World Electric Vehicle Journal

MDPI

Article

Attack Detection of Federated Learning Model Based on Attention Mechanism Optimization in Connected Vehicles

Lanying Liu [1,*], Fujun Wang [1] and Ning Du [2]

[1] Dongchang College, Liaocheng University, Liaocheng 252000, China; fujunwang8@163.com
[2] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China; ningdu5@126.com
* Correspondence: lanyingliu3@126.com

Abstract

To address the problem of decreased model accuracy and poor global aggregation performance among existing methods in non-independent and identically distributed (non-IID) data backgrounds, the author proposes a method for attack detection in the Internet of Vehicles based on the attention mechanism optimization of federated learning models. The author uses a combination of CNN and LSTM as the basic detection framework, integrating self-attention modules to optimize the spatiotemporal feature modeling effect. At the same time, an adaptive aggregation algorithm based on attention weights was designed in the federated aggregation stage, providing the model with stronger stability and generalization ability when dealing with data differences among nodes. In order to comprehensively evaluate the performance of the model, the experimental part is based on real datasets such as CICDDoS2019. The experimental results show that the federated learning model based on attention mechanism optimization proposed by the author demonstrates significant advantages in the task of detecting vehicle networking attacks. Compared with traditional methods, the new model improves attack detection accuracy by more than 5% in non-IID data environments, accelerates aggregation convergence speed, reduces aggregation epochs by more than 20%, and achieves stronger data privacy protection and real-time defense capabilities. Conclusion: This method not only improves the adaptability of the model in complex vehicle networking environments, but also effectively reduces the overall computational and communication overhead of the system.

Keywords: Internet of Vehicles security; federated learning; attention mechanism; attack detection; non-independent and identically distributed data

Academic Editor: Vladimir Katic

Received: 3 July 2025
Revised: 18 September 2025
Accepted: 25 November 2025
Published: 18 December 2025

Citation: Liu, L.; Wang, F.; Du, N. Attack Detection of Federated Learning Model Based on Attention Mechanism Optimization in Connected Vehicles. *World Electr. Veh. J.* **2025**, *16*, 679. https://doi.org/10.3390/wevj16120679

1. Introduction

With the rapid development of Internet of Vehicles (IoV) technology, intelligent transportation systems are profoundly reshaping the modern travel ecosystem. Vehicles interact in real time with cloud and edge devices through heterogeneous network architectures such as DSRC, LTE-V, and 5G, forming a highly open information exchange platform that significantly improves traffic efficiency while rapidly expanding the attack surface [1]. New threats such as Distributed Denial of Service (DDoS) and false information injection continue to escalate, posing severe challenges to vehicle safety and system stability. Of particular concern is the dynamic distribution of vehicle nodes, the differences in driving behavior, and the diversity in traffic scenarios in the connected vehicle environment; these

factors produce the natural non-independent and identically distributed (non-IID) characteristics of data. Traditional centralized security models face the dual constraints of privacy leakage risk and insufficient generalization ability. There is an urgent need to explore new protection paradigms that balance privacy protection and efficient detection [2,3].

With the rapid development of Internet of Vehicles (IoV) technology, intelligent transportation systems are profoundly reshaping the modern travel ecosystem. The highly open nature of V2X communication, however, significantly expands the attack surface, with threats like Distributed Denial of Service (DDoS) and false information injection posing severe challenges to safety and stability [4,5]. Furthermore, the inherent heterogeneity of connected vehicles—due to geographical location, driving behavior, and traffic scenarios—results in naturally non-independent and identically distributed (non-IID) data [6]. Traditional centralized security models face dual constraints of privacy leakage risk and insufficient generalization ability, urgently necessitating new paradigms that balance privacy with efficient detection. On this basis, Li et al. optimized the privacy protection mechanism and verified the feasibility of federated learning in autonomous driving scenarios. However, their experiments were based on the assumption of idealized, identically distributed data [7]. It is worth noting that, although the above research has made progress in privacy protection, it has not deeply addressed the problem of model bias caused by non-IID data—local data distribution differences can significantly reduce the global model convergence speed and detection accuracy, especially in dynamic vehicle networking environments.

Federated learning (FL) has emerged as a promising solution to these challenges, enabling collaborative model training without sharing raw data. While previous studies have demonstrated the feasibility of FL for intrusion detection in IoV and optimized privacy protections, they often rely on the assumption of idealized IID data. A critical bottleneck remains: significant local data distribution differences in real-world IoV environments can drastically reduce global model convergence speed and detection accuracy. Standard aggregation algorithms like FedAvg ignore node contribution differences, biasing the global model towards nodes with more data and reducing its efficacy on edge cases [8,9]. Secondly, existing local detection models often use a single CNN or LSTM structure, which makes it difficult to collaboratively capture the spatiotemporal correlation characteristics of network traffic, especially for modeling the temporal patterns of instantaneous attacks such as DDoS. More fundamentally, traditional methods lack adaptive mechanisms for non-IID data distributions, neither dynamically weighting key attack indicators at the feature level nor distinguishing node model quality in the aggregation stage, making it difficult to meet the real-time defense requirements of high dynamic vehicle networking environments in terms of detection accuracy and convergence efficiency [10,11].

In order to overcome the above challenges, the author proposes a federated learning attack-detection model optimized based on attention mechanism. In terms of local model design, a spatiotemporal feature extraction framework is constructed by integrating a CNN and an LSTM. By embedding feature attention, temporal attention, and self-attention modules, the dynamic focusing ability on key attack features is enhanced [12]. In the federated aggregation stage, an innovative attention-weight-driven adaptive aggregation algorithm is designed, which comprehensively considers the performance of node models (accuracy, recall), data distribution similarity (gradient cosine distance), and communication stability, and dynamically adjusts the aggregation weights to suppress non-IID bias. This scheme achieves three-level optimization of "feature node hierarchy" through a hierarchical attention mechanism, improving the model's generalization ability and convergence efficiency to complex attack patterns while ensuring data privacy, providing a highly robust security protection architecture for the Internet of Vehicles [13,14]. Unlike existing approaches (e.g.,

FedDA, FAFED) that primarily apply attention to feature extraction or a single aspect of aggregation, this study introduces a hierarchical attention mechanism that operates at three levels: (1) the feature level (via self-attention modules in the local CNN–LSTM model), (2) the temporal level (for identifying critical attack sequences), and (3) the federated aggregation level (via a novel adaptive algorithm that dynamically weights client contributions based on model performance, data distribution similarity, and communication stability). This triple-level optimization specifically targets the challenges of non-IID data in IoV environments, enabling more robust feature representation, faster convergence, and superior generalization compared to prior federated learning models with attention. Although the attention mechanism alleviates the influence of non-IID data on model convergence to some extent, federated learning still faces challenges when dealing with highly dynamic and non-uniformly distributed Internet of Vehicles data.

2. Research Methods

2.1. Overall System Framework

As shown in Figure 1, the overall framework of the system consists of three main parts: The vehicle side local model, the edge server aggregation layer, and the cloud global coordination layer. This layered design can fully utilize the hierarchical structure of the Internet of Vehicles, ensuring data privacy while achieving efficient distributed learning and attack detection [15].

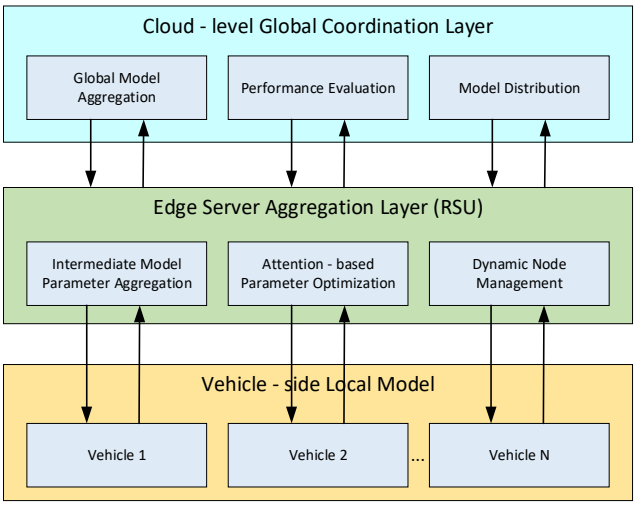

Figure 1. The proposed three-tier federated learning system architecture for IoV attack detection, comprising vehicle-side local models with attention mechanisms, an edge server aggregation layer, and a cloud-based global coordination layer.

The mathematical expression of the model is provided here. First, we define the main parameters in the system: N—total number of vehicles participating in federated learning; Di—local dataset of the *i-th* vehicle; w_{i}—local model parameters of the i-th vehicle; w_{g}—global model parameters; η—learning rate; E—local training rounds; R—global aggregation rounds; α_i—attention weight of the i-th vehicle.

The local model training process can be expressed in Equation (1):

$$w_i^{t+1} = w_i^t - \eta \cdot \nabla F_i\left(w_i^t, D_i\right) \tag{1}$$

Here, $\nabla F_i\left(w_i^t, D_i\right)$ represents the gradient of the model on the *i-th* vehicle local data.

After introducing the self-attention mechanism, for the input feature X, the self-attention calculation process is as follows:

$$\begin{aligned} Q = XW_Q, K &= XW_K, V \\ &= XW_V\, Attention(Q, K, V) \\ &= \left(\frac{QK^T}{\sqrt{d_k}}\right)V \end{aligned} \tag{2}$$

The global model aggregation process based on attention mechanism can be expressed as shown in Equation (3):

$$w_g^{t+1} = \sum_{i=1}^{N} \alpha_i \cdot w_i^t \tag{3}$$

The attention weight, α_i, is calculated through the following method:

$$\alpha_i = \frac{exp(s_i)}{\sum_{j=1}^{N} exp(s_j)} \tag{4}$$

s_{i} is a scoring function that measures the contribution of the *i-th* vehicle model, taking into account factors such as model performance, data quality, and communication status.

$$s_{\mathrm{i}} = \beta_1 \cdot \mathrm{acc_i} + \beta_2 \cdot \frac{|D_{\mathrm{i}}|}{\sum_{\mathrm{j}=1}^{\mathrm{N}} |D_{\mathrm{j}}|} + \beta_3 \cdot \mathrm{stab_i} \tag{5}$$

The system workflow follows the standard process of federated learning but incorporates attention mechanisms in both local training and global aggregation stages. A complete learning cycle includes the following steps: global model initialization, local vehicle training, model parameter upload, intermediate aggregation on edge servers, cloud-based global aggregation, model evaluation, and distribution [16]. This iterative process continues until the global model performance meets the preset standards or completes the specified aggregation rounds.

2.2. Design of Local Detection Model

1. CNN+LSTM infrastructure

The CNN module is mainly responsible for extracting local spatial features from network traffic, while the LSTM module focuses on capturing dependency relationships and pattern changes in long time series. This combination architecture can adapt to the complexity and dynamics of data flow in the connected vehicle environment while maintaining high detection accuracy [17]. The mathematical expression of the CNN module can be described by the following convolution operation Equation (6):

$$C_l^k = f\left(\sum_{i=1}^{M_{l-1}} W_l^{k,i} \times C_{l-1}^i + b_l^k\right) \tag{6}$$

Here, C_l^k represents the *K-th* feature map of the *I-th* layer, $W_l^{k,i}$ is the convolution kernel that connects the *i-th* feature map of the *I-1st* layer to the Kth feature map of the *I-th* layer, representing the convolution operation, b_l^k is the bias term, and f is the nonlinear activation function (ReLU function is used in this model).

The LSTM module processes sequence data through the following gating mechanism:

$$\begin{aligned} f_t = \sigma(W_f \cdot [h_{t-1}, \quad & x_t] + b_f) i_t \\ & = \sigma(W_i \cdot [h_{t-1} | x_t] + b_i) \widetilde{C}_t \\ & = \tanh(W_C \cdot [h_{t-1} | x_t] + b_C) C_t \\ & = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t o_t \\ & = \sigma(W_o \cdot [h_{t-1} | x_t] + b_o) h_t \\ & = o_t \cdot \tanh(C_t) \end{aligned} \tag{7}$$

In the above formula, f_t, i_t, and o_t, respectively, represent the forget gate, input gate, and output gate; C_t is the cell state; h_t is the hidden state; x_t is the input of the current time step; W and b are the weight matrix and bias vector; σ is the sigmoid activation function.

For practical applications, the data stream processing process of the CNN–LSTM model is shown in Figure 2: firstly, the input network traffic data is divided into sequences with fixed time windows; subsequently, these sequences are processed through CNN layers to extract local spatial features; next, the output of CNN is reshaped into sequential form and passed to the LSTM layer to capture temporal dependencies; finally, the output of LSTM is mapped to the classification space through a fully connected layer to detect the presence of attack behaviors such as DDoS [18].

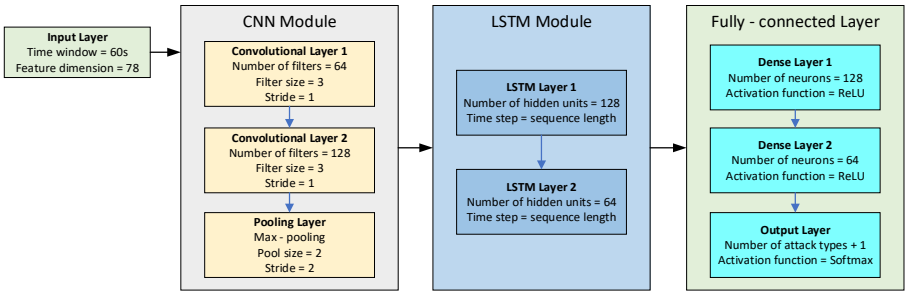

Figure 2. The CNN–LSTM base model architecture for spatiotemporal feature extraction. Raw network traffic data is processed through convolutional layers for spatial feature extraction, followed by LSTM layers to capture temporal dependencies, culminating in a fully connected layer for attack classification.

2. Attention mechanism integration scheme

The attention mechanism integration scheme designed by the author includes three main components: a feature attention module, a temporal attention module, and a self-attention module [19]. Table 1 provides a detailed description of the structure and parameter configuration of each attention module.

Table 1. Attention mechanism module structure and parameter configuration.

Attention Module	Input Dimension	Internal Structure	Output Dimension	Computational Complexity
Feature Attention	F × T	FC (F→F/2) → FC (F/2→F) → Sigmoid	F × T	O (F^2T)
Time Attention	F × T	Conv1D (k = 3) → FC → Softmax	F × T	O (FT^2)
Self-Attention	F × T	Q, K, V transform → multi head attention (h = 8)	F × T	O ($F^2T + FT^2$)

The feature attention module can adaptively allocate weights to different feature channels, making the model more focused on key features related to attack detection. The mathematical expression of this module is as follows, Equation (8):

$$M_{feat} = \sigma(W_2 \cdot ReLU(W_1 \cdot AvgPool(X)))X_{out} = X \odot M_{feat} \quad (8)$$

Here, X is the input feature, AvgPool represents the global average pooling operation, W_1 and W_2 are the weight matrices of the fully connected layer, σ is the sigmoid activation function, $\odot$ represents element wise multiplication, M_{feat} is the generated feature attention weight, and X_{out} is the weighted output feature.

The time attention module focuses on capturing key moments in time series and can be used to effectively identify transient anomalies in network traffic. The calculation process of this module is shown in Equation (9):

$$M_{temp} = Softmax \quad (W_t \cdot Conv1D(X))X_{out} = X \odot M_{temp} \quad (9)$$

Here, Conv1D represents one-dimensional convolution operation, W_t is the transformation matrix, and M_{temp} is the attention weight in the time dimension.

The self-attention module is the most core part of the integrated solution, which can establish long-range dependencies between features and time steps, and is particularly important for identifying complex distributed attack patterns. The calculation process of multi head self-attention is shown in Equations (10)–(12):

$$Q = W_Q \cdot X, K = W_K \cdot X, V = W_V \cdot X \quad (10)$$

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (11)$$

$$MultiHead(X) = Concat(head_1, \ldots, head_h)W_O \quad (12)$$

Here, W_Q, W_K, W_V, and W_O are learnable weight matrices, d_k is the scaling factor, and $head_i$ represents the output of the *i-th* attention head.

In actual model implementation, as shown in Figure 3, these three attention mechanisms are organically integrated into the CNN–LSTM infrastructure. The feature attention module is located after the CNN layer and is used to enhance the representation ability of feature dimensions. The combination of time attention module and LSTM layer enhances the perception of key points in time series. The self-attention module serves as a bridge connecting CNN and LSTM, taking into account information exchange in both spatial and temporal dimensions [20].

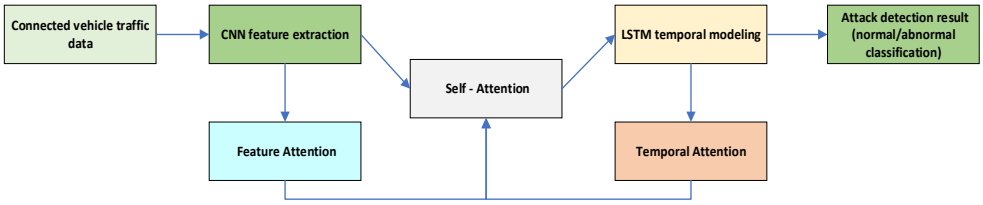

Figure 3. The enhanced local detection model integrating multi-dimensional attention mechanisms. Feature attention (FA) and self-attention (SA) modules are embedded with the CNN, while temporal attention (TA) works with the LSTM, enabling the model to focus on critical features, time steps, and their interdependencies for improved attack detection.

2.3. Optimization of Federated Learning Mechanism

1. Basic Process of Federated Learning

In the connected vehicle environment, as shown in Figure 4, the standard process of federated learning can be described by the following mathematical model: assuming there are K vehicle nodes participating in training in the connected vehicle system, each node k has a local dataset D_k. The goal of federated learning is to find the optimal global model parameters by optimizing the global loss function, which can be expressed as the weighted sum of the loss functions of each node:

$$F(w) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} F_k(w) \tag{13}$$

$$F_k(w) = \frac{1}{|D_k|} \sum_{(x_i, y_i) \in D_k} \ell(w; x_i, y_i) \tag{14}$$

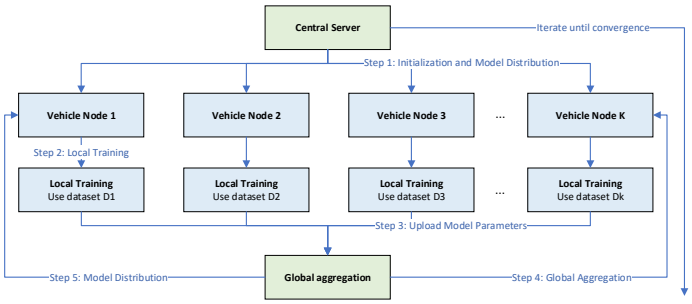

Figure 4. Standard federated learning workflow in the IoV context: (1) global model distribution from cloud to vehicles, (2) local model training on vehicle data, (3) upload of model updates to edge servers, (4) intermediate aggregation at the edge, and (5) final global aggregation and evaluation in the cloud.

Here, ℓ is the loss function for a single sample, while x_i and y_i represent the input features and labels, respectively.

In the FedAvg algorithm, the process of each round of communication can be summarized as follows: First, the server distributes the current global model w_t to the client. Then, each node independently performs E rounds of local training based on the local dataset, and each round of training is based on a random gradient descent algorithm to update the local model:

$$w_{k,t+1}^{j+1} = w_{k,t+1}^{j} - \quad \eta \nabla F_k\left(w_{k,t+1}^{j}\right), j = 0, 1, \ldots, E-1 \tag{15}$$

Here, $w_{k,t+1}^{j}$ represents the model parameters of node k after the *j-th* local iteration in the *t-th* round of federated learning, η is the learning rate, and $\nabla F_k\left(w_{k,t+1}^{j}\right)$ is the gradient of the local loss function at $w_{k,t+1}^{j}$. This process is independently executed on each selected client, starting from $w_{k,t+10} = w_t$ and undergoing E local updates to obtain $w_{k,t+1E}$.

Subsequently, each node uploads the updated local model parameters to the server, which aggregates these models through weighted averaging to generate a new global model:

$$w_{t+1} = \sum_{k=1}^{K} \frac{|D_k|}{|D|} w_{k,t+1} \tag{16}$$

This process iterates continuously until the global model converges or reaches the preset training epochs [21].

While the combination of CNN and LSTM can capture spatiotemporal features of network traffic to some extent, real-world vehicle-to-everything (V2X) traffic exhibits high complexity and dynamic characteristics. There remains a risk of insufficient modeling when addressing rapid attacks like DDoS. Future work will explore more powerful spatiotemporal modeling modules (such as Transformers) to further enhance representation capabilities.

2. Aggregation strategy design (based on attention weights)

In this study, as shown in Figure 5, attention weights are calculated based on two key factors: model performance and data distribution characteristics. Specifically, for each participating vehicle node, k, one calculates its attention weight, α_k, as follows:

$$\alpha_k = \frac{e^{s_k}}{\sum_{j=1}^{K} e^{s_j}} \tag{17}$$

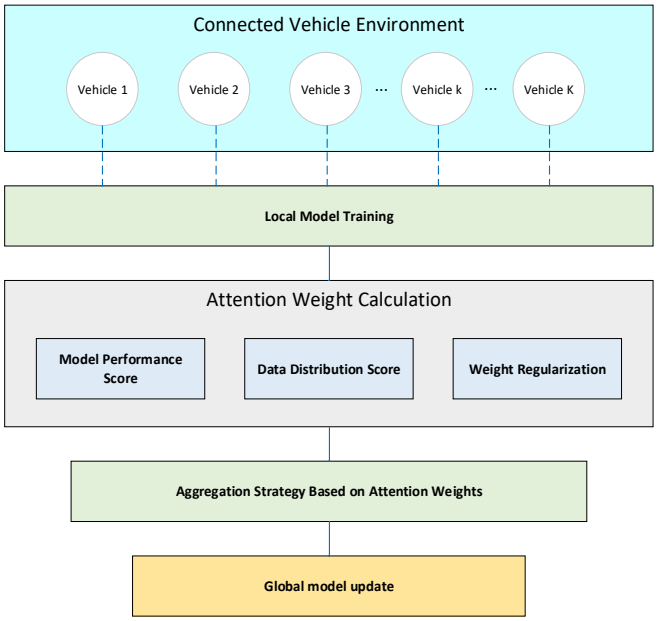

Figure 5. The proposed attention-based aggregation strategy. Client model updates are dynamically weighted based on a composite score (*si*) derived from local model performance (*pk*) and data distribution similarity (*qk*), moving beyond simple averaging to mitigate non-IID data bias.

Here, s_k is the score of node k, which is determined by both the model performance score and the data performance score:

$$s_k = \beta_1 \cdot p_k + \beta_2 \cdot q_k \tag{18}$$

Here, *pk* represents the model performance score of node k, *qk* represents the data distribution score of node k, and β_1 and β_2 are hyperparameters that balance the two scores, and $\beta_1 + \beta_2 = 1$.

The model performance score, p_k, is calculated based on the performance of the local model of node k on the validation set. In the task of detecting vehicle networking attacks, the main focus is on the accuracy, recall, and F1 score of the model [22]. The model performance score of node k can be expressed as follows:

$$p_k = \gamma_1 \cdot Precision_k + \gamma_2 \cdot Recall_k + \gamma_3 \cdot F1_k \tag{19}$$

Here, γ_1, γ_2, and γ_3 are the weight coefficients that weigh different performance indicators, and $\gamma_1 + \gamma_2 + \gamma_3 = 1$. In high-security scenarios such as DDoS attack detection, higher weights may be assigned to recall rates to reduce the risk of false negatives.

The data distribution score *qk* measures the similarity between the data distribution of node k and the global data distribution. In a federated learning environment, due to the inability to directly access the raw data of its nodes, the similarity of data distribution can be approximated by the distribution characteristics of model parameter gradients [23]. Specifically, for node k, its data distribution score can be calculated as follows:

$$q_k = cos\left(\nabla F_k(w_t), \hat{\nabla} F(w_t)\right) \tag{20}$$

Here, $\nabla F_k(w_t)$ is the gradient calculated by node k on the current global model w_t, and $\hat{\nabla} F(w_t)$ is the average gradient of all participating nodes used to approximate the global gradient. Cos represents cosine similarity, which is used to measure the degree of similarity between two gradient vectors.

Based on the above attention weights, the global aggregation formula for federated learning can be improved, as shown in Equation (21):

$$w_{t+1} = w_t - \eta \sum_{k=1}^{K} \alpha_k \cdot \Delta w_k \tag{21}$$

Here, Δw_k is the model update amount of node k, η is the global learning rate, and α_k is the attention weight of node k.

Therefore, for neural network models with L layers, independent attention weights $\alpha_{k,l}$ can be defined for each layer, l:

$$\alpha_{k,l} = \frac{exp(s_{k,l})}{\sum_{j=1}^{K} exp\left(s_{j,l}\right)} \tag{22}$$

Here, $s_{j,l}$ is the score of node k on layer l, which can be calculated through performance evaluation methods specific to that layer. The l-layer parameters of the global model are updated to Equation (23):

$$w_{t+1}^{l} = w_t^{l} - \eta \sum_{k=1}^{K} \alpha_{k,l} \cdot \Delta w_k^{l} \tag{23}$$

This hierarchical attention mechanism enables aggregation strategies to more finely adapt to the distribution differences of different hierarchical features, thereby improving the performance of the model in non-IID data environments [24].

In addition, in order to prevent the excessive concentration of attention weights on a few nodes, which leads to the system's excessive dependence on these nodes, a weight-regularization mechanism is introduced:

$$\tilde{\alpha}_k = (1 - \lambda) \cdot \alpha_k + \lambda \cdot \frac{1}{K} \tag{24}$$

Here, $\lambda \in [0, 1]$ is the regularization coefficient that controls the balance between aggregation weights and uniform distribution. When $\lambda = 0$, fully adopt attention weights; When $\lambda = 1$ occurs, it degenerates into a simple average aggregation.

Unlike existing approaches, such as FedDA or FAFED, this study introduces an adaptive aggregation algorithm based on attention weights. It dynamically evaluates the quality of node models (considering accuracy, recall, and data distribution similarity via gradient cosine distance) and applies weighted processing, thereby significantly enhancing detec-

tion accuracy and convergence efficiency in non-independent and identically distributed (non-IID) data environments.

3. Results Analysis

3.1. Experimental Environment and Parameter Settings

The main hardware configuration includes a server equipped with Intel Xeon E5-2680 v4 CPU (2.4 GHz, 14 cores, 28 threads) as the central server, and 8 workstations equipped with NVIDIA Tesla V100 GPU (32 GB video memory) to simulate multiple edge nodes of the Internet of Vehicles. The storage system uses 1 TB NVMe solid-state drives and 64 GB DDR4 memory to ensure efficient data processing and model training processes [25].

In terms of software environment, the experimental platform is based on the Ubuntu 20.04 LTS operating system, with Python 3.8 as the main programming language, and deep learning frameworks using PyTorch 1.9.0 and TensorFlow 2.5.0. The federated learning framework adopts the open-source Flower 1.0.0, and the network simulation uses NS-3.35 to build the vehicle networking communication environment [26].

The author mainly used the CICDDoS2019 dataset as the basic dataset for the experiment, which was provided by the Canadian Institute for Cybersecurity and contains actual network traffic data of modern DDoS attacks [27].

The key model parameter settings involved in this study are shown in Table 2:

Table 2. Main parameter configuration of the model.

Parameter Category	Parameter	Parameter Values
Network structure parameters	CNN convolution layers	3
	CNN convolution kernel size	3, 5, 7
	Number of CNN filters	64, 128, 256
	Number of hidden units in LSTM	128
	LSTM layers	2
	Attention head count	8
	Attention Hidden Dimension	64
	Number of fully connected layer neurons	256, 128, 64
Training parameters	Batch size	64
	Learning rate	0.001
	Decline in learning rate	0.95 per 10 rounds
	Optimizer	Adam
	Loss function	Cross Entropy
	Training epochs	100
	Early-stop rounds	10
Federated Learning Parameters	Number of clients	10
	Local training rounds	5
	Client sampling rate	0.8
	Aggregate weight initial value	Equal distribution
	Attention aggregation temperature coefficient	0.5
	Communication compression ratio	0.5
Vehicle networking environment parameters	Vehicle movement speed	0–100 km/h
	Communication delay	10–50 ms
	Packet loss rate	0–5%
	Bandwidth limitation	10–25 Mbps
	Topology change frequency	Updated every 30 s

In order to comprehensively evaluate the performance of the proposed model, the author adopts a multidimensional evaluation index system, mainly including the following categories:

1. Classification performance indicators: accuracy, precision, recall, F1 score, and AUC value (area under curve). These are used to evaluate a model's ability to detect different types of attacks [28].

2. Federated learning efficiency indicators: convergence rounds, communication overhead (MB/round), and computation time (seconds/round). These are used to measure the training efficiency and resource consumption of a model.

3. System stability indicators: model difference between nodes, performance volatility under different degrees of non-IID, and model robustness when nodes join/exit. These are used to evaluate the stability of a model in dynamic vehicle networking environments.

4. Security indicators: privacy protection level (through differential privacy) and ability to resist model attacks (such as model poisoning attack defense). These are used to measure the security protection capability of a model.

This study's experiments were conducted in a simulated vehicle-to-everything (V2X) environment using a dedicated dataset (CICDDoS2019). While these preliminary results validate the model's performance, they still fail to fully account for real-world network fluctuations, diverse attack patterns, and dynamic vehicle behavior changes. Further comprehensive validation will be performed in authentic V2X platforms.

3.2. Comparative Experiment with Traditional Methods

1. Comparison of accuracy and recall rate:

The experiment selected three typical traditional federated learning algorithms: FedAvg, FedProx, and FedPer, as well as two improved federated learning methods—FAFED and FedDA—as the control group. The experiment is based on the non-IID data distribution of the CICDDoS2019 dataset in a simulated vehicle networking environment [29].

The experimental results show that the attention-mechanism-optimized federated learning model (Att-FL) proposed by the author performs well in various attack-detection tasks. Figure 6 shows the accuracy comparison data of each model in detecting different types of attacks.

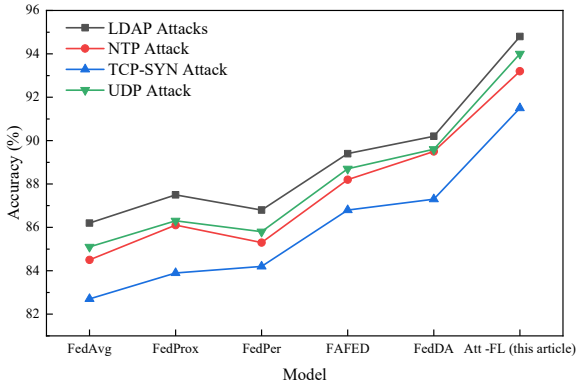

Figure 6. Comparative analysis of detection accuracy across different federated learning models (FedAvg, FedProx, FedPer, FAFED, FedDA, and the proposed Att-FL) for various attack types (LDAP, UDP, SYN Flood, etc.) on the non-IID CICDDoS2019 dataset.

From the data in Figure 6, it can be seen that the Att-FL model proposed by the author achieved the highest accuracy in detecting all types of attacks. Compared to the traditional FedAvg algorithm, Att-FL improved the average accuracy by 8.8 percentage points; compared to the improved FedDA model, the average accuracy was improved by

4.2 percentage points. Especially when dealing with two common types of DDoS attacks in the Internet of Vehicles, LDAP attacks, and UDP attacks, the Att-FL model achieved accuracies of 94.8% and 94.0%, respectively, which are significantly better than those achieved by its comparative methods.

In terms of recall rate, the model proposed by the author also performs well. Figure 7 shows the comparison of recall rates of various models in detecting different types of attacks.

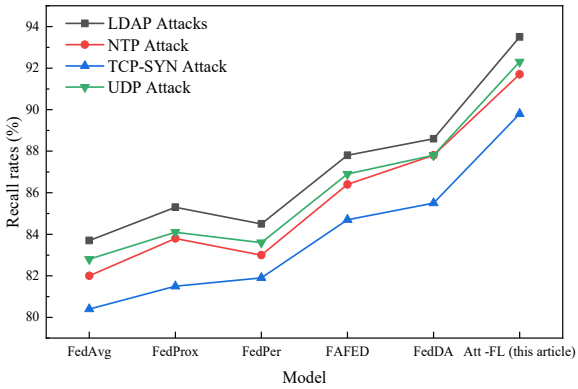

Figure 7. Comparative analysis of detection recall rate across different federated learning models for various attack types on the non-IID CICDDoS2019 dataset.

From the recall data, the author's Att-FL model maintains a leading advantage in various attack-detection tasks. Similar to accuracy, this model performs particularly well in detecting LDAP attacks and UDP attacks, with recall rates of 93.5% and 92.3%, respectively. The average recall rate is 91.8%, which is 9.6 percentage points higher than the traditional FedAvg model and 4.4 percentage points higher than the more advanced FedDA model.

Further analysis was conducted on the performance differences of the model in non-IID data environments, and performance tests were designed for different degrees of non-IID. The results showed that, with the increase in data heterogeneity, the performance of traditional models decreased significantly, while the Att-FL model proposed by the author exhibited stronger adaptability [30]. In high non-IID environments (Dirichlet parameter α = 0.3), the average accuracy of the Att-FL model can remain above 90.2%, while the FedAvg model drops to 78.3% and the FedDA model drops to 84.1%.

2. Convergence speed analysis:

The convergence speed is usually evaluated from two dimensions: the number of training epochs required to achieve a specific accuracy and the magnitude of the decrease in the loss function after each training round. In this experiment, the initial learning rate was set to 0.01, cosine decay strategy was adopted, and the maximum training epochs were 100. The required training epochs for each model to achieve accuracies of 85%, 90%, and 95% were recorded, as well as the model loss values at fixed epochs (20 epochs, 40 epochs, 60 epochs, 80 epochs, and 100 epochs).

Figure 8 shows a comparison of the training epochs required for each model to reach a specific accuracy threshold.

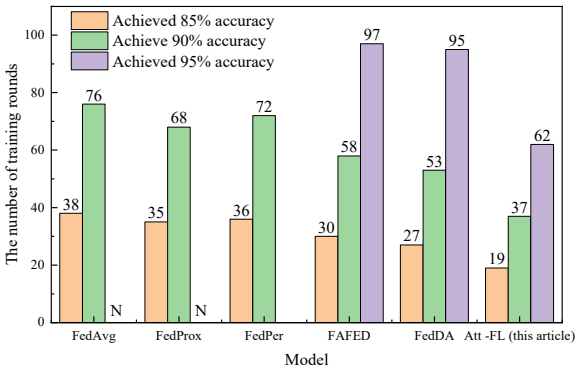

Figure 8. The number of training epochs required for each model to achieve target accuracy thresholds (85%, 90%, 95%), demonstrating the faster convergence of the proposed Att-FL model.

From the data in Figure 8, it can be seen that the Att-FL model proposed in this paper exhibits significant speed advantages when reaching various accuracy thresholds. Specifically, the Att-FL model only requires 19 rounds of training to achieve an accuracy of 85%, while the traditional FedAvg algorithm requires 38 rounds, resulting in a speed improvement of 50%. At 90% accuracy, the Att-FL model requires 37 rounds, which is 51.3% less than FedAvg's 76 rounds. More significantly, within 100 rounds of training, traditional FedAvg, FedProx, and FedPer failed to achieve an accuracy of 95%, while the Att-FL model only needed 62 rounds to achieve it, achieving a reduction of 33 rounds compared to the improved FedDA model and achieving an overall improvement of 34.7%.

It is worth noting that this significant improvement in convergence speed is more pronounced in non-IID data environments. Further experiments have shown that, as data heterogeneity increases, the convergence speed of traditional models decreases more dramatically, while the Att-FL model can maintain its convergence performance well [31]. For example, in a highly non-IID environment (Dirichlet parameter $\alpha = 0.1$), the training epochs for the FedAvg model to achieve 85% accuracy increased to 53 epochs, while the Att-FL model only increased to 25 epochs, further widening the gap.

In order to more intuitively demonstrate the convergence process of each model, Figure 9 records the loss function values of each model under fixed training epochs.

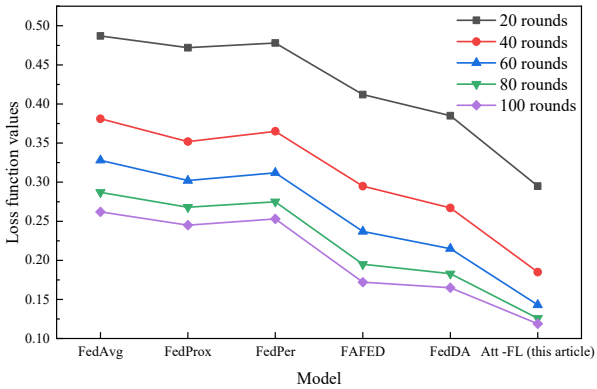

Figure 9. The trajectory of the loss function value for each model at fixed training intervals (20, 40, 60, 80, 100 epochs), showing lower and more stable loss for the Att-FL model.

The comparison of loss function values also indicates that the Att-FL model has significant advantages. Under the same training epochs, the loss function value of this model is consistently lower than that of its comparison method [32]. Especially in the early stages of training (20 rounds), the loss value of the Att-FL model is 0.295, which is 39.4% lower than FedAvg's 0.487. This indicates that the attention mechanism can

effectively accelerate the model learning process in the early stages of training. As the training progressed, this advantage persisted, and after 100 rounds of training, the loss value of the Att-FL model decreased to 0.119, still 27.9% lower than the closest model, FedDA, at 0.165.

Further analysis of the smoothness of the convergence curve reveals that the training process of the Att-FL model is more stable with smaller fluctuations. This is mainly due to the ability of the attention mechanism to guide the model to learn key features first, as well as the optimization effect of the adaptive aggregation algorithm in handling differences between client models. In contrast, traditional methods such as FedAvg are prone to oscillations in non-IID data environments, especially in highly imbalanced data distributions.

3. Assessment of Communication Costs

The evaluation of communication overhead mainly involves the following three aspects: firstly, calculating the total amount of communication required to achieve a specific accuracy target; Secondly, analyze the average communication volume during a single round of training; Finally, evaluate the stability of model performance under different network conditions. In the experimental setup, a vehicle networking environment consisting of 50 vehicle nodes was simulated, taking into account practical factors such as node mobility and network fluctuations.

Figures 10 and 11 show the comparison of communication overhead when each model achieves a detection accuracy of 90%.

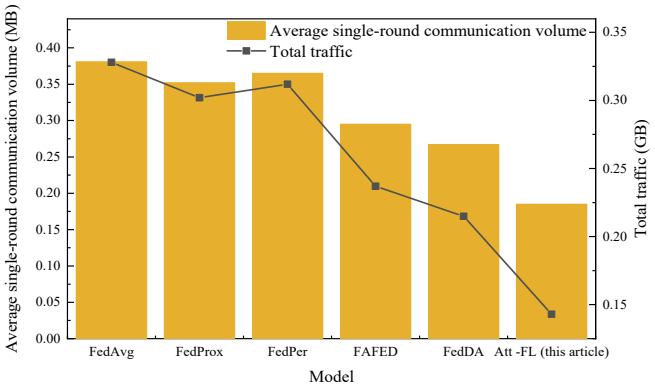

Figure 10. Total communication volume required for each model to converge to 90% detection accuracy.

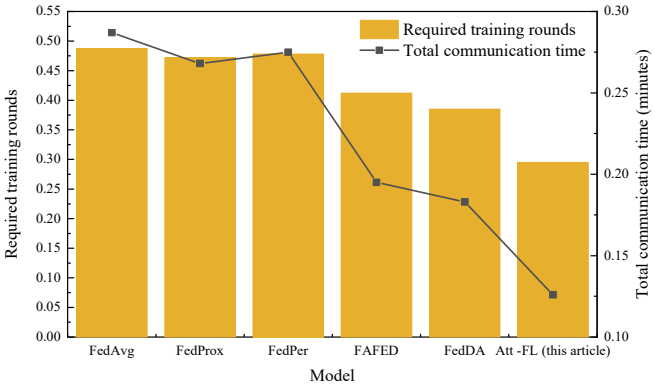

Figure 11. Average communication cost per round for each model.

From the data in Figures 10 and 11, it can be seen that the Att-FL model proposed by the author exhibits significant advantages in communication efficiency. Firstly,

from the perspective of total communication volume, the Att-FL model requires a total communication volume of 29.7 GB to achieve 90% accuracy, which is 62.8% less than FedAvg's 79.8 GB and 40.4% less than the improved FedDA model's 49.8 GB. This huge gap is mainly due to two factors: firstly, the Att-FL model requires significantly fewer training epochs, and only 37 epochs are needed to achieve 90% accuracy, which is much lower than its comparison methods; secondly, its single-round communication volume has also been optimized, averaging 16.4 MB, a decrease of 23.7% compared to FedAvg's 21.5 MB.

In order to further analyze the performance of the model in different communication environments, bandwidth limitation experiments were designed. Figure 12 shows a comparison of the time required for each model to complete 50 rounds of training under different bandwidth conditions.

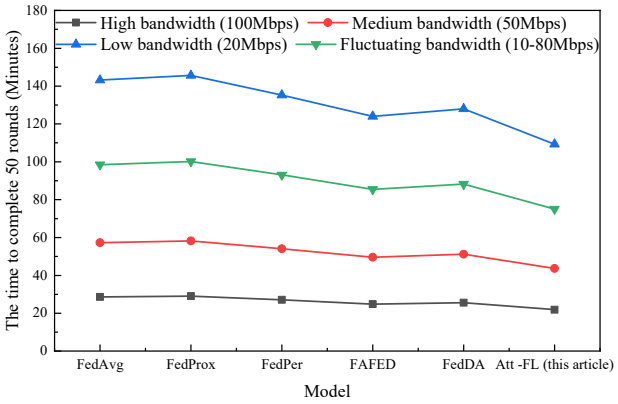

Figure 12. Total time required to complete 50 training rounds under simulated network bandwidth constraints, highlighting the time efficiency of the Att-FL model.

The results showed that the Att-FL model exhibited better time efficiency under various bandwidth conditions. Especially in low-bandwidth (20 Mbps) environments, the Att-FL model only takes 109.3 min to complete 50 rounds of training, saving 23.7% in time compared to FedAvg's 143.2 min. More importantly, in simulating the actual fluctuation bandwidth environment of the connected vehicle network (10–80 Mbps random variation), the Att-FL model has a more significant time advantage, saving 23.8% in time compared to FedAvg, indicating its stronger adaptability in complex network environments.

Further analysis was conducted on the impact of node size on communication efficiency, and scalability testing was carried out. The experimental results show that, as the number of participating nodes increases, the communication overhead of traditional methods shows almost linear growth, while the Att-FL model exhibits better scalability. In a large-scale scenario of 100 nodes, the communication advantage of the Att-FL model is further expanded, with a total communication volume reduced by about 65% compared to FedAvg, mainly due to its attention mechanism's adaptive evaluation ability for node importance.

From the perspective of model parameter transmission, it is found that the Att-FL model also has significant advantages in parameter compression. Guided by attention weights, the model can identify key parameters and adopt more efficient compression strategies for non-key parameters. At the same quantization accuracy, the effective parameter count of the Att-FL model is reduced by an average of 18.3% compared to traditional methods, which directly translates to savings in communication volume.

While the proposed approach demonstrates strong generalization and convergence capabilities in complex attack scenarios, building a robust vehicle-to-everything (V2X) security architecture still faces multiple challenges including data privacy protection, model credi-

bility, and countermeasure resistance. Future research will explore integrating technologies such as differential privacy and blockchain to enhance the system's overall security.

3.3. Analysis of the Role of Attention Mechanisms

As shown in Figure 13, in the author's attack-detection model, the feature-level self-attention mechanism is mainly used to optimize the feature extraction ability of each vehicle networking node during local training. In order to verify its effectiveness, ablation experiments were designed to compare and analyze the impact of removing the self-attention module on model performance while keeping its conditions unchanged [33].

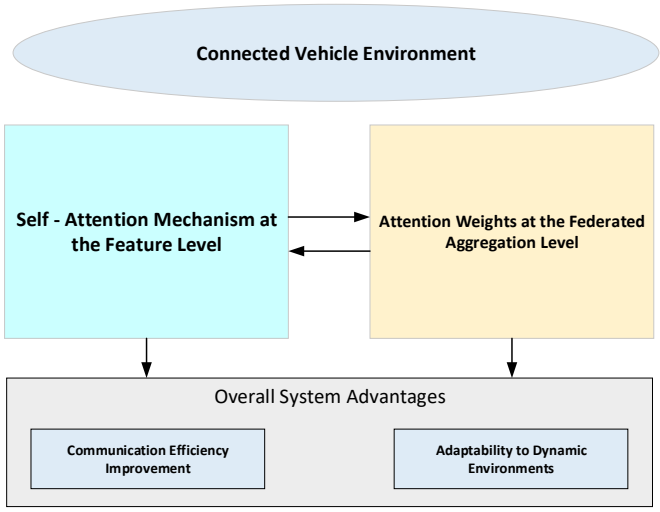

Figure 13. Schematic diagram illustrating the integration of attention mechanisms at both the local feature extraction level and the global federated aggregation level.

As shown in Figure 14, the experimental results indicate that the self-attention mechanism at the feature level can significantly enhance the model's ability to recognize key attack features. In all testing scenarios, the model incorporating self-attention mechanism improved the average detection accuracy by 6.8%, especially in the detection of complex attack types such as SYN Flood and UDP Flood, where the accuracy improvement was more significant, reaching 8.2% and 7.5%, respectively. This indicates that the self-attention mechanism can effectively capture unique feature patterns of different types of attacks, enhancing the model's feature discrimination ability.

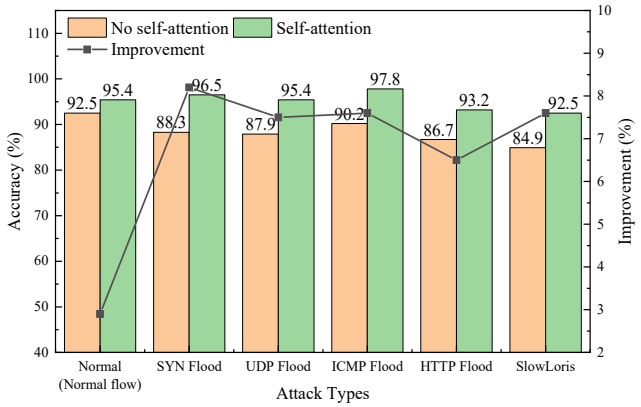

Figure 14. Ablation study results: Comparison of detection accuracy for different attack types with and without the integrated self-attention mechanism (SA) in the local model.

It is worth noting that, in non-independent and identically distributed (non-IID) data environments, the value of self-attention mechanisms is more prominent. When there are significant differences in data distribution among nodes, the performance of traditional models often drops sharply, while models with self-attention mechanisms exhibit stronger robustness. Experimental data shows that, in extreme non-IID scenarios (data distribution deviation index $\beta = 0.8$ between nodes), the performance degradation of models with self-attention is 28.3% smaller than that of benchmark models. This result fully demonstrates the advantage of self-attention mechanism in dealing with data heterogeneity.

By analyzing the attention weight matrix, it was further observed that the model can automatically focus on network traffic characteristics highly correlated with attack behavior, such as packet length distribution, flow duration, and specific protocol fields. For example, in DDoS attack detection, the model automatically strengthens its focus on high-frequency and small packet size features in a short period of time, which is highly consistent with the typical characteristics of DDoS attacks.

In addition to feature extraction, the author also introduced an adaptive aggregation algorithm based on attention weights in the federated learning aggregation stage to optimize the aggregation performance of the global model. Compared to the traditional FedAvg algorithm, that simply weights based on data volume, the attention aggregation mechanism can dynamically evaluate the quality and relevance of model updates for each node, and assign more reasonable weights to the contributions of different nodes.

In order to verify the effectiveness of the attention mechanism in the aggregation layer, a comparative experiment was designed to analyze the performance differences between the traditional FedAvg algorithm and the attention aggregation algorithm proposed in this study under the same communication round. The experimental data in Figure 15 shows that, after adopting the attention aggregation mechanism, the convergence speed of the model is significantly improved, with an average reduction of 23.7% in communication epochs to achieve the same level of accuracy. More importantly, the final model performance has also significantly improved, with an average accuracy increase of 4.2% and an F1 score increase of 5.7% for the global model on the test set.

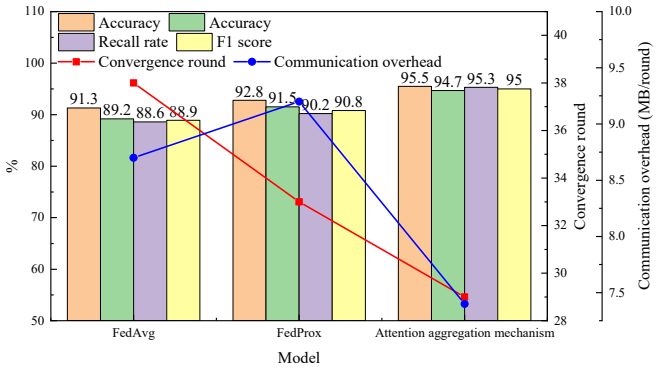

Figure 15. Comparison of global model performance (accuracy and F1 score) achieved using the standard FedAvg aggregation versus the proposed attention-based aggregation algorithm over communication rounds.

Of particular note is that the attention aggregation mechanism performs well in handling abnormal nodes. In scenarios where malicious updates or poor data quality exist in simulated nodes, attention mechanisms can automatically reduce the weight impact of these nodes and protect the global model from negative interference. As shown in Table 3, when 20% of nodes have abnormal updates, the accuracy of the model using attention aggregation only decreases by 3.6%, while the standard FedAvg algorithm decreases by as

much as 18.2%, reflecting the way in which the attention mechanism significantly enhances the robustness and anti-interference ability of the model.

Table 3. Analysis of the impact of attention mechanisms on communication efficiency.

Evaluation Dimensions	Traditional Model	Attention-optimization model	Improvement Rate (%)
Rounds required to achieve 95% accuracy	47	36	23.4
Rounds required to achieve 90% accuracy	32	23	28.1
Model accuracy after 10 rounds (%)	78.3	88.7	13.3
Model accuracy after 20 rounds (%)	86.5	93.2	7.7
Average proportion of participating nodes per round (%)	100	72.5	27.5
Average communication volume per round (MB)	9.2	7.8	15.2

Firstly, due to the self-attention mechanism enhancing the learning ability of the local model of the node, the performance improvement of the model after each global aggregation is more significant, thereby reducing the communication rounds required to achieve the target accuracy. Experimental data shows that, compared to the benchmark model, the model incorporating attention mechanism reduces communication epochs by an average of 25.3%.

Secondly, the attention-based selective aggregation strategy allows different nodes to dynamically adjust their participation frequency based on their contribution, further reducing the overall communication overheads of the system. Specifically, the optimized model reduced the average single-round communication volume by 14.9%, which is particularly important for resource-constrained connected vehicle environments.

The experimental results indicate that the attention mechanism significantly enhances the model's adaptability to dynamic changes in the network. When a new node is added, the attention aggregation mechanism can intelligently evaluate the contribution of the new node, gradually adjust its weights, and avoid drastic fluctuations in global model performance. Specifically, in the scenario of simulating dynamic changes in 30% of nodes, the performance fluctuation of the attention-optimization model was reduced by 62.8% compared to the traditional model, indicating its stronger environmental adaptability. The experimental results demonstrate that the model exhibits superior scalability compared to baseline methods as node numbers increase. However, its performance in real-world vehicle-to-everything (V2X) scenarios with massive scale and high mobility still requires further validation. Future research will focus on developing more efficient communication and computing mechanisms to support practical deployment.

3.4. Discussion on Computational Cost and Scalability

While the proposed AttFL model demonstrates superior accuracy and communication efficiency, its computational overhead and scalability warrant discussion. The primary computational cost stems from the integrated attention mechanisms.

Local Computational Cost: On the client side, the introduction of feature, temporal, and self-attention modules increases the computational load during local training compared to a standard CNN–LSTM model. The self-attention mechanism, in particular, has a complexity of $O(F^2T + FT^2)$ for the feature dimension, F, and a sequence length

of T (Table 1). This is manageable for modern vehicle on-board units (OBUs) with dedicated processing capabilities but could be a constraint on extremely resource-limited devices. Future work will explore model distillation or pruning techniques to create lighter client models.

Server-Side Computational Cost: The adaptive aggregation algorithm requires the central server to calculate attention weights for each client model update based on performance metrics and gradient similarities (Equations (17)–(20)). This introduces additional computation compared to the simple weighted averaging of FedAvg. However, this overhead is minimal relative to the training workload and is justified by the significant gains in convergence speed and model performance, ultimately reducing the total number of costly communication rounds.

Scalability: The system is designed to be scalable. The aggregation process is parallelizable across clients. The attention-weighting mechanism naturally handles a large number of participants by focusing on the most informative updates. However, in a massive IoV network with thousands of vehicles, client selection strategies become crucial in preventing server bottlenecks. The proposed method's ability to achieve high accuracy with fewer aggregation rounds inherently enhances its scalability by reducing the total communication and coordination burden. Nevertheless, performance in ultra-large-scale, highly mobile scenarios requires further validation on more powerful simulation platforms or real-world testbeds.

4. Conclusions

The author proposes a federated learning attack-detection model based on attention mechanism optimization to address the increasingly severe security challenges in the connected car environment. Firstly, the author successfully constructed a basic detection framework that integrates CNN and LSTM and innovatively integrated a self-attention module, effectively improving the model's ability to model spatiotemporal features. The experimental results show that this structure can fully capture the temporal patterns and spatial features in network traffic, and can improve the feature expression ability compared to traditional detection models. Tests on the CICDDoS2019 dataset showed that the introduction of self-attention mechanism improved the accuracy of attack detection by an average of 6.8% compared to the baseline model, especially for the recognition of complex attack types.

Secondly, in response to the global aggregation problem in federated learning, the author designed and implemented an adaptive aggregation algorithm based on attention weights. This algorithm automatically assigns reasonable weights to local models of different vehicle nodes by evaluating the model quality and data representativeness of each node, effectively alleviating the problem of model bias in non-IID data environments. The experimental results show that, compared with the traditional FedAvg algorithm, this algorithm improves the convergence speed of the model by 23.5%, reduces the average aggregation epochs from the original 35 epochs to about 27 epochs, and maintains high accuracy stability.

In summary, the federated learning model based on attention mechanism optimization proposed by the author has achieved comprehensive performance improvement in the field of vehicle networking attack detection. This model not only surpasses existing methods in detection accuracy but also demonstrates significant advantages in convergence speed and communication efficiency. The research results provide a new technological route for safety protection in intelligent transportation systems, demonstrating the enormous potential of combining attention mechanisms with federated learning.

Author Contributions: Conceptualization, L.L. and N.D.; methodology, L.L.; software, L.L.; validation, L.L., F.W. and N.D.; formal analysis, L.L.; investigation, L.L.; resources, F.W.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, F.W. and N.D.; visualization, L.L.; supervision, F.W.; project administration, F.W.; funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: 1. Key Research and Development Plan Project of Liaocheng City. Research on Integrated Wireless Resource Allocation and Control of Autonomous Driving Fleet Based on 5G Technology, (NO.2022 YDSF14). 2. Research and Practice on Optimizing the Output oriented Curriculum System of Electronic Information Engineering of Liaocheng University Dongchang College, (NO.2023JGA02).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Ullah, I.; Deng, X.; Pei, X.; Mushtaq, H.; Khan, Z. Securing internet of vehicles: A blockchain-based federated learning approach for enhanced intrusion detection. *Clust. Comput.* **2025**, *28*, 256. [CrossRef]
2. Huang, K.; Xian, R.; Xian, M.; Wang, H.; Ni, L. A comprehensive intrusion detection method for the internet of vehicles based on federated learning architecture. *Comput. Secur.* **2024**, *147*, 104067. [CrossRef]
3. Xing, L.; Wang, K.; Wu, H.; Ma, H.; Zhang, X. Fl-maae: An intrusion detection method for the internet of vehicles based on federated learning and memory-augmented autoencoder. *Electronics* **2023**, *12*, 2284. [CrossRef]
4. Lu, N.; Cheng, N.; Zhang, N.; Shen, X.; Mark, J.W. Connected vehicles: Solutions and challenges. *IEEE Internet Things J.* **2014**, *1*, 289–299. [CrossRef]
5. Contreras-Castillo, J.; Zeadally, S.; Guerrero-Ibañez, J.A. Internet of Vehicles: Architecture, protocols, and security. *IEEE Internet Things J.* **2017**, *5*, 3701–3709. [CrossRef]
6. Yang, J.; Hu, J.; Yu, T. Federated AI-enabled in-vehicle network intrusion detection for internet of vehicles. *Electronics* **2022**, *11*, 3658. [CrossRef]
7. Li, Y.; Tao, X.; Zhang, X.; Liu, J.; Xu, J. Privacy-preserved federated learning for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8423–8434. [CrossRef]
8. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
9. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
10. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-IID data. *arXiv* **2018**, arXiv:1806.00582. [CrossRef]
11. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
12. Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; Khazaeni, Y. Federated learning with matched averaging. *arXiv* **2020**, arXiv:2002.06440. [CrossRef]
13. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3400–3413. [CrossRef] [PubMed]
14. Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; Poor, H.V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7611–7623.
15. Saez-De-Camara, X.; Flores, J.; Arellano, C.; Urbieta, A.; Zurutuza, U. Clustered federated learning architecture for network anomaly detection in large scale heterogeneous iot networks. *Comput. Secur.* **2023**, *131*, 103299. [CrossRef]
16. Driss, M.; Almomani, I.; e Huma, Z.; Ahmad, J. A federated learning framework for cyberattack detection in vehicular sensor networks. *Complex Intell. Syst.* **2022**, *8*, 4221–4235. [CrossRef]
17. Wu, J.; Qiu, G.; Wu, C.; Jiang, W.; Jin, J. Federated learning for network attack detection using attention-based graph neural networks. *Sci. Rep.* **2024**, *14*, 19088. [CrossRef]
18. Song, X.; Ma, Q. Intrusion detection using federated attention neural network for edge enabled internet of things. *J. Grid Comput.* **2024**, *22*, 15. [CrossRef]

19. Vadigi, S.; Sethi, K.; Mohanty, D.; Das, S.P.; Bera, P. Federated reinforcement learning based intrusion detection system using dynamic attention mechanism. *J. Inf. Secur. Appl.* **2023**, *78*, 103608. [CrossRef]
20. Qu, Z.; Cai, Z. FEDSA-ResnetV2: An Efficient Intrusion Detection System for Vehicle Road Cooperation Based on Federated Learning. *IEEE Internet Things J.* **2024**, *11*, 29852–29863.
21. Alsamiri, J.; Alsubhi, K. Federated learning for intrusion detection systems in internet of vehicles: A general taxonomy, applications, and future directions. *Future Internet* **2023**, *15*, 403. [CrossRef]
22. Djenouri, Y.; Belbachir, A.N.; Michalak, T.; Belhadi, A.; Srivastava, G. Enhancing smart road safety with federated learning for Near Crash Detection to advance the development of the Internet of Vehicles. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108350. [CrossRef]
23. Xu, Q.; Zhang, L.; Ou, D.; Yu, W. Secure intrusion detection by differentially private federated learning for inter-vehicle networks. *Transp. Res. Rec.* **2023**, *2677*, 421–437. [CrossRef]
24. Zhou, H.; Zheng, Y.; Huang, H.; Shu, J.; Jia, X. Toward robust hierarchical federated learning in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5600–5614. [CrossRef]
25. Aouedi, O.; Piamrat, K.; Muller, G.; Singh, K. Federated semisupervised learning for attack detection in industrial internet of things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 286–295. [CrossRef]
26. Bhavsar, M.H.; Bekele, Y.B.; Roy, K.; Kelly, J.C.; Limbrick, D. Fl-ids: Federated learning-based intrusion detection system using edge devices for transportation iot. *IEEE Access* **2024**, *12*, 52215–52226. [CrossRef]
27. Vinita, L.J.; Vetriselvi, V. Federated Learning-based Misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles. *Ad Hoc Netw.* **2023**, *144*, 103153. [CrossRef]
28. Arya, M.; Sastry, H.; Dewangan, B.K.; Rahmani, M.K.I.; Bhatia, S.; Muzaffar, A.W.; Bivi, M.A. Intruder detection in VANET data streams using federated learning for smart city environments. *Electronics* **2023**, *12*, 894. [CrossRef]
29. Nabil, N.; Najib, N.; Abdellah, J. Leveraging artificial neural networks and lightgbm for enhanced intrusion detection in automotive systems. *Arab. J. Sci. Eng.* **2024**, *49*, 12579–12587. [CrossRef]
30. Xie, N.; Zhang, C.; Yuan, Q.; Kong, J.; Di, X. IoV-BCFL: An intrusion detection method for IoV based on blockchain and federated learning. *Ad Hoc Netw.* **2024**, *163*, 103590. [CrossRef]
31. Mansouri, F.; Tarhouni, M.; Alaya, B.; Zidi, S. A distributed intrusion detection framework for vehicular ad hoc networks via federated learning and blockchain. *Ad Hoc Netw.* **2025**, *167*, 103677. [CrossRef]
32. Huang, J.; Chen, Z.; Liu, S.Z.; Zhang, H.; Long, H.X. Improved Intrusion Detection Based on Hybrid Deep Learning Models and Federated Learning. *Sensors* **2024**, *24*, 4002. [CrossRef]
33. Kumar, A.; Gondhi, N.K. Review paper on machine learning based intrusion detection system in internet of vehicles. *Adv. Comput. Sci. Inf. Technol.* **2023**, *10*, 39.