking dataset, it contains more attack types such as replay, suspension and masquerade attacks. Due to safety concerns, the data collection and attack injections of other datasets are performed while the vehicle is stationary or driving on a dynamometer. In the CAN-MIRGU dataset, the benign datasets were collected while the vehicle was driving on public roads, which mimics real-world driving cases. Conversely, the attacks were injected while the vehicle was driven on a proving ground with a speed limit of 30 mph. It is the most comprehensive dataset covering the most attack types, and the data collection environment is the most similar to normal road driving behaviour. There is no IDS built with the CAN-MIRGU dataset that has been published yet, as it is relatively new.

4.1 Feature Extraction

Since the datasets contain only basic information about CAN bus messages, e.g., timestamp, CAN ID and data, the number of messages are counted in three time windows: 1, 0.5 and 0.25 s, based on the same CAN IDs and all CAN IDs. Therefore, six features are extracted from the data. Lastly, the extracted features are normalised with Min-Max Normalisation.

4.2 Conversion to Binary

The six normalised features, as well as CAN ID and data, are converted to binary format. Since the features are normalised, their values are multiplied by 1000, then the integer parts are converted to binary. These eight columns of data will be the input data for training FL models.

4.3 Sampling

Since the datasets have tens of millions of records, it is infeasible to include all the data to build FL models. Therefore, sampling a small amount of data is necessary before developing the IDS models. Both datasets contain two types of sub-datasets: attack datasets and benign datasets. The attack datasets contain both benign data and one type of attack data, while the benign datasets contain only benign data. In each attack dataset, the number of benign data is already overwhelming compared to the number of attack data. Therefore, benign datasets are excluded from data processing to make the sampled datasets more balanced. Moreover, to train FL models, each client should have their own local training dataset, so the training dataset is randomly divided into 100 sub-datasets for 100 clients in the FL model. Lastly, each sampled dataset contains a test dataset and 100 local training datasets.

4.3.1 Car-Hacking Dataset

The numbers of the four attack types are generally balanced, but benign data are six times more than attack data. Before sampling, 50% of the benign data are removed randomly. The numbers of records are shown in Table 4.

Table 4: Summary of sampled datasets.

Dataset	Sampled Dataset	Total Records	Attack Records	Benign Records
Car-Hacking	Training	400,000	98,997	301,003
	Test	40,000	9816	30,184
CAN-MIRGU (Real Attacks)	Training	400,000	131,938	268,062
	Test	40,000	14,674	25,326
CAN-MIRGU (Suspension Attacks)	Training	400,000	131,287	268,713
	Test	40,000	13,173	26,827
CAN-MIRGU (Masquerade Attacks)	Training	88,500	22,730	65,770
	Test	9761	2453	7308

4.3.2 CAN-MIRGU Dataset

The CAN-MIRGU dataset comprises 36 attack types, with the number of records for each type varying from around one hundred to hundreds of thousands. To make the data more balanced and less complicated, the CAN-MIRGU dataset is divided into three sub-datasets based on the three main groups: real attacks, suspension attacks, and masquerade attacks.

4.3.3 Real Attacks

Real attacks in CAN-MIRGU contain 26 attack types with the number of records ranging from 118 to 40,759, which is extremely unbalanced. Therefore, the numbers of attack records are sampled using Algorithm 1. Attack types with more records are sampled with a smaller proportion so that their numbers are more balanced. In addition, the ratio of benign to attack records is nearly 65:1, so only one-thirtieth of the benign data is randomly chosen before sampling training and test datasets. The numbers of records are shown in Table 4.

Algorithm 1: Sampling attack records for real attacks

Number of Attacks: N;
if $N \geq 30,000$ **then**
  $N = N \cdot 0.25$
else
  if $N \geq 10,000$ **then**
    $N = N \cdot 0.7$
  end if
end if

4.3.4 Suspension Attacks

Suspension attacks in CAN-MIRGU comprise 5 attack types, with approximately 20,000 records for each type, and the ratio between benign and attack records is approximately 2:1. The training and test datasets are sampled directly, as they are generally balanced, as shown in Table 4.

4.3.5 Masquerade Attacks

Masquerade attacks in CAN-MIRGU contain 5 attack types with numbers of records ranging from 118 to 10,936, which is also unbalanced. The ratio of benign to attack records is approximately 87:1. Since the total number of attack records is significantly smaller than that of other datasets, all attacks are included in the sampled dataset, while only one-thirtieth of benign data is selected, as shown in Table 4.

5 Federated Learning Model

The FL model usually consists of a central server and multiple fixed clients, with each client possessing its own local training datasets. However, in VANET, clients are equivalent to vehicles, and they are moving all the time. Therefore, the clients connected to the central server are supposed to change from time to time. To simulate practical cases, the FL models consist of 100 clients with individual local training datasets, and 10 clients are randomly selected in each round of training. Clients can train their models locally with their own datasets, then upload model weights to the central server rather than share any local data [19].

The phases of one round of training are as shown in Fig. 3:

1. **Server-to-Client Broadcast:** The central server updates all clients with the weight of the global model.
2. **Local Clients Update:** Each selected client trains their local model with local training datasets.
3. **Client-to-Server Upload:** Selected clients upload weights of local models to the central server.
4. **Server Update:** The central server updates the global model by aggregating local model weights with an aggregation algorithm.

The training process of FL models involves five rounds of these phases. In the **Local Clients Update** phase, local models are built as Convolutional Neural Network (CNN) [20] for binary classification. In the **Server Update** phase, two aggregation algorithms are applied: Federated Averaging (FedAvg) [21] and Median [22]. FedAvg is a simple aggregation algorithm to obtain a new global model by calculating the average values of local model weights. Conversely, the Median aggregation is a Byzantine-robust method to eliminate malicious clients by choosing the median value among all local model weights.

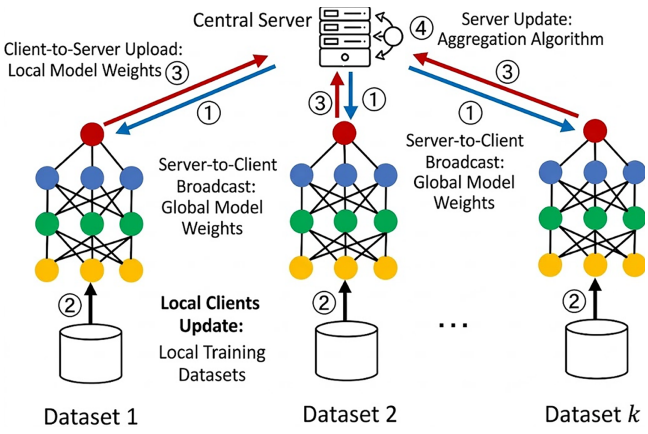

Figure 3: Federated learning processes.

6 FedAvg Aggregation

In FedAvg aggregation, the global model weight is obtained as the mean of selected local model weights. The FedAvg process specified for this study is shown in Algorithm 2.

Algorithm 2: FedAvg process

for each round r = 1, 2, . . ., 5 **do**
1. Server-to-Client Broadcast:
if r = 1 **then**
Central server initialises global weight $\mathbf{G}_1$
end if
for each client $n \in N$, $|N| = 100$ **do**
Local model weight $\mathbf{L}_r^n \leftarrow \mathbf{G}_r$
end for
2. Local Clients Update:
for each randomly selected clients $k \in K$, $|K| = 10$, $K \subset N$ **do**
Clients train local models with local datasets, obtaining new trained local weights $\mathbf{T}_r^k$
$\mathbf{L}_r^n \leftarrow \mathbf{T}_r^k$ **if** $n = k$
end for
3. Client-to-Server Upload:
for $k \in K$ **do**
Selected clients upload $\mathbf{L}_r^k$ to the central server
end for
4. Server Update:
Central server updates a new global model weight
$\mathbf{G}_{r+1} = \frac{\sum_{k \in K} \mathbf{L}_r^k}{|K|}$
end for

6.1 Simulation Cases

In addition to benign datasets, different FL models are also trained by changing the number of compromised clients and the percentage of data contamination, so that the robustness of FL models is tested. To contaminate local training datasets, a certain percentage of labels are randomly flipped each time, so attacks are labelled as benign and vice versa. The numbers of compromised clients are set to be 20, 40, 60, 80

and 100, while the percentages of flipped labels in their local training datasets are set to be 20%, 40%, 60%, 80% and 100%. Hence, there are a total of 26 cases: 25 cases with abnormal data and 1 case with normal data.

Multiple FL models are built in the 26 cases, and for each case, the model is trained and tested 100 times. In this section, the performance and robustness of FL-based IDS models built with different datasets are tested and compared.

6.2 Car-Hacking Dataset Results

For the Car-Hacking dataset, the FL model achieves an accuracy of over 96% when no client is compromised, as shown in Fig. 4. When 20 out of 100 clients are compromised, the accuracy is hardly affected. Similarly, when only 20% of the labels are flipped, the accuracy is maintained above 96%. The model has high robustness, so it achieves accuracy above 80% when at least 60% of the clients or data are normal. The accuracy starts to drop rapidly as the majority of clients are compromised, with a higher percentage of data labels being flipped.

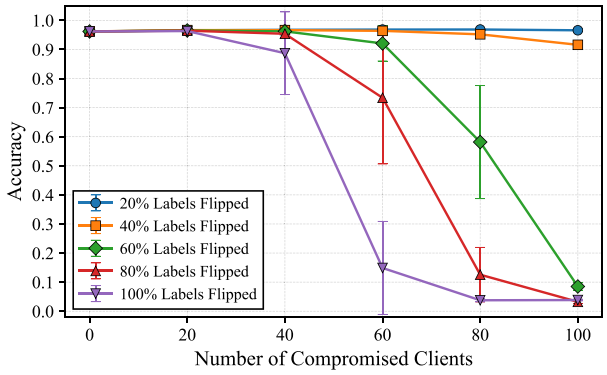

Figure 4: Accuracy vs. compromised clients and flipped labels for car-hacking with FedAvg.

Compare the following two cases:

1. 60 clients are compromised, and 80% of labels are flipped
2. 80 clients are compromised, and 60% of labels are flipped

Both cases indicate that 48% of the overall data is contaminated among all clients. However, Case 2 has an obviously higher accuracy than that of Case 1, which is 73.4% and 58.2%, respectively. Therefore, increasing the number of compromised clients may have a bigger impact on the robustness of models than increasing the percentage of flipped labels, given that the overall percentage of contaminated data is equal.

The graphs have shown an abnormal trend, in which the accuracy increases slightly as the number of clients increases. However, the increment in accuracy is less than 0.7%, which can be neglected. It can be a minor fluctuation because the clients are reselected in each round of training.

Since the datasets are unbalanced, it is also necessary to consider the F1-score of the model, as it is more reliable in balancing the performance of precision and recall [23]. Since the FL models are trained and tested 100 times for each case, with different selected clients in each training round, the average F1-score should be calculated. A more accurate way to obtain the average F1-score is to calculate it with the total numbers of true positives (TP), false positives (FP) and false negatives (FN). However, since the values of TP, FP, and FN are not available, the average F1-score is calculated with the average values of precision and recall:

$$F1 = \frac{2 \cdot (precision \cdot recall)}{(precision + recall)}$$

Since the datasets have positive classes of more than 10%, which is not highly unbalanced, the bias of the calculated F1-score is minor [24].

The FL model also has a high F1-score greater than 0.91, as shown in Fig. 5. The F1-score is similar to the accuracy, and it remains high when the majority of clients or data are not compromised or contaminated.

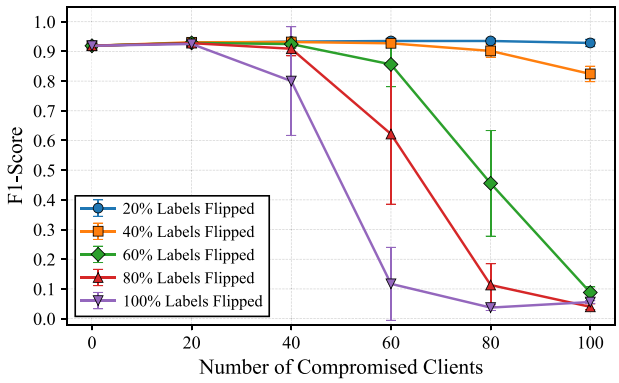

Figure 5: F1-score vs. compromised clients and flipped labels for car-hacking with FedAvg.

6.3 CAN-MIRGU Dataset Results

For the CAN-MIRGU dataset, the FL models have also demonstrated high robustness to compromised clients and contaminated data, as shown in Figs. 6 and 7. Both the accuracy and the F1-score decrease slightly when less than half of the clients and local data are affected by flipped labels. The FL models for both real attacks and masquerade attacks achieve very high performance, comparable to that of the Car-Hacking dataset. However, the accuracy and F1-score are significantly lower for suspension attacks.

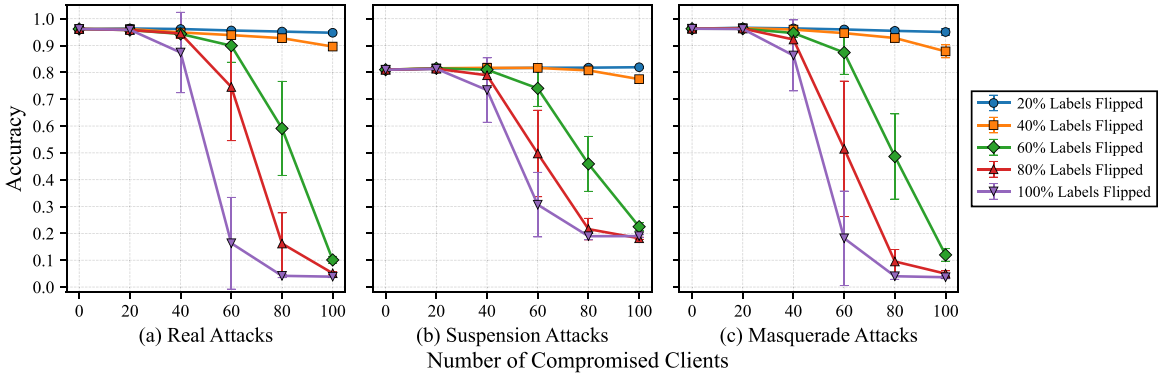

Figure 6: Accuracy vs. compromised clients and flipped labels for CAN-MIRGU with FedAvg.

The FL model performs poorly for suspension attacks due to its low recall, as shown in Fig. 8. The recall values are always around 0.5, which indicates that around 50% of attacks are correctly classified. In addition, contaminated data can barely affect recall values, even though all 100 clients are compromised with 100% of labels flipped. Since suspension attacks are launched by compromising Electronic Control Units (ECUs) for a period and preventing their messages from being sent, which does not involve attack injection, the attack messages may contain the same CAN IDs and data as the benign ones, only with different timestamps and labels. Therefore, flipping the labels does not affect the local training datasets. Moreover, since suspension attacks do not inject additional attack frames, unlike real attacks, which inject attacks in a fixed time interval,

the extracted features of message counts cannot provide useful information for attack detection. Hence, the FL model has low detection rates for suspension attacks.

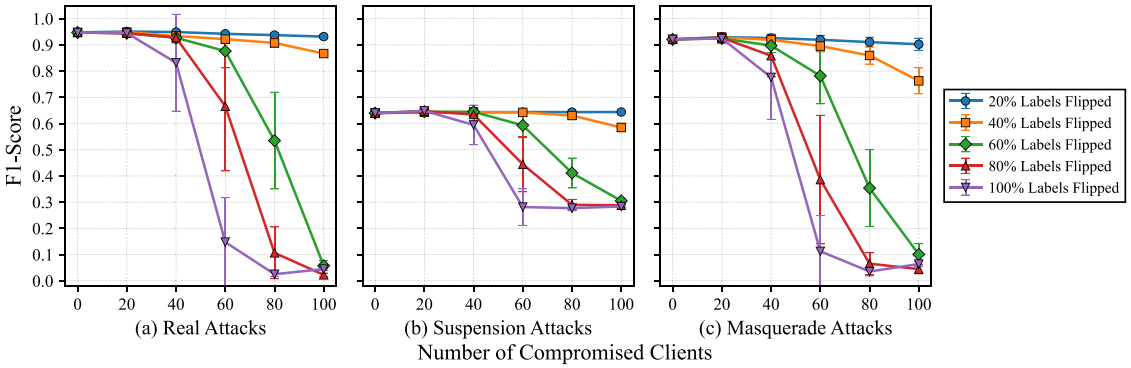

Figure 7: F1-score vs. compromised clients and flipped labels for CAN-MIRGU with FedAvg.

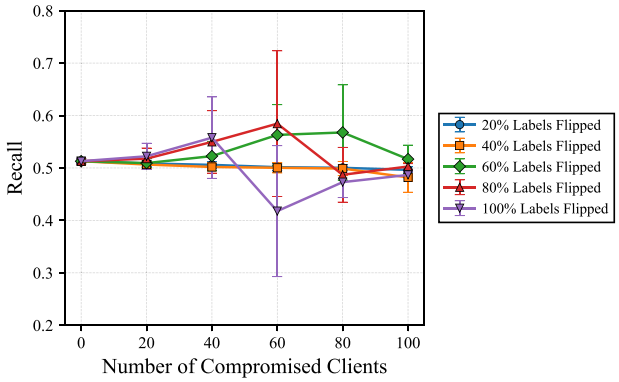

Figure 8: Recall vs. compromised clients and flipped labels for suspension attacks with FedAvg.

7 Median Aggregation

In Median aggregation, the global model weight is calculated as the median value of selected local model weights. In this study, since 10 clients are selected in each round, the Median aggregation is equivalent to eliminating the 4 largest and 4 smallest local weights before processing FedAvg.

7.1 Simulation Cases

Firstly, since the Median aggregation obtains the median value of local model weights, it is not supposed to work properly when the majority of clients are compromised [25]. Therefore, unlike the cases of FedAvg, the percentages of flipped labels are only up to 60%. Hence, there are a total of 16 cases: 15 cases with abnormal data and 1 case with normal data. Secondly, for the CAN-MIRGU dataset, suspension attacks are not included in the test since the flipping label attack does not have an obvious effect on the data, as discussed in the previous results.

7.2 Car-Hacking Dataset Results

The flipping label attacks have a minor impact on the FL model when only 20% of clients are compromised, regardless of the percentage of flipped labels, as shown in Fig. 9. As the percentage reaches 60%, the accuracy is even lower with 60% or 80% compromised clients, compared to FedAvg as shown

in Fig. 4. The Median aggregation selects the median value among local model weights. Therefore, it is calculated as the average value of the 5th and 6th local weights out of 10 clients. Since more than 50% of clients are compromised, there is a very high chance that at least one of the chosen local models is compromised. Therefore, the global model is affected by the chosen compromised model, and thus, the accuracy is reduced. When 40% of clients are compromised with over 60% flipped labels, the accuracy is slightly lower than FedAvg. Theoretically, Median aggregation can eliminate malicious clients as long as they are less than 50%. The reason for this result is explained in 7.4.

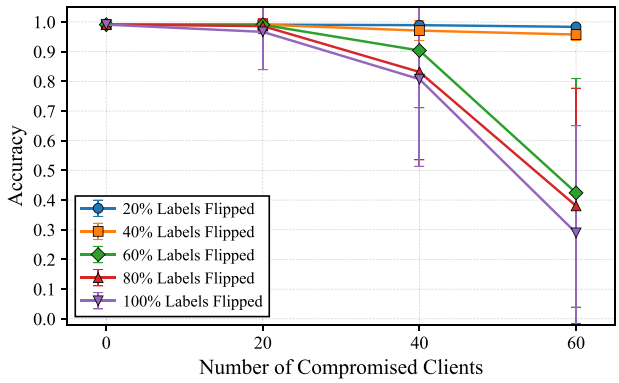

Figure 9: Accuracy vs. compromised clients and flipped labels for car-hacking with median.

The FL model achieves a high F1-score when no more than 20% clients are compromised, as shown in Fig. 10. However, the performance is not as good as FedAvg when more than 40% clients are compromised, which is similar to the accuracy in Fig. 9.

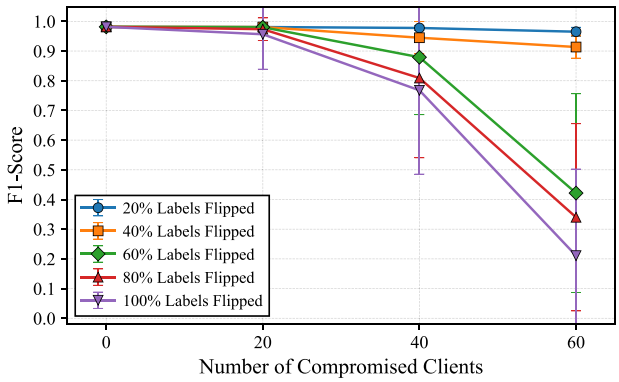

Figure 10: F1-score vs. compromised clients and flipped labels for car-hacking with median.

7.3 CAN-MIRGU Dataset Results

The accuracy and F1-scores are shown in Figs. 11 and 12. Both have similar performance to the results of the Car-Hacking dataset. In comparison with FedAvg in Figs. 6 and 7, the FL model with Median aggregation has slightly lower accuracy and F1-scores with 40% malicious clients. When 60% of clients are compromised, these values are even much lower as malicious clients are selected to update the global model. The impact of no more than 40% of flipped labels is minor, which aligns with the results of FedAvg.

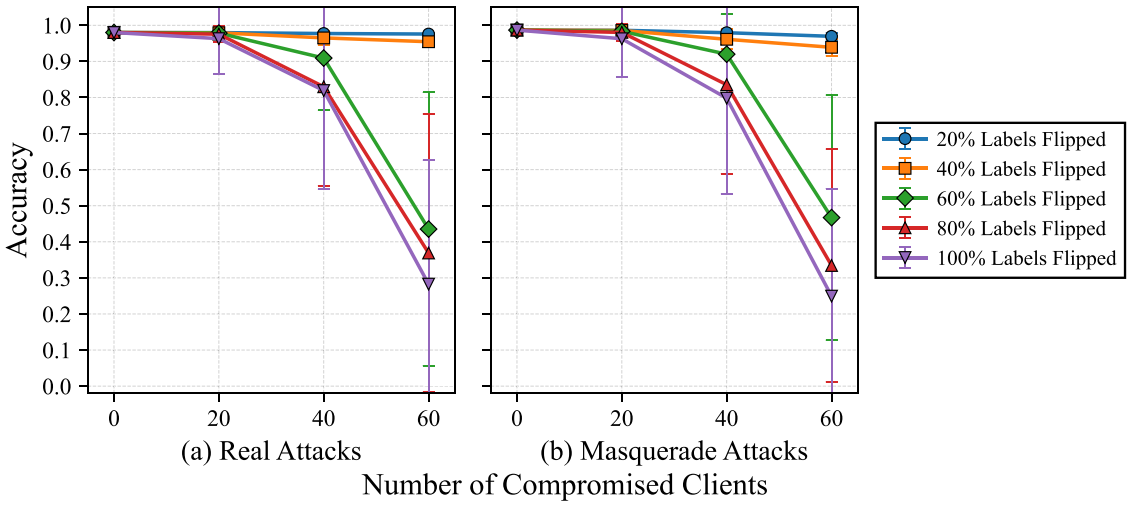

Figure 11: Accuracy vs. compromised clients and flipped labels for CAN-MIRGU with median.

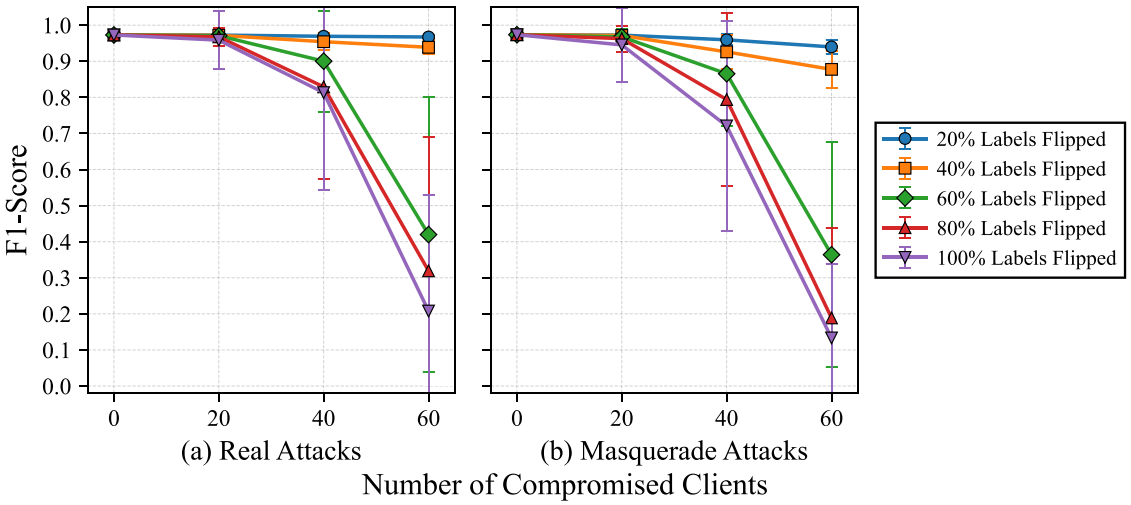

Figure 12: F1-score vs. compromised clients and flipped labels for CAN-MIRGU with median.

7.4 Selected Clients in Median Aggregation

Although the Median aggregation is Byzantine-robust, both datasets have shown the result that the FL model performance is lower than FedAvg with 40% malicious clients. Therefore, the selected clients during the training process should be analysed. Even though 40% of the total 100 clients are compromised, when 10 random clients are chosen in each round of training, the chance for having more than 5 malicious clients is still very high. In addition, since each training process has 5 rounds of training, if the 5th round just happens to select 1 or even 2 malicious clients in the aggregation, the global model weight will be greatly affected. By analysing selected clients in the 5th rounds with 40% malicious clients and 80% flipped labels, the results show that 16 out of 100 times of training processes have chosen 2 malicious clients in the 5th round, as shown in Table 5. Although 40% overall are compromised, it is still common to have 6 or more compromised clients out of 10 random selections. In this table, the accuracy values are much lower than those of benign local models, which are over 90% as shown in the above figures when no clients are compromised. Therefore, the average accuracy is slightly reduced by these training results with 2 compromised clients.

Table 5: Training processes with 2 selected compromised clients in the 5th round.

Training	Total Compromised	Selected Compromised	Accuracy
1	6	2	0.294800
3	6	2	0.104575
7	7	2	0.443025
9	9	2	0.243300
13	6	2	0.380025
18	7	2	0.199025
31	6	2	0.089150
42	7	2	0.228075
46	6	2	0.384775
49	7	2	0.115625
62	8	2	0.139650
77	7	2	0.036150
81	6	2	0.331950
83	7	2	0.328850
88	7	2	0.142725
93	6	2	0.098150

8 Operating Environment and Training Time

The simulations were carried out in VS Code sessions with 24 CPU cores and no GPU on Bunya-a supercomputer at the University of Queensland. In this study, each local client possesses a small amount of local training data, e.g., 885 records for Masquerade Attacks and 4000 records for other attacks. Therefore, a GPU is unnecessary as training a single local model takes only about 1 s. However, since there are multiple datasets and various simulation cases, and each case is tested for 5 rounds 100 times, the total training time through the study is remarkably long. The training times for FedAvg and Median aggregations are shown in Tables 6 and 7, respectively. The training times of Car-Hacking, Real Attacks and Suspension Attacks are basically the same since they have the same sizes of training sets and test sets, as shown in Table 4. Conversely, the training times for Masquerade Attacks are shorter because of smaller datasets. There is no significant difference in training times between FedAvg and Median aggregation algorithms. Notably, each local model is trained sequentially in the simulations as if by a single computer rather than by multiple local clients. However, in practice, local model training should be performed in parallel by different local clients, so the training time is expected to be much shorter.

Table 6: Training times of simulation cases for FedAvg aggregation.

Dataset	Simulation Cases	Time per Case	Total Time
Car-Hacking	26	206 min	5356 min
CAN-MIRGU (Real Attacks)	26	206 min	5356 min
CAN-MIRGU (Suspension Attacks)	26	206 min	5356 min
CAN-MIRGU (Masquerade Attacks)	26	76 min	1976 min

Table 7: Training times of simulation cases for median aggregation.

Dataset	Simulation Cases	Time per Case	Total Time
Car-Hacking	16	206 min	3296 min
CAN-MIRGU (Real Attacks)	16	206 min	3296 min
CAN-MIRGU (Masquerade Attacks)	16	116 min	1216 min

9 Conclusion

In this paper, a new study on IDS for autonomous vehicles has been carried out with two novel contributions: FL models built with real vehicular datasets and the robustness of FL models against poisoning attacks. Firstly, new FL-based IDS models are developed for autonomous vehicles with real datasets containing CAN bus messages. In particular, there is no prior work on IDS models built with the new and comprehensive CAN-MIRGU datasets. The IDS models show the advantages of decentralisation, that local clients with limited computational resources can still detect intrusions with high accuracy by sharing the training process with a small amount of local data. The FL models achieve high performance in detecting most attack types, except for suspension attacks, which have distinct characteristics. Secondly, the FL models have also shown strong robustness against data contamination. The models can maintain relatively high performance as long as more than half of the clients or local datasets are not under poisoning attacks. However, this study has included only attack types from the Car-Hacking and CAN-MIRGU datasets, but the performance of FL models for detecting novel attacks is unknown. In reality, attackers can also perform evasion attacks, such as varying time intervals of attack injections, to trick IDS models. Therefore, extracted features of message counts will not provide meaningful information for intrusion detection. For future work, the FL models will be further improved to detect more attack types, including suspension attacks. One potential solution is to extract more meaningful features from the limited information contained in CAN bus messages. To address evasion attacks, defence strategies such as adversarial training should also be included in developing the FL IDS models. In addition, the models will achieve higher robustness against poisoning attacks by using an advanced aggregation algorithm.

Acknowledgement: The authors acknowledge the computational resources provided by The University of Queensland Research Computing Centre's Bunya supercomputer (https://dx.doi.org/10.48610/wf6c-qy55).

Funding Statement: This material is partly supported by the International Technology Center Indo-Pacific (ITC IPAC) under Contract No. FA520924C0027 and Army Research Office grant No. W911NF-24-2-0241.

Author Contributions: Conceptualisation: Ulysses Lam and Dan Dongseong Kim; formal analysis: Ulysses Lam; project administration: Hyuk Lim; funding acquisition: Terrence Moore and Frederica Free-Nelson; investigation: Terrence Moore and Frederica Free-Nelson; methodology: Jin-Hee Cho; software: Ulysses Lam; supervision: Jin-Hee Cho, Hyuk Lim and Dan Dongseong Kim; visualisation: Hyunjae Kang; writing–original draft: Ulysses Lam and Hyunjae Kang; writing—review & editing: Ulysses Lam. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The Car-Hacking dataset is publicly available in https://ocslab.hksecurity.net/Datasets/car-hacking-dataset. The CAN-MIRGU dataset is publicly available in https://archive.ics.uci.edu/dataset/1035/can-mirgu.

Ethics Approval: Not applicable. This study did not involve human or animal subjects.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lam U, Cho JH, Lim H, Moore T, Free-Nelson F, Kang H, et al. Federated learning-based intrusion detection system for internet of autonomous vehicles against poisoning attacks. In: Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec 2025); 2025 Dec 16–18; Sapporo, Japan. p. 1–12.
2. Karabulut MA, Shah AFMS, Ilhan H, Pathan ASK, Atiquzzaman M. Inspecting VANET with various critical aspects—a systematic review. Ad Hoc Netw. 2023;150(2):103281. doi:10.1016/j.adhoc.2023.103281.
3. Hallaji E, Razavi-Far R, Saif M, Wang B, Yang Q. Decentralized federated learning: a survey on security and privacy. IEEE Trans Big Data. 2024;10(2):194–213.
4. Zhang X, Liu J, Hu T, Chang Z, Zhang Y, Min G. Federated learning-assisted vehicular edge computing: architecture and research directions. IEEE Veh Technol Mag. 2023;18(4):75–84. doi:10.1109/mvt.2023.3297793.
5. Tolpegin V, Truex S, Gursoy ME, Liu L. Data poisoning attacks against federated learning systems. In: Chen L, Li N, Liang K, Schneider S, editors. Computer security—ESORICS 2020. Berlin/Heidelberg, Germany: Springer; 2020. p. 480–501.
6. Fang M, Cao X, Jia J, Gong N. Local model poisoning attacks to Byzantine-Robust federated learning. In: Proceedings of the 29th USENIX Security Symposium (USENIX Security 20); 2020 Aug 12–14; Boston, MA, USA. p. 1605–22.
7. Chen X, Qiu W, Chen L, Ma Y, Ma J. Fast and practical intrusion detection system based on federated learning for VANET. Comput Secur. 2024;142(8):103881. doi:10.1016/j.cose.2024.103881.
8. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018); 2018 Jan 22–24; Madeira, Portugal. p. 108–16.
9. Gurjar D, Grover J, Kheterpal V, Vasilakos A. Federated learning-based misbehavior classification system for VANET intrusion detection. J Intell Inf Syst. 2025;63(3):807–30. doi:10.1007/s10844-025-00920-0.
10. Huang K, Xian R, Xian M, Wang H, Ni L. A comprehensive intrusion detection method for the internet of vehicles based on federated learning architecture. Comput Secur. 2024;147(1):104067. doi:10.1016/j.cose.2024.104067.
11. Song HM, Woo J, Kim HK. In-vehicle network intrusion detection using deep convolutional neural network. Veh Commun. 2020;21(1):100198. doi:10.1016/j.vehcom.2019.100198.
12. Althunayyan M, Javed A, Rana O. A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning. Veh Commun. 2024;49(6):100837. doi:10.1016/j.vehcom.2024.100837.
13. Bhavsar MH, Bekele YB, Roy K, Kelly JC, Limbrick D. FL-IDS: federated learning-based intrusion detection system using edge devices for transportation IoT. IEEE Access. 2024;12:52215–26.
14. Dhanabal L, Shantharajah DSP. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. 2015 [cited 2026 Jan 1]. Available from: https://api.semanticscholar.org/CorpusID:16298036.
15. Abou El Houda Z, Moudoud H, Brik B, Khoukhi L. Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing. IEEE Trans Intell Transp Syst. 2024;25(7):7661–72. doi:10.1109/tits.2024.3351699.
16. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS); 2015 Nov 10–12; Canberra, Australia. p. 1–6.
17. Ullah I, Deng X, Pei X, Mushtaq H, Uzair M, Qayyum S. A blockchain-based federated learning framework against poisoning attacks in the internet of vehicles. Comput Netw. 2025;272(1):111705. doi:10.1016/j.comnet.2025.111705.
18. Rajapaksha S, Madzudzo G, Kalutarage H, Petrovski A, Al-Kadri MO. CAN-MIRGU: a comprehensive CAN bus attack dataset from moving vehicles for intrusion detection system evaluation. In: Proceedings of the Symposium on Vehicles Security and Privacy (VehicleSec) 2024; 2024 Feb 26; San Diego, CA, USA. p. 1–11.
19. Tang Y, Zhang Y, Niu T, Li Z, Zhang Z, Chen H, et al. A survey on blockchain-based federated learning: categorization, application and analysis. Comput Model Eng Sci. 2024;139(3):2451–77. doi:10.32604/cmes.2024.030084.
20. Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst. 2022;33(12):6999–7019. doi:10.1109/tnnls.2021.3084827.

21. Sun T, Li D, Wang B. Decentralized federated averaging. IEEE Trans Pattern Anal Mach Intell. 2023;45(4):4289–301. doi:10.1109/tpami.2022.3196503.
22. Pillutla K, Kakade SM, Harchaoui Z. Robust aggregation for federated learning. IEEE Trans Signal Process. 2022;70:1142–54. doi:10.1109/tsp.2022.3153135.
23. Bej S, Davtyan N, Wolfien M, Nassar M, Wolkenhauer O. LoRAS: an oversampling approach for imbalanced datasets. Mach Learn. 2021;110(2):279–301. doi:10.1007/s10994-020-05913-4.
24. Forman G, Scholz M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. ACM Sigkdd Explor Newsl. 2010;12(1):49–57. doi:10.1145/1882471.1882479.
25. Zhao L, Jiang J, Feng B, Wang Q, Shen C, Li Q. SEAR: secure and efficient aggregation for byzantine-robust federated learning. IEEE Trans Dependable Secur Comput. 2022;19(5):3329–42.