Private and Utility Enhanced Intrusion Detection Based on Attack Behavior Analysis With Local Differential Privacy on IoV

Rui Chen[a,*], Xiaoyu Chen[a] and Jing Zhao[a]

[a]School of Software Technology, Dalian University of Technology, DaLian 116024, China

ARTICLE INFO

Keywords:
Internet of Vehicles
Behavior Analysis
Intrusion Detection
Privacy Protection
Federated Learning

ABSTRACT

In recent years, with the escalating security demands of the Internet of Vehicles (IoV), concerns over safety have intensified. To prevent security incidents and privacy breaches, IoV must address various threats promptly and effectively. The use of deep learning methods for intrusion detection in IoV has garnered widespread attention. Compared to traditional security defenses, deep learning can learn from heterogeneous data sources, enhancing the accuracy of detecting various security threats. However, current research based on deep learning primarily focuses on constructing intrusion detection models and overlooks the analysis and processing of extensive behavioral data. Moreover, model training requires access to and transmission of sensitive data, which may lead to high communication costs and potential privacy leaks. To ensure the network security of IoV, we propose FDL-IDM, an innovative behavior-analysis-based intrusion detection model leveraging differential privacy within federated learning. It extracts driving behavior spatiotemporally and employs noise perturbation pre-aggregation, reducing communication costs and ensuring privacy without compromising accuracy. Specifically, we process data from both temporal and spatial dimensions. Data are grouped based on sender identity and then sliced according to the time sequence to create state matrices that vary over time, enhancing the performance and robustness of the detection model. Next, we incorporate an attention mechanism to merge outputs from each time step and hidden layer, strengthening the time series model and reducing information loss. Lastly, in federated learning, we add noise perturbation to the uploaded parameters, reducing the risk of privacy breaches. Additionally, we employ a random scheduling strategy during training to select clients and assign an adjusted learning rate that decreases with iterations, enhancing the stability of model training. Therefore, FDL-IDM helps prevent security attacks and protect IoV privacy. Through experiments and privacy analysis, as well as tests on vehicle-level devices, FDL-IDM achieved F1-scores of 0.9751, 0.9851, and 0.9789 on three public datasets, demonstrating not only high accuracy but also robust privacy protection capabilities.

1. Introduction

In recent years, with the rapid development of mobile communication technology, the Internet of Vehicles (IoV), as an emerging and promising paradigm within the new generation of Intelligent Transportation Systems (ITS), is anticipated to bring revolutionary changes to the underlying communication and transportation infrastructure[19]. As vehicles connect to mobile communication networks, achieving interconnectivity with surrounding infrastructure and the public internet, they may be exposed to various network attacks[37]. These assaults have the potential to take control of vehicles on the road, posing a serious challenge to human life and safety[37, 28, 40, 7]. From Fig. 1, the security threats faced by the IoV can be primarily categorized into two types: one pertains to attacks on the IoV itself, and the other relates to attacks associated with the vehicle's connection to the external world network.

Although numerous traditional security mechanisms, such as encryption and decryption techniques and identity authentication technologies, have been deployed in IoV, these mechanisms often lack proactive defensive capabilities and are not sufficiently efficient in timely detecting new types of attacks. Therefore, it is particularly important to research and develop intrusion detection models that can protect communication entities and vehicle data from malicious attacks. By employing intrusion detection technology, IoV systems can promptly identify and respond to various degrees and types of network attacks, effectively isolating compromised network regions or switching the system to a secure mode, thereby significantly reducing safety threats during vehicle operation. Intrusion detection technology driven by deep learning is capable of processing massive amounts of data and learning from heterogeneous sources, greatly enhancing the accuracy of IoV devices in detecting various security threats within the IoV[1]. Despite the widespread application of deep learning technologies in IoV intrusion detection systems, current systems still face three main challenges.

The first challenge is that current research on intrusion detection focuses on model construction while neglecting behavior analysis. This leads to a decline in model detection performance in different communication scenarios and when dealing with various attacks, as well as issues with the model's capacity to converge effectively during training. For instance, during vehicle operation, the sunroof command is not frequently used. Behavior analysis can reveal that if the sunroof is used frequently at a certain point by the user, such a command is highly suspicious[4, 19, 41]. The second challenge lies in the fact that IoV data in vehicle contain a considerable amount of user privacy. Current models, during training, involve a vast array of private data. However,

*Corresponding author
72117004@mail.dlut.edu.cn (R. Chen); chenxyz@mail.dlut.edu.cn (X. Chen); zhaoj9988@dlut.edu.cn (J. Zhao)
ORCID(s):

current intrusion detection research does not address this issue and directly uses unencrypted user data for model training and detection. If a hacker targets the detection model and acquires the unencrypted model data, then reverses this to obtain the original data, this could lead to a breach of IoV privacy data[43, 42, 30]. The third challenge is that communication bandwidth within the IoV is still a very precious resource. Current model training adopts distributed training methods to accelerate the process, but this approach can impose a significant burden on communication due to the exchange of training data[42, 19, 38].

To address the existing challenges, we propose a novel differential privacy-preserving federated learning-based intrusion detection model called FDL-IDM centered around attack behavior analysis. This model's data processing algorithm analyzes driving behavior from a spatiotemporal state perspective, yielding data that includes temporal state features. At the same time, noise perturbation is added prior to parameter aggregation, which reduces communication cost and protects model privacy.

Firstly, we propose a dataset processing algorithm that analyzes IoV communication data, a rich source of user behavior information as reflected in the driver's operational patterns [16]. This information is critical for differentiating between legitimate driving activities and hacker-induced anomalies. The algorithm harnesses the homogeneity property in the data, which stems from consistent driving behaviors, to categorize the data both temporally and spatially. By organizing the data based on the source address and capitalizing on this homogeneity, we construct dynamic state matrices. This approach not only captures the evolution of driving behavior over time but also enhances the detection capabilities for anomalous activities.

Subsequently, an improved temporal sequence model is employed to convert the segmented data blocks into vector data forms of behavioral characteristics. That is, the temporal sequence model uses an additive attention mechanism [14] to combine the output of each time step with the final hidden layer through additive computation, thereby endowing the behavioral features with more behavior information and further enhancing the accuracy and stability of the detection model.

Finally, We employ a federated learning algorithm that trains the detection model without sharing local clients' data with the central server[39, 44]. The server aggregates perturbed model parameters, enhanced with differential privacy noise [34, 22], reducing communication cost and bolstering privacy. A stochastic scheduling method [39] aligns training with real-world scenarios and improves robustness. Our analysis and experiments confirm the algorithm's compliance with differential privacy standards.

Above all, we are the major contribution of this work is as follows:

- For the first time in the field of IoV intrusion detection, we propose federated learning with differential privacy, innovatively proposing a data processing algorithm for driving behavior analysis that considers both temporal and spatial dimensions. The aim is to significantly enhance the effectiveness of model training at reduced communication costs while simultaneously minimizing the risk of privacy breaches for clients, and maintaining the precision of the model's detection capabilities.

- We devise a novel data processing algorithm for user driving behavior analysis across temporal and spatial dimensions based on our principle of data homogeneity. It identifies behavior-rich data from targeted driver-device interactions within specified timeframes. Grouping and normalizing this data, we create a time-reflective state matrix, boosting the accuracy and reliability of federated learning models. Additionally, we improve our temporal sequence detection model with an additive attention mechanism that enriches behavior features, enhancing detection precision.

- We use federated learning to train our proposed model, reducing communication pressure between clients and central servers. We use a random strategy mechanism to select the clients to be trained, making the model training more stable and with better generalization performance, with small errors between training and test datasets. We also use the Laplace noise mechanism to add perturbation to the local model's weight parameters, ensuring that even if a hacker obtains the weight parameters, they cannot reverse engineer the original data, further protecting the model's privacy.

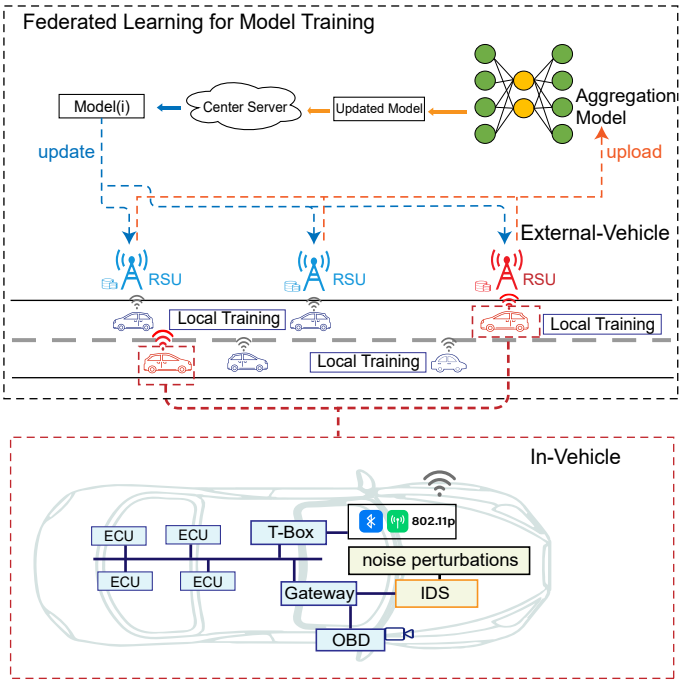


Figure 1: A hierarchical example of the IoV detection model.

- We conduct experiments and validations on three publicly available datasets, UNSW NB-15[26], CAN-intrusion-dataset[20], and CIC-IDS 2017[31], and compare our model with the latest models. The experimental results show that our model achieves F1 values of 0.9751, 0.9851 and 0.9789 respectively. Additionally, we publicly release our dataset processing algorithm and model implementation code to reduce the scarcity of reproducible and effective models in this field.

The rest of this paper is organized as follows. Section 2 summarizes background and related work. In section 4, we present a background on federated learning versus local differential privacy, and the deep learning algorithms used by the intrusion detection model. In Section 3, we introduce the threat models that need to be studied in this paper. In section 5, we elaborate on the overall system design of the intrusion detection model, the newly designed datasets processing algorithm and the detection model training process. Section 6 and 7 presents comparative experiments on the detection models. Section 7.7 concludes the work.

2. Background and Literature Review

IoV security has gradually become a focal point of concern in both research and industry circles. This section will review research works in the related fields to identify gaps our study aims to fill.

2.1. Intrusion Detection in In-Vehicle Network of IoV

In the field of intrusion detection research for in-vehicle networks, a variety of methods have been proposed to identify anomalous vehicle behaviors and faulty sensors. Notably, some of these methods identify anomalies by analyzing patterns of normal behavior, effectively reducing computational costs. These methods do not require pre-labeling of attack data; instead, they detect by constructing models of normal behavior, where any deviation from established patterns may indicate an anomaly. For instance, Almutlaq [4] developed a method utilizing a set of rules to detect attacks on the vehicle's CAN bus, significantly reducing the performance overhead of the model.

With the rise of deep learning technologies, due to their excellent automatic classification capabilities and the advantage of obviating manual intervention, these have been widely applied in the field of vehicle anomaly detection. For instance, Stefano et al. [23] proposed an intrusion detection system (IDS) using a Long Short-Term Memory (LSTM) autoencoder, which works by creating a reconstructed sequence of CAN data and comparing it to the actual sequence to detect anomalies. However, this method may underperform in detecting complex data fabrication attacks, where an attacker could compromise sensors or electronic control unit (ECU) and send fabricated, legitimate-looking data to the CAN bus. Sun [32] introduced a novel IDS that integrates one-dimensional convolution, Bi-directional Long Short-Term Memory networks (Bi-LSTM), and attention mechanisms, demonstrating superior detection performance. Nonetheless, the direct dropout strategy employed while processing the hidden layers of sequences could result in a significant loss of valuable detection information. Van Wyk [35] proposed an anomaly detection approach that combines deep learning convolutional neural networks with Kalman filters. However, this method assumes that attacks occur independently on single sensors, which may not detect more complex, coordinated attack types.

Although the research by Wang et al. [38] mentioned that multiple clients could utilize local data for collaborative training to defend against federation attacks, the security of the model itself and the communication cost during training have not been fully considered. In the study presented in [13], the authors proposed an intrusion detection model for vehicular networks based on a CAN image encoding scheme. While maintaining a lightweight design, the model also achieved good detection performance. However, the study did not fully consider the specific requirements of vehicular devices. In [15], considering the privacy protection and communication cost issues at the vehicle-side, the authors proposed an intrusion detection framework that integrates federated learning with transfer learning. Meanwhile, in [6], the authors addressed the issue of slow model convergence in CNN models due to improper setting of hyperparameters. By optimizing the configuration of the CNN's hyperparameters, the robustness of the model was enhanced. In existing research on federated learning for intrusion detection, the detection speed is often slow. The study in [42] broke this limitation, achieving a response time of less than 3 milliseconds during the experimental testing phase, and the model training process is transparent to vehicle operations, not affecting normal driving behavior.

2.2. Intrusion Detection in External-Vehicle Network of IoV

Intrusion detection for external-vehicle network, has also garnered widespread attention in the research community. Aloqaily [5] proposed an IDS for IoV utilizing a combination of DBN and DT algorithms. This approach demonstrated high detection accuracy on the NSL-KDD dataset, although it exhibited relatively high latency, especially when the number of vehicle-to-vehicle nodes was limited.

Other research efforts have also centered on the development of IDS for general networks, evaluating their methods using benchmark datasets. For example, Injadat et al. [18] introduced a novel multi-level optimization IDS based on machine learning that features low model complexity and requires less data for effective network attack detection. The performance of the model was evaluated on the CIC-IDS 2017 and UNSW-NB15 datasets. Kumar [19] combined blockchain and deep learning technologies to address security vulnerabilities between vehicles, enhancing the privacy, transparency, verifiability, scalability, and integrity of IoV data. However, the model neglected the potential communication cost incurred during the training process. Almutlaq

Table 1
Comparison of Related Works on Intrusion Detection Model for In-Vehicle Network of IoV

Work	Detection Techniques	Contribution	Emerging Attack (Zero-Day)	(V) Vehicle Level (B) Behavior Analysis (P) Privacy Protection	Strength	Weakness
[4]	Rule Set	Utilizing a two-stage deep learning model with rule extraction techniques to detect suspicious network activities in IoV.	✘	✘ B ✘	Detection results are credible, and the classification of malicious activities is effective.	Detection of new malicious activities exhibits significant fluctuations in results, and requires manual intervention for rule configuration.
[23]	LSTM	Utilizing an LSTM-based autoencoder to detect anomalies in the CAN network.	✓	✘ B ✘	Do not need prior understanding of anomalous activity semantic information.	Insufficient in detecting sophisticated data forgery attacks.
[32]	CNN, Bi-LSTM, Attention	In-Vehicle network intrusion detection using CNN, Bi-LSTM, and Attention Mechanism models.	✓	✘ ✘ ✘	Able to implement a low-latency, generic intrusion detection model without the need for understanding the encode knowledge of in-vehicle networks.	Discarding the hidden layers in time series models may result in a substantial loss of valuable information for detection.
[35]	1-dimension Convolution	Using CNN and Kalman filtering with a x^2-detector model to detect anomalous behaviors in the IoV.	✓	✘ ✘ ✘	High detection rate.	Unable to detect coordinated attacks.
[38]	CNN	Utilizing spatiotemporal features in sensor data to detect isolated and coordinated attacks in the IoV.	✓	✘ B ✘	Able to detect coordinated isolated and attacks.	Communication cost has not been adequately considered.
[41]	Tree Based Models	A multi-layered hybrid intrusion detection system based on features and anomalous code.	✓	V ✘ ✘	Considering the IoV environment, efficient detection results have been achieved.	Relies on centralized training, unable to facilitate collaborative training among vehicles.
[13]	CNN	Intrusion detection model based on CAN image encoding scheme	✓	V ✘ ✘	Has good detection performance, lightweight model.	The model lacks privacy protection.
[15]	MMD, FL, Transfer Learning	An intrusion detection framework for CAN networks based on the integration of federated learning and transfer learning.	✘	✘ ✘ P	Select data highly relevant to intrusion detection from source domains similar to the target domain.	The model takes an excessively long time for detection.
[6]	FI, CNN	IoV intrusion detection model based on CNN with hyperparameter optimization.	✓	V ✘ ✘	Optimized the learning rate, dropout rate and freeze layers hyperparameters in CNN.	The vehicle-side model lacks privacy protection and is difficult to implement on vehicle-side devices.
[42]	FL, GNN	Proposed a GNN-based intrusion detection system that can detect in-vehicle network threats within 3ms.	✓	✘ B P	Model training is transparent to vehicles, with fast detection speed.	The complexity of GNN-based models is too high, making data preparation and model training very difficult for attack detection.

LSTM: long-short term memory network; CNN: convolutional neural network; Rule Set: a set of rule algorithms extracted from deep learning; Bi-LSTM: bi-directional long-short term memory networks; MMD: Maximum Mean Discrepancy; GNN: graph neural network.

[4] employed interpretable neural networks for intrusion detection in IoV to mitigate the overhead introduced by deep learning. Nevertheless, the detection model failed to fully consider its own security, potentially causing secondary damage to the model in the detection of external IoV intrusions. Li [41] proposed a multi-tiered hybrid intrusion detection system based on features and anomalous codes that achieved efficient detection results and considered the IoV environmental factors. Oseni et al. [27] presented an explainable neural network intrusion detection model based on the Deep SHAP method [24]. This model facilitates a deeper understanding of the details and principles of Internet of Vehicles (IoV) security threats for network security experts by extracting explainability rules from trained deep learning models. However, when encountering new types of security

Table 2
Comparison of Related Works on Intrusion Detection Model for External-Vehicle Network of IoV

Work	Detection Techniques	Contribution	Emerging Attack (Zero-Day)	(V) Vehicle Level (B) Behavior Analysis (P) Privacy Protection	Strength	Weakness
[5]	DBN DT	Detect anomalous communication between vehicles by integrating DBN and DT algorithms.	✓	✘ ✘ ✘	High accuracy and low false positive rate.	Even with few vehicle-to-vehicle nodes, the latency remains relatively high, and the algorithm complexity is also high.
[18]	Machine Learning	Using training data sampling techniques, a low-latency intrusion detection framework based on minimal training datasets has been designed.	✓	✘ ✘ ✘	The model has low complexity and relies on smaller datasets.	Traditional machine learning algorithms.
[19]	LSTM, Blockchain	By integrating blockchain and deep learning technologies, the defensive capabilities of vehicle systems against security threats have been enhanced.	✓	✘ ✘ P	While implementing attack detection, it also features proactive privacy protection capabilities.	As data volume increases, it becomes challenging to implement a blockchain-based detection model on vehicle endpoints.
[41]	Tree Based Models	A multi-layered hybrid intrusion detection system based on features and anomalous code.	✓	V ✘ ✘	Considering the IoV environment, efficient detection results have been achieved.	Relies on centralized training, unable to facilitate collaborative training among vehicles.
[27]	Explainable DL	Proposed an interpretable neural network using the Deep SHAP method to detect security threats in the IoV.	✘	✘ ✘ ✘	Helps cybersecurity experts better understand the details and principles of IoV security threats.	For new security threats, rules need to be re-extracted.
[43]	Fl, LSTM	A privacy-preserving detection model for IoV security threats based on federated learning.	✓	✘ ✘ ✘	Considering multi-level information fusion, including data from the physical layer and application layer.	In federated learning training, the model on the vehicle-side lacks privacy protection.
[29]	FL, Distillation	Intrusion detection model using federated distillation algorithm based on FL baseline.	✓	✘ ✘ ✘	The model has a high accuracy.	The distilled model was not validated on the vehicle-side.
[36]	FL, fuzzy logic	FL-based fuzzy logic with IoV intrusion detection model.	✘	✘ ✘ ✘	Resolved the difficulty of adoption under the condition of non-IID vehicle data.	The detection model uses rule extraction, which requires resetting the rules for newly emerging attacks.
[30]	FL, EMs	Proposed an intrusion detection framework based on federated learning that aggregates nodes with similar data distributions for intrusion detection.	✘	✘ ✘ ✘	Able to adaptively reduce the aggregation weights of those below the standard model.	The intrusion detection model demonstrated unstable performance across different datasets.

LSTM: long-short term memory network; Rule Set: a set of rule algorithms extracted from deep learning; DBN DT: deep Belief Network with decision tree; Explainable DL: Explainable Deep Learning; EMs: evaluation metrics.

threats, the model requires the re-extraction of rules to adapt to the new threat environment.

However, this method relied on centralized training, which could lead to additional privacy leakage issues if the model were intercepted. Ayodeji[27] proposed an interpretable deep learning-based intrusion detection framework that enhances the transparency and resilience of deep learning-based IDS within IoV. Although it has improved the efficiency of IDS detection to a certain extent, interpretable deep learning still falls short in effectively uncovering potential attacks.

Although research on IDS for IoV has made certain strides, in our previous study [43], we adopted a federated learning framework to train models for IoV attack detection, effectively mitigating privacy leakage issues. Nonetheless, this approach still has limitations. In practice, the model [17] parameters exchanged between clients and the central server could be susceptible to inference attacks. Rani [29] employed a federated distillation-based intrusion detection

model with federated learning baseline. The model achieved high accuracy with low resource usage. However, no real vehicle-side validation was done. [36] adopted a FL-based IoV intrusion detection model, handling non-IID vehicle-side data via sampling techniques. It used rule extraction, requiring new rules when facing new threats. [30] proposed a federated learning intrusion detection framework, aggregating nodes with similar distributions and selecting optimal local models, improving convergence. But experiments exhibited instability across datasets.

In summary, Tables 1 and 2 comprehensively summarize related in-vehicle and external-vehicle network intrusion detection research, covering main contributions, detection techniques, and strength/weakness. Moreover, we analyzed whether these studies address key IoV intrusion detection focuses like privacy protection and vehicle-side validation. Despite extensive research, current focuses remain on model construction, with gaps in communication cost, vehicle-side privacy, and driver behaviors for federated learning-based detection.

Therefore, an IDS suitable for detecting intrusions in in-vehicle and external networks of IoV is needed. Our proposed IDS outperforms existing IoV intrusion detection research in several aspects. First, we process IoV communication data based on driver behaviors through temporal and spatial dimensions, generating time-varying state matrices to improve detection robustness. Next, our IDS adopts federated learning, training on public datasets of non-sensitive features to mitigate private data leakage risks, and adding appropriate noise perturbation on the client side. Finally, compared to other ML/DL-based IDS, our proposed model FDL-IDM demonstrates higher accuracy in improving detection rates.

3. Threat Model

IoV face various types of attacks including denial-of-service, deception, backdoor, man-in-the-middle, and forged data injection. In particular, external-vehicle or in-vehicle network communicating with the outside world are highly vulnerable to attacks of varying degrees, as shown in Fig. 1. For instance, packet interception can lead to privacy leaks, and more seriously, the hijacking of driving control during traveling poses major safety threats to drivers and other road users. Fortunately, intrusion detection techniques can isolate attacked networks or switch vehicles to a safe mode, thereby mitigating risks in driving. Owing to the capacity of processing large-scale heterogeneous data and learning features, deep learning can detect different security threats in IoV with high accuracy.

However, current deep learning based research focuses more on constructing intrusion detection models, while neglecting data processing. Yet the data contains substantial information that models need. For example, in normal driving, driver operations exhibit continuity on the time axis. In IoV, brake and throttle CAN bus signals alternate during driving, while also transmitting steering signals. Additionally, interactions between different target devices reflect spatial characteristics. A new signal sender may correlate with the brake, throttle and steering wheel. Such behavioral information hides in the communication signals, requiring specially designed algorithms aligned with behavior analysis to uncover, so as to enhance model robustness.

Meanwhile, conventional machine learning requires uploading all raw data to the cloud for model training, incurring enormous communication costs and serious privacy risks. In federated learning, a central server coordinates clients to train detection models via multiple rounds of global iterations. In each round, the server randomly selects clients to distribute the latest model, and clients train using local private data before sending updated models back to the server. However, federated learning still cannot guarantee privacy, as advanced inference attacks can infer sensitive training data from uploaded model parameters. For instance, given inputs and the target model, membership inference attacks can train an attack model to determine if a sample was used to train the target model. Therefore, considering the IoV context, privacy-preserving techniques need to be proposed to enhance privacy protection capabilities.

4. Preliminaries

In this section, we present a comprehensive study of a deep learning-based IDS for the IoV, then elaborate on the threat model underpinning the IDS and offer an in-depth exposition, along with the mathematical derivation, of the pivotal techniques implemented within the detection model.

4.1. Federated Learning with RNN-Attention Deep Learning Model

4.1.1. Federated Learning

Fig. 1 illustrates a federated learning-based intrusion detection system that integrates a IoV with an external network, consisting of a central server and multiple clients $c_1 \sim c_n$ (representing various vehicle-sides).

Within this system, each client c_i (where $i \in [1, n]$) independently possesses a set of data samples denoted as D_i. The primary objective of the federated learning detection system is to employ the data samples from the clients to train a machine learning model. Specifically for each client c_i, they independently hold a training dataset $D_i = (X_i, Y_i)$, with the model parameters denoted by θ_i, where i refers to the i-th client. X_i and Y_i correspond to the training data and labels of the i-th client, respectively. The goal of the federated learning detection system is to train an intrusion detection model using the clients' data samples, represented by the function $f_\theta : X_i \rightarrow Y_i$. The loss of the model on the data samples D_i is calculated using the loss function $Loss(f_\theta(X_i), Y_i)$.

To protect data privacy, the federated learning detection system process involves multiple global iterations by transmission differential privacy noise-perturbed gradients between the clients and the server.

a) At the beginning of global iteration t, the server distributes the latest model parameters (denoted as θ_t) to

a randomly selected subset of clients (vehicle-side) (represented by S_t);

b) The selected client c_i conducts local iterations using its private dataset D_i with the latest model parameters θ_t to obtain the gradient $g_{i,t}$, then adds Laplace noise Lap($\frac{\Delta s}{\epsilon}$) to generate the perturbed gradient $a_{i,t}$, which is sent back to the server;

c) The server aggregates the returned noisy gradients into $\theta_{t+1} \leftarrow \theta_t - \sum_{i \in S_t} \alpha_i a_{i,t}$, where α_i is the weight for the client i, with $\alpha_i = \frac{|D_i|}{|D|}$. After the aggregation process, the server initiates a new round of global iteration with θ_{t+1}.

The local independent dataset D_i of the client integrates communication data from both the external/in-vehicle of IoV. These communication data are arranged in a temporal sequence, and at any given time step t_r, a small batch of input data is represented as $x_t \in R^{m \times d}$, where m denotes the number of samples in the training dataset, and d represents the dimensionality of features for each sample. Concurrently, we set the activation function for the hidden layer as σ.

4.1.2. RNN-Attention Deep Learning Model

In the Bi-LSTM, the forward and backward hidden states at each time step are denoted as $\overrightarrow{H}_t \in R^{m \times h}$ and $\overleftarrow{H}_t \in R^{m \times h}$, respectively, where h indicates the number of computational units in the hidden layer. The training update process for the forward and backward hidden states is described as follows:

$$\begin{aligned} \overrightarrow{H}_t &= \sigma(x_t \overrightarrow{W}_{xh} + \overrightarrow{H}_{t-1} \overrightarrow{W}_{hh} + \overrightarrow{b}_h) \\ \overleftarrow{H}_t &= \sigma(x_t \overleftarrow{W}_{xh} + \overleftarrow{H}_{t+1} \overleftarrow{W}_{hh} + \overleftarrow{b}_h) \end{aligned} \tag{1}$$

In this model, the forward weight parameters $\overrightarrow{W}_{xh} \in R^{d \times h}$, $\overrightarrow{W}_{hh} \in R^{h \times h}$, and the backward weight parameters $\overleftarrow{W}_{xh} \in R^{d \times h}$, $\overleftarrow{W}_{hh} \in R^{h \times h}$, along with the forward bias $\overrightarrow{b}_h \in R^{1 \times h}$, and the backward bias $\overleftarrow{b}_h \in R^{1 \times h}$ are all part of the model's weight parameters; these are updated globally after local training is completed.

By concatenating the states of the forward and backward hidden layers, we obtain the hidden state $H_t \in R^{m \times 2h}$ to be transmitted to the output layer. In a Bi-LSTM with multiple hidden layers, this information serves as the input to the next bidirectional layer, and this process occurs iteratively at each time step. Moreover, at each time step in the computation, there is an output layer that outputs the result of the time step's computation $O_t \in R^{m \times q}$, where q represents the number of units in the output layer.

$$O_t = H_t W_{hq} + b_q \tag{2}$$

In this paper, the weight $W_{hq} \in R^{2h \times q}$ and bias $b_q \in R^{1 \times q}$ constitute the model parameters for the output layer. In the application of bidirectional recurrent neural networks, it is common practice not to directly merge the backward $\overleftarrow{H}_t$ and forward $\overrightarrow{H}_t$ hidden states into H_t. Instead, they participate independently in the output layer computation, producing two separate outputs at each time step, while the hidden layer also includes matrices for both forward and backward states.

Inspired by the Sequence-to-Sequence (Seq2Seq) model [33] and our research findings, we investigated that in existing studies, time-series models such as LSTM and Bi-LSTM typically use the final hidden layer as the output result, while outputs at each time step, O_t, are often discarded and not included in the model training. To enhance the model's detection capabilities, this paper introduces an additive attention mechanism that combines the output at each time step O_t with the corresponding hidden state H_t, computing the attention distribution through addition. Specifically, the additive attention is calculated through a fully connected layer as $V^t tanh(W_q Q + W_k K)$. Here, the activation function $tanh$ is used, where O_t serves both as the query Q and the key K, while the hidden state H_t acts as the value V^t. This computation method effectively integrates the output of each time step with the hidden state information.

We found that the output at each time step contains a wealth of information crucial for model detection. Discarding these outputs can lead to significant model instability. By effectively utilizing these time step outputs, we can significantly enhance the model's accuracy and robustness. In the ablation study presented in Section 7.5, we validated the significant contributions of the improved time-series detection model and the attention mechanism implemented herein.

4.2. Federated Learning Privacy Analysis

In this paper, it is assumed that the central server is trusted, but there is a risk of external hackers infiltrating the network to acquire unprocessed weight data of the vehicle's detection model. Although the personal data set D_i of the i-th client is used only for local model training, the model weight parameters θ_i need to be shared with the central server. This may lead to the leakage of the client's private information, as demonstrated by model inversion attacks. For instance, in [12], the authors demonstrated a method for equation-solving model extraction attacks on linear models, where seemingly legitimate queries are used to solve linear equations and obtain the demonstrated model information. In [22], the authors presented a model inversion attack that could recover recognizable facial images solely from the name of the model trainer and access to the machine learning model weight parameters.

Although IoV intrusion detection based on federated learning can reduce the risk of data leakage from the client to some extent, during the process of weight aggregation between the client and the central server, hackers can analyze the global parameters θ and use reverse engineering on the unencrypted weights θ to solve $X_i = f_\theta(Y_i)'$ for the client's local data D_i, thereby obtaining private information about the vehicle, such as its location and unique identification number.

4.3. Local Differential Privacy

Local differential privacy enables the protection of data privacy while maximizing query accuracy, typically by adding randomized noise to prevent attackers from obtaining the original data[10, 11]. During the distributed model training process, the transmission of model weight data may lead to privacy breaches[34, 22]. Local differential privacy is applicable to distributed federated learning and can achieve the protection of private data during the local training process[39]. Unlike traditional centralized differential privacy, local differential privacy focuses on the privacy protection during the data collection process, does not require the assumption of a trusted third party, and can also prevent model inversion attacks by adversaries with prior knowledge.

Definition 1 Differential Privacy: Assume two datasets D and $D^{'}$ that differ by a Hamming distance of 1, and consider a randomized algorithm A whose outputs follow a certain distribution.

$$Pr[A[D] \in S] \leq e^{\epsilon} Pr[A[D^{'} \in S]] \tag{3}$$

Where $\varepsilon > 0$ is the differential privacy budget; the smaller the privacy budget, the greater the noise added, resulting in a higher level of privacy protection. $D^{'}$ can be understood as being generated by perturbing the dataset D. For any given datum in the perturbed dataset $D^{'}$, the probability ratio Pr compared to any datum in the original dataset D is less than e^{ϵ}.

Definition 2 Global Sensitivity: The motivation for studying this issue arises from our inability to restrict the types of queries users make on a dataset. When the query is about the number of individuals, the difference between adjacent datasets is small (differing by only 1), so adding a small amount of noise can obfuscate the results between the two; however, when querying something like individual salaries, where the differences between data points are significant, even a slight change can make the disparities in the data evident. Thus, adding only a small amount of noise is clearly insufficient to meet the application requirements. Therefore, the design of a differential privacy mechanism is closely related to the nature of the queries. Given that the absence of even a single record in a dataset can have a certain impact on the query results, it is necessary to quantify the maximum extent of this impact, such as through the computation of a sensitivity measure denoted by Δs [39].

$$\Delta s = \max_{\forall \theta, D} ||s(D) - s(D^{'})||_1 \tag{4}$$

The term Δs represents the quantified value of sensitivity, which characterizes the maximum change in the output s due to the alteration of a single record. Upon determining the sensitivity, it is necessary to add noise drawn from a Laplace distribution with a mean of zero, in accordance with the sensitivity of the adjacent datasets D and $D^{'}$.

Definition 3 Laplace Noise Perturbation: Given a dataset D and query parameter θ, a Laplace mechanism satisfying ϵ-DP would perturb the query result a_L by adding noise as follows:

$$a_L(w, \epsilon) = g(w, D) + Z \tag{5}$$

Where Z represents the noise generated from the Laplace distribution, with the probability density function $Pr(Z) = \frac{\epsilon}{2\Delta s} \exp\left(-\frac{\epsilon|Z|}{\Delta s}\right)$, which can also be denoted as $Z \sim \text{Lap}\left(\frac{\Delta s}{\epsilon}\right)$. According to Definition 1, the consumption of the privacy budget is closely related to the act of responding to queries.

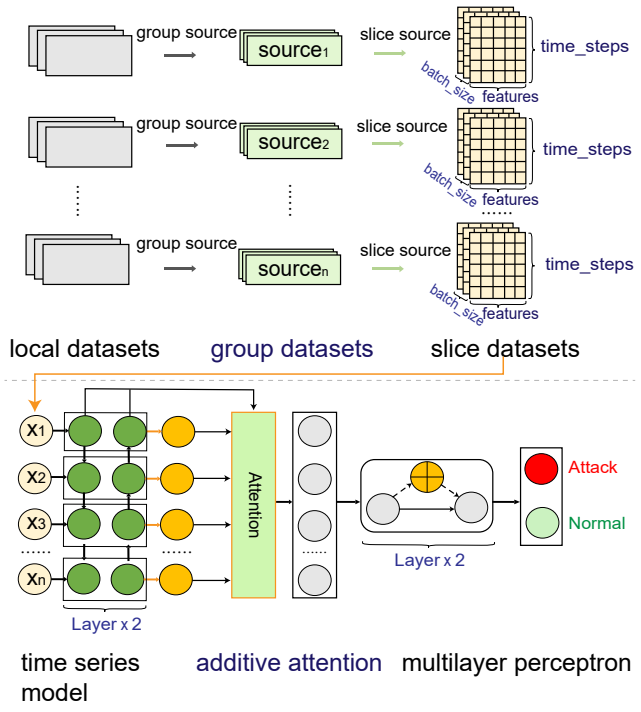


Figure 2: Intrusion detection model framework incorporating a data processing algorithm based on user behavior and a temporal sequence model incorporating attention mechanisms.

5. Proposed Method

In this section, we first introduce the system framework of the intrusion detection model, and then propose a dataset processing algorithm that combines local differential privacy. We analyze the application of the intrusion detection federated learning model and its relationship with the performance of local differential privacy. Furthermore, we prove that our proposed intrusion detection model can satisfy the requirements of differential privacy by adding appropriate noise perturbation on the client side.

5.1. Intrusion Detection System Framework

To address the intrusion detection problem in both external-vehicle networks and in-vehicle networks described in Section 3 Threat Model, we have devised a novel federated

Table 3
Key Notations and Abbreviations

Notations or Abbreviations	Description
c_i	The i-th client.
D_i	The i-th client's local data.
θ	Universal weights representing the model, with θ_i denoting the model weights of the i-th client.
f_θ	An intrusion detection model with weights denoted by θ.
m	the number of dimension contained in D_i.
h	hidden layer.
$\overrightarrow{H_t}$ and $\overleftarrow{H_t}$	The forward and backward of the bidirectional temporal sequence hidden layer.
t	Rounds for global model aggregation in federated learning.
X_i^g	The data grouped within the i-th client.
$r_{i,x}$	In the i-th client, a particular feature value x within the private data falls within the range $[1, m]$.
s_i	In the i-th client, structured data is in the form of a state matrix that varies over time.
ITS	Intelligent Transportation Systems
IDS	intrusion detection system
IoV	Internet of Vehicles
CAN	controller area network
ECU	electronic control unit

deep learning intrusion detection model (FDL-IDM) based on differential privacy. This model employs a comprehensive data set processing algorithm that considers both temporal and spatial dimensions of homogeneity, reducing communication cost per iteration during training and ensuring model accuracy without compromising data privacy.

From Algorithm 1, in the designed data processing algorithm, different clients $c_i, i \in [1, n]$ each possess their independent dataset D_i. Considering the homogeneity of the data, we first group the communication message datasets by sender address, starting from the spatial dimension. Subsequently, from the temporal dimension, we slice the grouped data according to the temporal sequence to form structured data with temporal states. This method is more effective in capturing the characteristics of data across time and space, providing richer information for subsequent intrusion detection.

From Fig. 4.3, during model training and detection, we combine the output of the temporal sequence model with the hidden layers using an attention mechanism to obtain vectors that encode vehicle behavior patterns. Then we apply a residual multi-layer perceptron for a nonlinear transformation to obtain the final attack detection results. Due to the mobility of vehicles, centralized server training would require sending model parameters to the in-vehicle terminals and uploading local training data to the central server. However, this data transmission process is susceptible to privacy risks due to unreliable communications, potentially leading to training data leakage. Moreover, extensive data exchanges would strain the IoV communication resources. To avoid these issues, we have designed a federated learning intrusion detection model training framework as shown in Fig. 1. Training data is distributed to local processors in different vehicles, creating a distributed training process. During local training, we apply a local differential privacy algorithm to process gradient updates, and the server aggregates the model weights uploaded by the vehicles. Even if attackers access the model weights, they cannot reverse-engineer the original parameters, ensuring the model's privacy and security.

Thus, intrusion detection in IoV involves two aspects: a data set processing algorithm that integrates the homogeneity property and the federated learning intrusion detection model based on differential privacy. By adopting a local differential privacy algorithm and a distributed training approach for the federated learning intrusion detection model, we can effectively detect intrusions while ensuring privacy and security.

5.2. Data Preprocessing

5.2.1. The Homogeneity Property

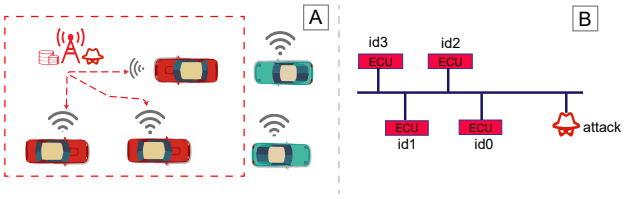


Figure 3: A: The *homogeneity property* of the attacker on the External-Vehicle Network. B: The *homogeneity property* of the attacker on the In-Vehicle Network.

As shown in Fig. 4.3, this study leverages the principle of homogeneity to uncover potential attack behaviors. In computer caching, there exists a general algorithm: after a user accesses a memory region, it is likely that the same region will be accessed again in the near future[21]. Inspired by the principles of computer caching, this research has found through data analysis that after accessing an address, users typically access the same address again within a close temporal proximity. This phenomenon has been defined as the homogeneity property.

More specifically, we have deeply considered both temporal and spatial dimensions. During normal driving, a driver's actions exhibit continuity over a temporal sequence. For example, in the context of IoV, the CAN bus messages for braking and throttle will alternately change as the vehicle moves, and steering operations will also be transmitted. In addition, the interactive changes between different target devices reflect spatial characteristics. For instance, a new

message's sender might be associated with three devices: the brakes, throttle, and steering wheel.

Therefore, within a certain time frame, the vehicle's operations related to starting, braking, throttle, and steering will alternate, demonstrating local continuity in both temporal and spatial. If commands related to vehicle doors or windows being opened and closed occur frequently during this period, it generally indicates an anomaly, which is a key indicator for intrusion detection.

In cases where attackers are more sophisticated, they might manipulate ECU devices only during specific times and forge messages, such as the sender identity and timestamps (i.e., controlling the timing of message sending). However, no matter how the timing of the messages sent is altered, there will be intervals between messages, and currently, there is no research capable of precisely characterizing the distribution of normal message intervals for different vehicle states. Deep learning, on the other hand, can learn this distribution by combining heterogeneous data and adjust timely with the continuous input of new data.

Furthermore, in the messages sent, besides the source address and timestamp, there are additional dimensional features, such as the angle of the steering wheel and the intensity of the throttle. These angles and intensities change over time, equivalent to a state matrix that varies with time. Deep learning can quickly learn the potential associations between these states through training data. Therefore, against such intelligent attacks, our approach can provide effective protection.

For joint attacks with the ability to intelligently modify messages, our data set processing algorithm combines homogeneity property through grouping and segmentation operations to form a message state matrix. These vehicle state matrices help to reveal the potential multidimensional data correlations between each ECU and others. This method also considers the case of multiple sensors, enhancing the robustness of the detection system against combined attacks.

As shown in Fig. 3A, in external-vehicle network attacks, attackers may perform different attacks on the same address to achieve their goals, often making multiple attempts, and the temporal sequence series is unordered. Even if attackers deliberately disguise their temporal sequence, they will expose themselves as attackers in other features. In contrast, normal users accessing the vehicle usually maintain a contextual relationship and will not arbitrarily choose to send messages from different vehicles multiple times. Similarly, in the in-vehicle network illustrated in Fig. 3B, the homogeneity property is better reflected. For example, when driving a car, users often call the brake and acceleration ECUs, rather than frequently opening the sunroof and windows in a short period of time.

5.2.2. Vehicle-side Training Data Grouping and Slicing

Based on the analysis of homogeneity and described in Algorithm 1, we processed the dataset by considering the spatial and temporal dimensions reflected by the sender of the sensor messages and the sequence of message transmission. Specifically, for the features and labels $\{X_i, Y_i\}$ in the local dataset D_i of the vehicle side, we group them according to the identifier of the sender. Given the aforementioned homogeneity, grouping can be done by the unique identifier of the sender; for example, in external-vehicle network data, the source IP address can be used as the basis for grouping, while in the in-vehicle network dataset, the CAN ID can serve as the grouping marker. We select the x-th feature $X_{i,x}$ from D_i as the grouping flag. Through this method, we obtain the dataset X_i^g grouped by the sender's identity, and the grouped data exhibits certain spatial properties. Furthermore, we need to consider these spatially characterized data from the temporal dimension to obtain structured data in the form of a state matrix that varies over time and is suitable for training.

In order to make the spatially characterized data suitable for training the detection model f_θ, we first normalize the feature values $r_{i,x}$ in the data X_i^g by employing the following formula before proceeding to slice the data: $r_{i,x} = \frac{r_{i,x} - r_{i,min}}{r_{i,max} - r_{i,min}}$. Subsequently, based on the identifiers of the senders, we store data with the same grouping identifier in the same file. This approach not only effectively reduces memory consumption but also facilitates the monitoring of groupings during algorithm execution to promptly detect any errors that may arise during data processing.

Next, we slice the data to construct a collection of datasets with temporal sequence states $\{s_1^i, s_2^i, ..., s_c^i\}$, where c represents the predefined time steps, thus obtaining structured data in the form of state matrices that vary over time.

During the slicing operation, it is imperative to ensure the continuity and order of the dataset's temporal sequence are preserved. If the volume of data in X_i^g exceeds the time step c, it should be truncated; if the remaining part is larger than time step c, this truncation process should be repeated; if the remaining data volume is less than c, it should be discarded. The purpose of this processing method is to obtain structured behavioral data that exhibits the state characteristics of different devices over time, providing precise input data for subsequent temporal sequence analysis.

5.3. Intrusion Detection Model with Local Differential Privacy

Our proposed intrusion detection model is well-adapted for scenarios in which vehicles communicate sensitive data to a potentially untrustworthy central server. Fig. 1 illustrates our intrusion detection framework, which incorporates a novel strategy for preserving privacy: local model gradients perturbation guided by a local differential privacy algorithm. This method represents an advancement over the conventional Federated Averaging (FedAvg) model, as introduced by McMahan[25], enhancing the protection against data privacy breaches on the vehicle-side.

In our approach, when local model weights are transmitted to the central server, they are obfuscated through the addition of noise adhering to a Laplace distribution—a mechanism known for its efficacy in differential privacy.

Algorithm 1 Dataset Processing Based on Homogeneity Property

Require: Local dataset $D_i = \{X_i, Y_i\}$ on the vehicle side
Ensure: Structured data $\{s_1^i, s_2^i, ..., s_c^i\}$ with the same time step c

Group the dataset D_i according to the homogeneity property based on identifier of the sender to obtain the grouped data X_i^g
Normalize the feature value $r_{i,x}$ in X_i^g using the formula $r_{i,x} = (r_{i,x} - r_{i,min})/(r_{i,max} - r_{i,min})$
Store the data with the same grouping criterion X_i^g in a single file
Slice the data in X_i^g into $\{s_1^i, s_2^i, ..., s_c^i\}$ with the same time step c, while maintaining the temporal sequence order of the datasets
if the volume of X_i^g exceeds the time step c **then**
    continue to slice the remaining data
end if
if the volume is less than c **then**
    discard the data
end if
return Output the structured data $\{s_1^i, s_2^i, ..., s_c^i\}$ with the same time step c

This ensures that, even in the unfortunate event of interception by malicious entities, the original model weights remain inscrutable, effectively mitigating the risk of back-tracing and subsequent privacy compromises from the vehicle's data.

In subsequent sections, we delve into the intricacies of each component comprising our intrusion detection system, providing a comprehensive understanding of its operation and privacy-preserving capabilities.

5.3.1. Intrusion Detection Model Design

After processing data as described in Section 5.2.1 and 5.2.2, we obtain structured training data $\{s_1^i, s_2^i, ..., s_c^i\}$. Utilizing the bidirectional temporal sequence model introduced in Section 4.1, we take each feature $r_{i,x}, x \in [1, m]$ from D_i and combine them with the previous time step's hidden states $\overleftarrow{H}_{t-1}$ and $\overrightarrow{H}_{t-1}$ as inputs to the Bidirectional Gated Recurrent Unit (Bi-GRU) network. At each time step, the network generates a hidden state $\overleftarrow{H}_t$ and $\overrightarrow{H}_t$, along with an output O_t for each time step.

In contrast with traditional Seq2Seq models, we apply additive attention to integrate the output results O_t at each time step with the hidden states $\overleftarrow{H}_t$ and $\overrightarrow{H}_t$, using the additive attention mechanism proposed in Section 4.1 to calculate $S^i, S^i \in \mathbb{R}^{n \times h}$, where n is the size of the time slice and h is the size of the hidden layer.

The final detection result is computed as:

$$\begin{aligned} Re^i = Dropout(ReLu(S^i W_s + S_b^i)) \\ + ReLu(S^i W_s + S_b^i) \end{aligned} \tag{6}$$

And the loss function is $Loss(f_\theta(X_i), Y_i)$. Here, the *Dropout* function serves as a regularization technique in neural networks to prevent overfitting, as described in the literature. The term Re^i denotes the model's final detection result, where $Re^i \in \mathbb{R}^{b \times 2}$ and b represents the batch size used in model training. The weight matrix $W \in \mathbb{R}^{\Omega \times 2}$ corresponds to $S^i \in \mathbb{R}^{h \times \Omega}$, with h indicating the size of the hidden layer and Ω the output layer size of the multilayer perceptron. For different vehicle nodes C_i, the model iteratively updates the weight parameters until the loss is minimized.

5.3.2. Model Weight Parameters Noise Perturbation

During the gradient update process of the intrusion detection model, we incorporate Laplace random noise into the optimization of model parameters using gradient descent, ensuring the entire process adheres to differential privacy. To demonstrate that the random perturbation z applied to gradient descent complies with the differential privacy requirements proposed in the Section 4.3, we select a random function $A(D)$. This function is applied to both the original dataset D and a differentially private version D' to produce random values and to calculate the probability distribution. We also define $A(D) = f(D) + x$ as the noise-added function for the dataset.

$$\begin{aligned} \frac{Pr[A(D) = t]}{Pr[A(D') = t]} &= \frac{Pr[A(D) + x = t]}{Pr[A(D') + x = t]} \\ &= \frac{Pr[x = t - A(D)]}{Pr[x = t - A(D')]} \\ &= \frac{\frac{1}{2\beta} exp[-\frac{|t - f(d)|}{\beta}]}{\frac{1}{2\beta} exp[-\frac{|t - f(d')|}{\beta}]} \\ = exp[\frac{|t - f(d')| - |t - f(d)|}{\beta}] \\ &\leq exp[\frac{|f(d) - f(d')|}{\beta}] \\ &\leq exp(\frac{\Delta s}{\beta}) \end{aligned} \tag{7}$$

By substituting the Laplace distribution function with mean $\mu = 0$ into Eq. 7, we obtain the inequality $\leq \exp(\Delta f/\beta)$ using the triangle inequality $|a| - |b| \leq |a - b|$. Our proof reveals the relationship between the sensitivity Δf and the random noise β, which is $\beta > \Delta f/\epsilon$. Therefore, by controlling the ratio of sensitivity Δf and β to be less than ϵ, we can ensure the definition of differential privacy. Hence, in the proof of Eq. 7, we demonstrate that incorporating local differential privacy in the training process of the federated learning intrusion detection model can satisfy the definition of differential privacy.

Here is our proposed Algorithm 2 for local differential privacy federated learning model training. To avoid the problem of the learning rate α being too large in the local model training process, which may prevent the model from converging, we multiply the learning rate in each local training by 0.95 of the previous learning rate α_{t-1}. At the

Algorithm 2 Incorporating LDP in the training process of the federated learning intrusion detection model algorithm

Initializes the global weight parameters as θ_0
Initializes clients and communication rounds and learning rate
for t from 1 to communication rounds **do**
  local clients ← the server randomly selects some clients
  for client i in local clients **do**
    weight parameters, learning rate← client i **LocalTrain**(learning rate, θ_t)
    sum weight parameters ← cumulative weight parameters
  end for
  θ_t ← average of sum parameters
  learning rate ← learning rate × 0.95
end for
LocalTrain(learning rate, θ_t):
new θ_t ← Update the local model weight parameters according to the local data ,learning rate and θ_t
new θ_t with noise ← Adding noise perturbations to the weighting parameters using LDP

Table 4
Datasets Details for Intrusion Detection Model Data

Dataset	Total Number	Attack	Normal
UNSW-NB 15[26]	2,540,047	321,283	2,218,764
CAN-intrusion datasets [20]	4,613,909	2,244,041	2,369,868
CIC-IDS-2017[31]	2,830,773	2,273,097	557,676

same time, to ensure the efficiency of the algorithm, we only calculate the gradient for a batch of samples in the dataset D_i during each iteration.

6. Experimentation

In this section, we evaluate the effectiveness of our proposed intrusion detection model using three real-world datasets.

6.1. Datasets

We utilized the CAN-Intrusion-datasets [20] which records automobile hacking data, as well as the UNSW-NB 15 [26] and CIC-IDS-2017 [31] datasets which document the communications between OBUs and RSUs. These datasets are used to validate our research since they cover the latest attack data and serve as ideal resources for detecting contemporary threats. Three datasets were employed in our evaluations: UNSW-NB 15, CAN-Intrusion-datasets, and CIC-IDS-2017. Table 4 provides the detailed parameters of the datasets. To make the data more comprehensive and balanced, we collected and shuffled the data randomly after feature extraction for each client, aiming for a 1:1 ratio between normal and attack samples as much as possible. These datasets contain various types of attacks, covering normal and attack records of both CAN and external-network traffic, obtained by simulating automobile hacking attacks and NIDS testing platforms.

For experimental validation of intrusion detection in the in-vehicle network, the dataset used in this study is the CAN-intrusion-dataset [20] proposed in 2018 . This dataset was generated based on CAN communication traffic recorded through the OBD-II port of vehicles during CAN attacks. The features of the dataset mainly include timestamps, CAN ID, data length code (DLC), and the 8-byte data fields (DATA[0]-DATA[7]) in CAN frames. To facilitate effective model learning, we devised a processing algorithm for the DATA fields as shown in Algorithm 3, which converted the raw data fields into numeric features suitable for collaborative model training.

In the experimental validation of external network IDS, public IoV benchmark datasets have significant deficiencies due to issues like popularity, privacy protection, and commercialization. On the other hand, WLAN and cellular networks are the mainstream technologies for IoV and commercial vehicle communications. Therefore, attack methods against conventional computer networks can be considered to bear similarities with intrusions against external-vehicle networks. In light of this situation, many research works [8, 5, 19, 41] have adopted universal network security datasets to develop IDS for external-vehicle networks, including KDD-99, NSL-KDD, Kyoto 2006+, UNSW-NB 15, and CIC-IDS-2017. Among these network security datasets, UNSW-NB 15[26] and CIC-IDS-2017[20] are regarded as the most representative datasets in the current external-vehicle network IDS field, owing to not only their technical advancement, but also the larger numbers of features, instances, and network attack types they contain compared to other datasets. In the validation process of our proposed external/in-vehicle network intrusion detection algorithm, we specifically chose the network communication data in the UNSW-NB 15 and CIC-IDS-2017 datasets to simulate the complex external-vehicle network environment. In this way, the performance of our algorithm in real-world IoV contexts can be evaluated more effectively.

All network communication messages are stored in CSV files along with timestamps. These datasets consist of diverse attack types including DoS, reconnaissance and injection attacks, gear spoofing, RPM spoofing, and fuzzing attacks. In our investigation, we recognized that any form of attack, once successfully executed within the vehicular network, could lead to grave consequences for both the driver and the vehicle. Consequently, in this paper, we treat all types of attacks as generic aggressive actions without distinguishing them into specific categories.

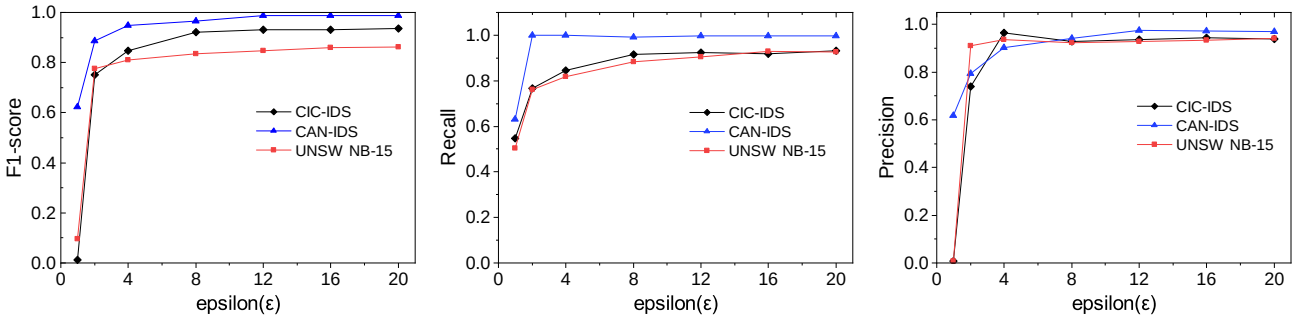


Figure 4: Utility Comparison of detection model with different epsilon on three datasets

Algorithm 3 The field of *DATA* processing algorithm for generating numerical features

Input: data, an array of CAN bus data with a length of 8 bytes

Output: *data label*, an array of normalized CAN bus features with a length of 8 bytes

can data bytes ← an array of CAN bus data obtained from data[-1] after removing any leading or trailing white space and then splitting the string using space as the delimiter
if the length of *can data bytes* is greater than 8 **then**
  set *can data bytes* to the first 8 bytes of *can data bytes*
end if
if the length of *can data bytes* is equal to 1 and *can data bytes*[0] is an empty string **then**
  set *can data bytes*[0] to 'FF' and set dlc to 1
end if
for *can byte* in *can data bytes* **do**
  append the corresponding normalized value (i.e., int(*can byte*, 16) / 255) to *data label*
end for
for i in the range $[dlc, 8]$ **do**
  append 1.0 to *data label* to ensure that the length of *data label* is always 8 bytes
end for
for i in the range $[dlc, 8]$ **do**
  append 1.0 to *data label* to ensure that the length of *data label* is always 8 bytes
end for
return *data label*

6.2. Evaluation Metrics

To evaluate the performance of our models, we use terms such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). We employ precision, recall, and F1 score.

Accuracy is the metric that quantifies the ratio of correctly predicted observations to the total observations in the dataset.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Precision measures the proportion of correctly predicted positive samples to all samples predicted as positive.

$$Precision(P) = \frac{TP}{TP + FP} \tag{9}$$

Similarly, recall is the proportion of correctly predicted positive samples to all actual positive samples.

$$Recall(R) = \frac{TP}{TP + FN} \tag{10}$$

The F1 score is a harmonic mean of precision and recall.

$$F1\ score = 2 \cdot \frac{P \cdot R}{P + R} \tag{11}$$

Since precision and recall affect and constrain each other, we use the F1 score to balance these estimates and evaluate the model's overall performance with a single value.

A. Federated Learning Training B. Vehicle-level Testing

Figure 5: Experimental Setup

6.3. Experimental Setup

As shown in Fig. 5, the algorithms proposed in this paper were implemented and executed on the Ubuntu/Linux 18.04 operating system. The code was written in Python 3.10, and we utilized the PyTorch 1.13 deep learning framework's API to construct a deep neural network for the IDS detection model. From Fig. 5A, model training was conducted on a platform equipped with an Nvidia RTX 3060Ti graphics card, which also featured a 2.9 GHz octa-core Intel Core

Table 5
Federated Learning for Training Model Parameters and Hyperparameters

Round number	150
Client number	100
Number of clients selected for a round	10
Local clients batch size	10
Local clients epoch	5
Learning rate	0.015
Learning rate scheduler	0.95
optimization function	Adam
loss function	Cross Entropy Loss

i7 10700F CPU and 32GB RAM. From Fig. 5B, vehicle-level model testing was carried out on a Raspberry Pi 4B embedded device, which is equipped with a 64-bit dual-core Cortex-A72 CPU and 4GB of memory. Each model was trained on the GPU using the Adam optimizer and underwent 150 rounds of federated learning.

Each federated learning client performed 10 rounds of local training. The initial learning rate of the model was set to 0.015, and after each training round, the learning rate of each client was multiplied by 0.95 to decay the rate. Detailed model parameters and hyperparameter configurations are provided in Table 5. For the privacy budget $\varepsilon, \varepsilon \in [1, 2, 4, 8, 12, 16, 20]$, we consider a range of values, where lower values of privacy budget ε provide stronger privacy protection for vehicles.

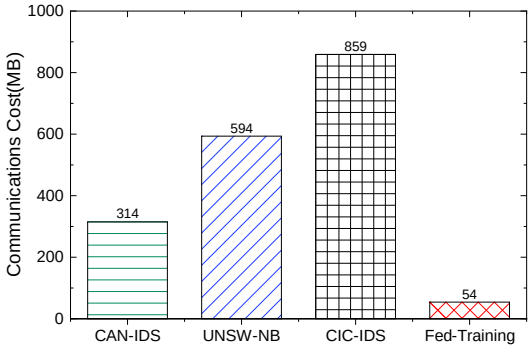


Figure 6: Communication cost in Federated Learning and Central Learning with different data set

7. Evaluation and Discussion

7.1. Privacy Analysis

In Section 5.3.2, we theoretically derived the Laplace noise mechanism perturbation in intrusion detection models. In this section, we demonstrate the performance of the noise distribution Laplace perturbation on various metrics of the proposed detection model. We use the Laplace mechanism to train 70% of the dataset for training and 30% for testing, while the privacy budget for local differential privacy is [1, 2, 4, 8, 12, 16, 20].

From in Fig. 4, this paper demonstrates the detection metrics of the proposed detection model across different privacy budget ε values and various datasets. When we set the privacy budget ε of our model to 4, we found that both the detection performance and privacy are effectively protected. If the privacy budget ε is set below 4, although the privacy protection effect is optimal, the model's detection metrics significantly decline. On the other hand, when the privacy budget ε exceeds 4, the detection metrics of the model tend to stabilize, indicating that the effectiveness of privacy protection becomes negligible. Therefore, based on the experimental results, we determine that the optimal privacy budget ε for our model is 4.

7.2. Noise distribution evaluation

As shown in Fig. 7, we present the pattern of loss values during the entire process of training 100 local models. Although we observe a decreasing trend in the loss values, they do not consistently decrease. This is because the local models are randomly selected to participate in the training, and the learning rate of the local models decreases with the increase of their participation times. Therefore, the loss values exhibit periodic increases. From the figure, we can observe a decrease in loss values compared to the previous period.

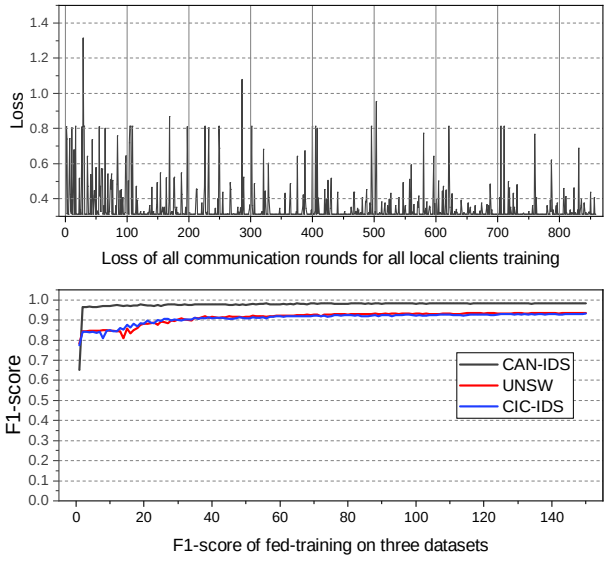


Figure 7: Loss value of local detection model involved in training and F1-score of 150 Fed-training communication rounds on three datasets

During the local training process, we used three publicly available datasets and set the differential privacy budget ε to 20. As shown in Fig. 7, we found that the F1-score of the federated learning becomes stable and no longer fluctuates after more than 60 rounds. The detailed model detection metrics of the three datasets are shown in Table 6. Therefore, we conclude that the proposed model meets our design goals: it can protect the weight parameters of the local models while

Table 6
Ablation Study of Modules on Detection Model

	CAN-IDS				UNSW-NB				CIC-IDS			
Modules	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec
Bi-LSTM w/o att, w/ slice	0.9516	0.9538	0.9340	0.9698	0.9558	0.9741	0.9838	0.9294	0.9448	0.9643	0.9704	0.9204
Bi-LSTM w/o att, w/o slice	0.9308	0.9294	0.9091	0.9535	0.9295	0.9603	0.9388	0.9204	0.9380	0.9631	0.9464	0.9300
LSTM w/o att, w/o slice	0.8682	0.8790	0.8168	0.9265	0.8834	0.9259	0.9182	0.8512	0.8724	0.9423	0.9642	0.8806
Bi-GRU w/ att, w/ slice	0.9749	0.9775	0.9652	0.9848	0.9551	0.9700	0.9730	0.9378	0.9641	0.9739	0.9797	0.9490
Bi-LSTM w/ att and slice	0.9851	0.9856	0.9726	0.9979	0.9751	0.9649	0.9768	0.9733	0.9789	0.9694	0.9780	0.9798

having the ability to train detection models through federated learning, and the model protection utility is not reduced.

7.3. Results of Performance Comparison Between Federated and Centralized Model

In the IoV environment, the local clients (vehicle-side) communicate wirelessly through 5G or 802.11p technology to participate in the training process of the federated learning model. The communication cost of federated learning model training is the communication traffic cost of uploading or downloading model parameters when the local vehicle interacts with the RSU throughout the training process. In contrast, the communication cost of centralized training is the cost associated with uploading all local data to the central server during vehicle operation (eg, window opening command, brake command). We compare these communication costs in terms of the data size exchanged between the local vehicle and the RSU.

Fig. 6 shows that during central training, the communication cost is dependent on the size of the dataset that the client is trained on. If the dataset is large, it leads to high communication cost consumption. In contrast, distributed federated learning training is not impacted by dataset size, as only the model weights are uploaded each time. In our experiments, we set the upload cost for a single round of federated training to be the sum of all local model sizes and the size of the dataset attributes that need to be uploaded. The download cost is then set to the sum of the global model sizes obtained by all local vehicle entities.

We recorded the total size of data exchanged between all vehicle entities and the RSU during the entire communication cycle (150 rounds) for the federated learning communication cost calculation method $cost = \sum_{i}^{round} W_s(i)$, where $round$ is the communication cycle, and W_s is the model weights bit size. On the other hand, the communication cost of centralized training is set to the total size of the datasets. This is because central training needs to obtain local data of all vehicles through the RSU and the central server, and these two datasets store all local data of vehicles.

Our experiments show that the total amount of data communicated in federated training is significantly smaller than that in central training methods.

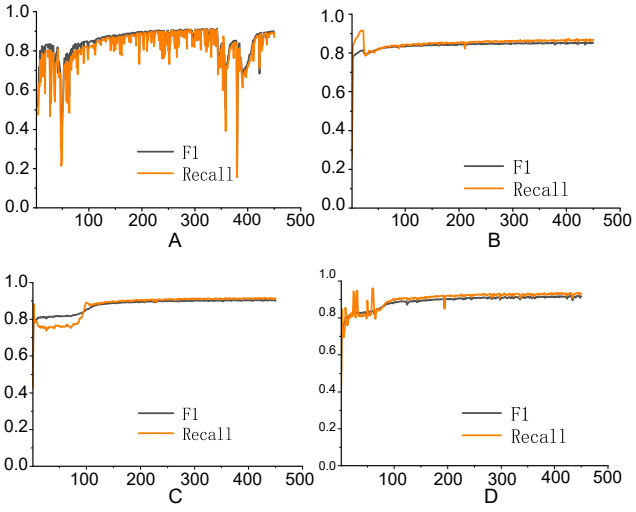


Figure 8: Model training process for experimental validation of the *homogeneity property*. A: LSTM without slice and addictive-attention, B:LSTM with slice without addictive-attention, C: LSTM with slice and addictive-attention, D:Bi-LSTM with slice and addictive-attention.

7.4. The homogeneity property validation experiment

Fig. 8 depicts the results of four experiments designed to validate the effectiveness of the homogeneity property of user behavior as proposed in our study. Experiment A (Fig. 8A) involves capturing 64 records at fixed intervals without applying the homogeneity property for grouping or slicing the dataset. To assess the impact of potential temporal fluctuations and to examine the necessity of the attention mechanism, control experiments were also performed without the attention component in both Experiment A (Fig. 8A) and Experiment B (Fig. 8B).

While Experiment B (Fig. 8B) incorporated the dataset processed with grouped binning according to the homogeneity property, it was evident that Experiment A's (Fig. 8A) training exhibited considerable fluctuations and struggled

with model convergence, leading to instability. In stark contrast, Experiments C and D (Fig. 8C and Fig. 8D), which utilized the grouped slice dataset processing algorithm aligned with the homogeneity property outlined in this paper and implemented a revised temporal model, demonstrated stable convergence after 200 training iterations with no significant jitter as seen in Experiment A (Fig. 8A). From these observations, it is inferred that the homogeneity property we introduced significantly bolsters the training stability and fosters more effective model convergence.

7.5. Ablation Study of Modules

In order to verify whether the model modules proposed in this paper are necessary or redundant, this study primarily evaluates the effectiveness of the time-series algorithm based on the time attention mechanism in Section4.1.2, the behavioral data processing algorithm based on the Homogeneity Property in Section 5.2.1 and 5.2.2, and other time-series models through a series of ablation studies. To eliminate potential experimental errors, we systematically removed different components and conducted controlled experiments, repeating each experiment five times to reduce the impact of environmental variations and selecting the median accuracy as the result.

As shown in Table 6, we replaced or discarded different modules of the model proposed in this paper to test their effectiveness on three datasets in application detection tasks. For instance, in the first row of Table 6, after grouping and slicing the datasets using the data processing algorithm proposed in this article, we utilized Bi-LSTM for time series recognition while removing the attention module. In contrast, in the fifth row, we used the same time series model but retained the time attention mechanism, demonstrating that the detection metrics of the model on different datasets were superior to those without the attention mechanism.

On the other hand, Fig. 8 reveals significant instability in the training process when the Homogeneity Property is not applied, supporting the non-redundancy of incorporating data aggregation and processing algorithms, thereby enhancing the model's effectiveness and reducing the possibility of component redundancy. Additionally, both Bi-GRU and Bi-LSTM exhibited comparable performance across all metrics, highlighting the robustness of bidirectional time series models in capturing contextual features of user behavior. It was also observed that without the synergistic effect of the attention mechanism, unidirectional models proved to be insufficient, thus highlighting the superiority of bidirectional models. Considering these observations, Bi-LSTM has emerged as an outstanding time series model, surpassing similar models in detection effectiveness.

Furthermore, the third row of the table opts not to use Bi-LSTM but instead uses the LSTM time series model, and does not employ the the Homogeneity Property for grouping the datasets. We found that without using Bi-LSTM and the homogeneity property binning algorithm, the model's detection results in terms of F1-score significantly dropped.

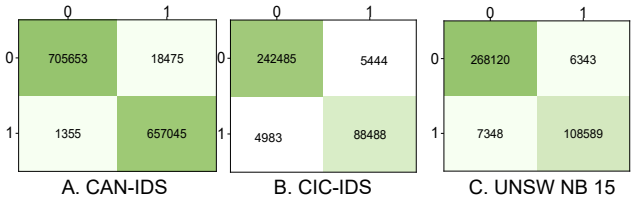


Figure 9: Confusion Matrix for the Three Test Datasets

7.6. Comparisons With The State of The Art Models

In the field of IoV intrusion detection, this study selected two categories encompassing six representative state-of-the-art research works as benchmarks for evaluation. We conducted a comprehensive comparison and analysis of our FDL-IDM against these existing algorithms to highlight the contributions of our research.

1) Within IoV intrusion detection research, only FedMix [43] accounted for privacy and communication costs, yet it had deficiencies in model upload and server-side model aggregation strategies, leading to reduced accuracy. Other studies focused on enhancing detection accuracy, neglecting model privacy and communication challenges. This paper, set against the backdrop of real-world IoV scenarios, reduces the risk of model reverse engineering attacks by perturbing the parameters of the uploaded models.

2) Ayodeji[27] and Almutlaq[4] utilized explainable neural networks for intrusion detection, achieving high accuracy and detection efficiency in literature, particularly suitable for resource-constrained IoV environments. Our experiments revealed that explainable neural networks currently struggle with learning complex heterogeneous data, and potential attacks are not detected effectively.

3) Wang[38] processed data through behavioral analysis and transformed it into spatiotemporal state matrices using Convolutional Neural Networks (CNNs), effectively defending against joint attacks with impressive detection performance. However, the detection model used is not yet optimized; adding noise as described in the literature would lead to significant performance fluctuations. In contrast, our improved temporal sequence model effectively avoids accuracy degradation due to privacy protection measures, an issue also present in[32, 41].

4) The studies in[32, 41], while using deep learning for intrusion detection and recognizing attacks, did not sufficiently address the communication burden and privacy leakage on the client (vehicle-side). Our FDL-IDM enhanced these strategies with federated learning incorporating differential privacy perturbations, demonstrating reduced communication costs compared to centralized training, as shown in Fig. 6, with the benefit being independent of dataset size.

In Table 8, we compared the performance differences between the proposed FDL-IDM model and other state-of-the-art models trained on public datasets. Although FDL-IDM did not rank first in accuracy and F1 score, it still performed at a leading level with a score of 0.9851 and confusion matrix

Table 7
Performance Comparison with State-of-The-Art Models

Model	F1	Rec	Prec	Acc
Sun [32]	0.805-0.967	0.857-0.971	0.859-0.962	-
Almutlaq[4]	0.922-0.9632	0.9142-0.9746	0.8942-0.9832	-
Wang [38]	0.8580-0.9713	-	0.9705-0.9780	0.8720-0.9715
Ayodeji[27]	0.9883	0.9915	0.9910	0.9915
Yang [41]	0.99895	-	-	-
Zhao [43]	0.913	0.851	0.985	-
Proposed Model	**0.9751-0.9851**	**0.9733-0.9979**	**0.9726-0.9780**	**0.9649-0.9856**

Table 8
Vehicle-Level Testing on Different Datasets

	CAN-IDS				UNSW-NB				CIC-IDS				
Num	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec	Avg(ms)
1	0.9826	0.9834	0.9777	0.9876	0.9762	0.9828	0.9867	0.9658	0.9582	0.9754	0.9844	0.9335	6-8
2	0.9829	0.9846	0.9698	0.9964	0.9785	0.9856	0.9955	0.9620	0.9608	0.9772	0.9854	0.9373	5-7
3	0.9719	0.9878	0.9565	0.9878	0.9669	0.9845	0.9462	0.9886	0.9761	0.9882	0.9827	0.9695	6-9

in Fig. 9. Additionally, by analyzing the data in Table 1, we found that current leading IoV intrusion detection models generally lack measures to protect model security, with few models being trained using federated learning optimized for IoV environments combined with differential privacy. Given these results, we conclude that the FDL-IDM model is highly suitable for IoV intrusion detection tasks and can effectively perform its functions.

7.7. Vehicle-Level Model Evaluation

The proposed FDL-IDM trained model was tested on the onboard computer Raspberry Pi 4B to evaluate its practical usability in IoV environments. Furthermore, the generalization performance of the model can be further assessed by applying it to a test set that was not used for training.

The experimental results on the test set are shown in Table 8, as indicated, the F1 scores of the proposed FDL-IDM on three datasets ranged between 0.9669 and 0.9829. In addition, the confusion matrices for evaluating the proposed method on different dataset test sets are shown in Fig. 9. The primary reason for achieving high accuracy in CAN intrusion detection is the clear distinction between the attack patterns and normal patterns in the DATA payload of the CAN-IDS.

In Fig. 9B and Fig. 9C, we observed that the model also achieved a low false-positive rate in external-vehicle network intrusion detection, mainly due to training on a large dataset. The use of more data samples has enhanced the generalization ability of the proposed method. Additionally, the data processing algorithm was designed with driving behavior characteristics in mind, filtering out irrelevant and misleading features that could cause overfitting, thus further improving the model's generalization performance.

On the other hand, to ensure that the proposed FDL-IDM can be deployed in real vehicle systems and meet the real-time requirements of vehicle safety services, the alarm generation time for each data packet transmitted over the IoV should not exceed 10 milliseconds [2]. As shown in the table, experiments were conducted on three different datasets, and the results showed that the detection time for each packet was maintained within 10 milliseconds, thus fully meeting the real-time requirements.

Practical Application Discussion and Future Work

In this section, we explore the multifunctionality of the model when deployed in IoV systems. As depicted in Fig. 5, we present the workflow for training our proposed detection model and testing it on vehicle-level devices. Before the model workflow begins, the central server utilizes its computational advantage to initialize training of the model. After the central server initializes the model, it is delivered to vehicle-side devices, where local data is used for further training and testing. The trained model is then sent back to the central server after being perturbed using differential privacy techniques, where it is aggregated into a global model. Subsequently, individual vehicles download and decrypt this aggregated model for integration into their onboard systems.

Once deployed in the in-vehicle network environment, the model can help detect potential cyber threats in real-time from the various ECUs connected to the CAN bus, such as rpm, power windows, and brake indicators. When deployed in the external vehicle network environment, the model helps monitor network threats from external communications like DSRC, WiFi, and Bluetooth. The entire process reflects the complex co-training simulation environment between vehicles and the central server. Our proposed solution aims to be deployed in a way that meets the practical needs of modern IoV systems.

In the experiments of Section 7.3, compared to traditional training on the central server, transferring training data between vehicle-side and central server led to communication cost directly correlating with training dataset size. As shown in Fig. 6, communication cost reached 314MB when using the CAN-IDS dataset. However, the federated learning training method proposed in this paper has overhead between vehicle-side and central server independent of the central server, with stable communication cost around 54MB. In addition, we evaluated the performance of our proposed model at the vehicle level, through a series of real tests and extensive simulations, aiming to validate the performance of our detection model in terms of real-time operation and efficacy. In Section 7.7, we selected test data from the three datasets that was not involved in training, and conducted three experiments on each dataset. Table 7 shows the detection metrics and inference times of FDL-IDM on different datasets, with F1 scores controlled around 97% and inference times within 10 milliseconds. These results demonstrate the stability and real-time performance of our proposed model when facing different scenarios.

However, the vehicle-level experiments in this paper were based on the Raspberry Pi 4B embedded device. With the development of the automotive industry, the computing power of most current vehicles has exceeded that of the device used in the experiments. Nevertheless, there are still many vehicles whose computing power has not reached the level of the experimental device. Therefore, after deploying the model proposed in this paper, the model's response time may not meet the strict requirements of real-time applications. To protect data privacy during training, differential privacy techniques based on the Laplace mechanism were used to perturb the uploaded gradients. The current differential privacy strategy does not perform bounded processing of the noise, which sometimes causes excessive noise leading to decreased model accuracy, thus slowing down training convergence and prolonging training time.

As a direction for future work, we aim to enhance our framework for more effective identification of cyber threats in IoV networks. By reviewing the latest research literature[3, 9], we find that federated learning still has significant research gaps in handling large-scale data and data poisoning attacks, with relatively scarce related research. To this end, we plan improvements in two aspects: first, during federated learning training, we intend to sparsify the uploaded model gradients, i.e., upload only gradients of significant importance, while the central server focuses on aggregating these key gradients. This method can not only effectively reduce bandwidth usage and privacy budget consumption, but also adapt to vehicles with weaker computing power. Secondly, in the implementation of differential privacy perturbation, we plan to introduce bounded noise to avoid strongly interfering noise, thereby accelerating model convergence and enhancing the robustness of the detection model. Through implementing these strategies, we expect to improve the applicability of federated learning systems in IoV cybersecurity, demonstrating higher efficiency and security in real-world deployments. Meanwhile, these improvements also aim to provide new solutions and research methodologies for researchers and academia in the same field.

Conclusion

In this paper, we propose a federated deep learning algorithm that uses behavioral analysis, distributed training, and Laplace noise mechanism perturbations to enhance the accuracy and privacy of the detection model. Our method groups data based on the source address and uses an improved temporal sequence model with additive attention mechanisms to enhance behavioral features. Federated learning training and Laplace perturbations during local training processes ensure privacy and improve the accuracy of the model.

Although this study has tested the proposed model on vehicle-level embedded devices, the data utilized still comes from public datasets, which diverges somewhat from real-world scenarios. In our future work, we plan to implement sparsification of the model gradients uploaded to the central server by only transmitting gradients that are significantly important. This approach will allow the central server to focus on the aggregation of these key gradients. Additionally, we intend to apply boundary constraints to the noise, in order to avoid the generation of highly disruptive noise. These measures are expected to accelerate the convergence speed of the model and enhance the robustness of the detection model.

CRediT authorship contribution statement

Rui Chen: Conceptualization, Methodology, Software, Investigation, Formal Analysis, Writing - Original Draft. **Xiaoyu Chen:** Methodology, Software, Investigation, Data Curation, Writing - Original Draft. **Jing Zhao:** Conceptualization, Review and Editing, Supervision.

ACKNOWLEDGMENTS

We would like to sincerely thank the editors and anonymous reviewers for their helpful comments.

Data availability

The implemented code used to support the findings of this study is available from the corresponding author upon request. The datasets used in this paper are publicly available for download.

Declaration of Interest Statement

The author declares no competing interests

References

[1] Abdelmoumin, G., Rawat, D.B., Rahman, A., 2021. On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things. IEEE Internet of Things Journal 9, 4280–4290.

[2] Abualhoul, M.Y., Shagdar, O., Nashashibi, F., 2016. Visible light inter-vehicle communication for platooning of autonomous vehicles, in: 2016 IEEE Intelligent Vehicles Symposium (IV), IEEE. pp. 508–513.

[3] Aljanabi, M., Ahmad, H., 2023. Navigating the void: Uncovering research gaps in the detection of data poisoning attacks in federated learning-based big data processing: A systematic literature review. Mesopotamian Journal of Big Data 2023, 149–158.

[4] Almutlaq, S., Derhab, A., Hassan, M.M., Kaur, K., 2022. Two-stage intrusion detection system in intelligent transportation systems using rule extraction methods from deep neural networks. IEEE Transactions on Intelligent Transportation Systems .

[5] Aloqaily, M., Otoum, S., Al Ridhawi, I., Jararweh, Y., 2019. An intrusion detection system for connected vehicles in smart cities. Ad Hoc Networks 90, 101842.

[6] Anbalagan, S., Raja, G., Gurumoorthy, S., Suresh, R.D., Dev, K., 2023. Iids: Intelligent intrusion detection system for sustainable development in autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems .

[7] Ashraf, J., Bakhshi, A.D., Moustafa, N., Khurshid, H., Javed, A., Beheshti, A., 2020. Novel deep learning-enabled lstm autoencoder architecture for discovering anomalous events from intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems 22, 4507–4518.

[8] Aswal, K., Dobhal, D.C., Pathak, H., 2020. Comparative analysis of machine learning algorithms for identification of bot attack on the internet of vehicles (iov), in: 2020 International Conference on Inventive Computation Technologies (ICICT), IEEE. pp. 312–317.

[9] Cui, J., Sun, H., Zhong, H., Zhang, J., Wei, L., Bolodurina, I., He, D., 2023. Collaborative intrusion detection system for sdvn: A fairness federated deep learning approach. IEEE Transactions on Parallel and Distributed Systems .

[10] Duchi, J.C., Jordan, M.I., Wainwright, M.J., 2013. Local privacy and statistical minimax rates, in: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, IEEE. pp. 429–438.

[11] Erlingsson, Ú., Pihur, V., Korolova, A., 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response, in: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, pp. 1054–1067.

[12] Fredrikson, M., Jha, S., Ristenpart, T., 2015. Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pp. 1322–1333.

[13] Gao, S., Zhang, L., He, L., Deng, X., Yin, H., Zhang, H., 2023. Attack detection for intelligent vehicles via can-bus: A lightweight image network approach. IEEE Transactions on Vehicular Technology .

[14] Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks 18, 602–610.

[15] Haddaji, A., Ayed, S., Fourati, L.C., 2024. A novel and efficient framework for in-vehicle security enforcement. Ad Hoc Networks 158, 103481.

[16] Hajimaghsoodi, M., Jalili, R., 2022. Rad: A statistical mechanism based on behavioral analysis for ddos attack countermeasure. IEEE Transactions on Information Forensics and Security 17, 2732–2745.

[17] Hatamizadeh, A., Yin, H., Molchanov, P., Myronenko, A., Li, W., Dogra, P., Feng, A., Flores, M.G., Kautz, J., Xu, D., et al., 2023. Do gradient inversion attacks make federated learning unsafe? IEEE Transactions on Medical Imaging .

[18] Injadat, M., Moubayed, A., Nassif, A.B., Shami, A., 2020. Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management 18, 1803–1816.

[19] Kumar, R., Kumar, P., Tripathi, R., Gupta, G.P., Kumar, N., 2021. P2sf-iov: A privacy-preservation-based secured framework for internet of vehicles. IEEE Transactions on Intelligent Transportation Systems 23, 22571–22582.

[20] Lee, H., Jeong, S.H., Kim, H.K., 2017. Otids: A novel intrusion detection system for in-vehicle network by using remote frame, in: 2017 15th Annual Conference on Privacy, Security and Trust (PST), IEEE. pp. 57–5709.

[21] Li, C., Wu, M., Liu, Y., Zhou, K., Zhang, J., Sun, Y., 2022a. Ss-lru: a smart segmented lru caching, in: Proceedings of the 59th ACM/IEEE Design Automation Conference, pp. 397–402.

[22] Li, X., Yan, H., Cheng, Z., Sun, W., Li, H., 2022b. Protecting regression models with personalized local differential privacy. IEEE Transactions on Dependable and Secure Computing .

[23] Longari, S., Valcarcel, D.H.N., Zago, M., Carminati, M., Zanero, S., 2020. Cannolo: An anomaly detection system based on lstm autoencoders for controller area network. IEEE Transactions on Network and Service Management 18, 1913–1924.

[24] Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: Proc. Adv. Neural Inf. Process. Syst.,I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, pp. 4765–4774.

[25] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR. pp. 1273–1282.

[26] Moustafa, N., Slay, J., 2015. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 military communications and information systems conference (MilCIS), IEEE. pp. 1–6.

[27] Oseni, A., Moustafa, N., Creech, G., Sohrabi, N., Strelzoff, A., Tari, Z., Linkov, I., 2022. An explainable deep learning framework for resilient intrusion detection in iot-enabled transportation networks. IEEE Transactions on Intelligent Transportation Systems .

[28] Rajapaksha, S., Kalutarage, H., Al-Kadri, M.O., Petrovski, A., Madzudzo, G., Cheah, M., 2023. Ai-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys 55, 1–40.

[29] Rani, P., Sharma, C., Ramesh, J.V.N., Verma, S., Sharma, R., Alkhayyat, A., Kumar, S., 2023. Federated learning-based misbehaviour detection for the 5g-enabled internet of vehicles. IEEE Transactions on Consumer Electronics .

[30] Shan, Y., Yao, Y., Zhou, X., Zhao, T., Hu, B., Wang, L., 2023. Cfl-ids: An effective clustered federated learning framework for industrial internet of things intrusion detection. IEEE Internet of Things Journal .

[31] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp 1, 108–116.

[32] Sun, H., Chen, M., Weng, J., Liu, Z., Geng, G., 2021. Anomaly detection for in-vehicle network using cnn-lstm with attention mechanism. IEEE Transactions on Vehicular Technology 70, 10880–10893.

[33] Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. Advances in neural information processing systems 27.

[34] Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T., 2016. Stealing machine learning models via prediction apis., in: USENIX security symposium, pp. 601–618.

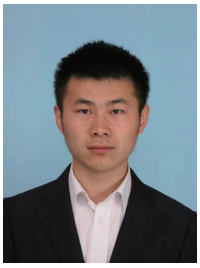
[35] Van Wyk, F., Wang, Y., Khojandi, A., Masoud, N., 2019. Real-time sensor anomaly detection and identification in automated vehicles. IEEE Transactions on Intelligent Transportation Systems 21, 1264–1276.

[36] Vinita, L.J., Vetriselvi, V., 2023. Federated learning-based misbehaviour detection on an emergency message dissemination scenario for the 6g-enabled internet of vehicles. Ad Hoc Networks 144, 103153.

[37] W. Wu, R. Li, G.X., 2019. A survey of intrusion detection for in-vehicle networks. IEEE Transactions on Intelligent Transportation Systems 21, 919 – 933.

[38] Wang, L., Zhang, X., Li, D., Liu, H., 2023. Multi-sensors space and time dimension based intrusion detection system in automated

vehicles. IEEE Transactions on Vehicular Technology .
[39] Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q., Poor, H.V., 2020. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security 15, 3454–3469.
[40] Xiao, S., Ge, X., Han, Q.L., Zhang, Y., 2021. Secure distributed adaptive platooning control of automated vehicles over vehicular ad-hoc networks under denial-of-service attacks. IEEE Transactions on Cybernetics 52, 12003–12015.
[41] Yang, L., Moubayed, A., Shami, A., 2021. Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles. IEEE Internet of Things Journal 9, 616–632.
[42] Zhang, H., Zeng, K., Lin, S., 2023. Federated graph neural network for fast anomaly detection in controller area networks. IEEE Transactions on Information Forensics and Security 18, 1566–1579.
[43] Zhao, J., Wang, R., 2022. Fedmix: A sybil attack detection system considering cross-layer information fusion and privacy protection, in: 2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE. pp. 199–207.
[44] Zhao, Y., Zhao, J., Yang, M., Wang, T., Wang, N., Lyu, L., Niyato, D., Lam, K.Y., 2020. Local differential privacy-based federated learning for internet of things. IEEE Internet of Things Journal 8, 8836–8853.



Rui Chen received the master's degree in software engineering from the East China Normal University, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Software Technology, Dalian University of Technology, China. His research interests include physical layer and MAC layer of vehicular Ad Hoc networks, federated learning, cyberspace security, differential privacy and privacy preserving.

Xiaoyu Chen is currently pursuing a master's degree in software engineering, Dalian University of Technology. His research interests include federated learning, binary security, and communication privacy protection.

Jing Zhao (Member, IEEE) received the Ph.D. degree in computer science and technology from the Harbin Institute of Technology of China in 2006. In 2010, she was with the Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina, working as a postdoctoral researcher under supervision of Dr. Kishor Trivedi. From 2006 to 2018, she was a professor at the School of Computer Science and Technology, Harbin Engineering University, China. She is currently a professor at the School of Software Technology, Dalian University of Technology, China. Her research interests include physical layer and MAC layer of vehicular ad hoc networks, reliability engineering, and network optimization.