Journal on
Internet of Things

Tech Science Press

DOI: 10.32604/jiot.2022.038914
Article

An Intrusion Detection Scheme Based on Federated Learning and Self-Attention Fusion Convolutional Neural Network for IoT

Jie Deng[1], **Ran Guo**[2] **and Zilong Jin**[1,3,*]

[1]School of Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[2]Cyberspace Institute Advanced Technology, Guangzhou University, Guangzhou, 510006, China
[3]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing, 210044, China
*Corresponding Author: Zilong Jin. Email: zljin@nuist.edu.cn

Received: 03 December 2022; Accepted: 05 January 2023; Published: 14 June 2023

Abstract: Traditional based deep learning intrusion detection methods face problems such as insufficient cloud storage, data privacy leaks, high communication costs, unsatisfactory detection rates, and false positive rate. To address existing issues in intrusion detection, this paper presents a novel approach called CS-FL, which combines Federated Learning and a Self-Attention Fusion Convolutional Neural Network. Federated Learning is a new distributed computing model that enables individual training of client data without uploading local data to a central server. at the same time, local training results are uploaded and integrated across all participating clients to produce a global model. The sharing model reduces communication costs, protects data privacy, and solves problems such as insufficient cloud storage and "data islands" for each client. In the proposed method, a hybrid model is formed by integrating the self-Attention and similar parts of the Convolutional Neural Network in the local data processing. This approach not only enhances the performance of the hybrid model but also reduces computational overhead compared to pure hybrid neural networks. Results from experiments on the NSL-KDD dataset show that the proposed method outperforms other intrusion detection techniques, resulting in a significant improvement in performance. This demonstrates the effectiveness of the proposed approach in improving intrusion detection accuracy.

Keywords: Intrusion detection; self-attention; convolutional neural network; federated learning

1 Introduction

The rapid development of the Internet of Things (IoT) has greatly facilitated the advancement of human life. However, with the growing sensitivity of data privacy, network security has become an increasingly pressing concern. As a result, numerous network security vendors and relevant national agencies have developed their own network intrusion detection systems to address these concerns.

However, these organizations can only do intrusion detection and analysis based on their own data, which has great limitations. There are “data islands” among big data [1], that is, government departments, scientific research departments, and Internet companies each own data but lack mutual data circulation. For example, some companies such as 360 can only do network intrusion detection based on their network security data. Due to privacy protection reasons, it cannot obtain intrusion detection data from network security vendors such as Tencent and Great Wall. The model is not comprehensive enough and has many deficiencies. These network security vendors are unwilling to share their data. In addition to commercial interests, there is another reason that these network security vendors have certain responsibilities to users. Their network security data involves many sensitive information protection issues [2].

Federated Learning (FL) is designed to mitigate data silos and privacy concerns. With FL, models are trained on local data that resides on decentralized devices, and these local models are subsequently aggregated to produce updated global models. By taking this collaborative approach to model training, each local participating device can maintain control over its local data while still benefiting from a shared model that capitalizes on the data strengths of all the clients [3]. FL offers several benefits over traditional centralized machine learning approaches. It minimizes communication costs and reduces the data storage and processing burden on cloud servers. Moreover, it enhances model training efficiency and generalization performance by leveraging diverse data sources from multiple clients.

Intrusion Detection System (IDS) refers to the software and hardware system used to identify abnormal behavior and attack behavior in computers and the Internet and is an important way to ensure network information security. The intrusion detection system plays a crucial role in identifying and preventing network attacks, thereby ensuring the smooth operation of computer systems and networks. By proactively monitoring network traffic data, reviewing user and system behavior, and obtaining system logs in real-time, the intrusion detection system can effectively detect and intercept network attacks promptly, thereby ensuring the smooth and secure operation of computer systems and the Internet. With the development of the Internet, network security issues emerge in an endless stream, IDS is a key technology to solving network security problems. It can realize powerful detection and protection against network attackers. It is of great significance in many fields such as politics, economy, culture, society, and national defense. As a dynamic network security technology, it can detect and block possible malicious network intrusions in real-time, reduce the intrusion and harassment of ordinary netizens by hackers, and alleviate the huge losses or even devastating blows caused by network security for digital and intelligent enterprises. It provides an important weapon for the country to fight against terrorism and criminal interaction, promotes the healthy development of IoT, and maintains social stability.

This paper introduces a novel intrusion detection approach, called CS-FL, which utilizes Federated Learning and Self-Attention Fusion Convolutional Neural Networks. With Federated Learning, the data of each client can be trained separately without uploading local data to a central server. Instead, local training results are uploaded and integrated across all clients to train a shared model. In local data processing, a hybrid model is formed by combining similar parts of the Self-Attention and Convolutional Neural Networks. This hybrid model not only enhances performance but also reduces computational overhead compared to pure hybrid neural networks.

This paper is organized as follows: Section 2 provides a discussion of related work. Section 3 outlines our proposed CS-FL method. Section 4 presents experimental results to demonstrate the effectiveness of our proposed approach. Finally, Section 5 concludes the paper and highlights future directions for this research.

2 Related Works

In 1980, Anderson [4] introduced the concept of intrusion detection, which aims to detect any behavior that may harm Internet of Things (IoT) devices. Anomaly detection is a key method to achieve this goal, and an effective detection algorithm is required to analyze network traffic data. In 1986, Tener [5] developed the Discovery system, which used COBOL on the IBM mainframe to detect abnormal user access to databases and became one of the earliest prototypes for mainframe-based intrusion detection systems (IDS). In 1987, Denning [6] proposed an abstract model for a real-time IDS, the Intrusion Detection Expert System (IDES), which introduced intrusion detection to the security defense of computer systems for the first time. In 1994, Mark Crosbie, Gene Spafford, and others proposed the integration of autonomous agents into the IDS, which showed good efficiency and fault tolerance. This concept is also compatible with other fields of computer science, such as software agents. The proxy method has improved the scalability and maintainability of IDS. In 1995, Houck et al. [7] proposed the Network Fault and Alarm Correlator and Tester (Net FACT) system, which can process incoming alarm streams to minimize network center failures. In 2003, Julisch [8] proposed a new method for detecting intrusion alerts. This method used a new alert clustering method to help researchers identify types of network anomaly detection, and it has shown good performance in experiments. In 2006, Zhang et al. [9] applied the random forest algorithm to the network intrusion detection system, and they constructed the network service model by using the random forest algorithm on network data flow. The algorithm is based on unsupervised learning, which overcomes the problem of label dependence in supervised learning. with the continuous development and improvement of intrusion detection theory and methods, intrusion detection has made rapid progress, but it still faces huge challenges. The accuracy rate and false positive rate of the intrusion detection system are still unsatisfactory, and the attack cannot be detected efficiently. For traffic data, it is difficult to accurately detect new types of attacks in real-time, and the generalization is poor. With the sudden emergence of neural networks, network intrusion detection began to introduce neural network models to make up for the lack of self-detection. In 2006, Yann et al. [10] proposed the theory of deep learning, which opened a new era of artificial intelligence.

Since 2014, more and more researchers have combined deep learning technology with the field of intrusion detection, which has dramatically improved the performance of detection rate and false positive rates. Gao et al. [11] were among the first to introduce deep belief networks into anomaly detection, combining multi-layer restricted Boltzmann machines into a neural network classifier. They applied the deep belief network to the KDD99 dataset and compared it with a support vector machine model, demonstrating superior performance. In 2016, Javaid et al. [12] also applied deep learning methods to anomaly detection, achieving better results in intrusion detection on the NSL-KDD dataset. Similarly, Li et al. [13] utilized Convolutional Neural Networks (CNN) for network intrusion detection, modeling network traffic as a time series and using supervised learning methods to model packets of the TCP/IP protocol within a predetermined time range. The effectiveness of this network structure in intrusion detection was also demonstrated on the KDD99 dataset. In 2018, Al-Qatf et al. [14] presented a deep self-learning framework-based method for feature learning and dimensionality reduction, which effectively enhanced the accuracy of support vector machines in detecting attacks and reduced the network intrusion detection time of the detection system. This deep learning method can be applied to a wide range of intrusion detection systems and has shown promising results in various datasets. With the rapid development of network intrusion detection based on deep learning in China, literature [15] proposed to use of an intrusion detection model composed of two simple convolutional neural networks for model training and converting the original data into two-dimensional images for model training, the experimental results reduce the training time, effectively improve the accuracy and reduce the false positive rate. In 2020, literature [16] proposed a new neural network structure that combines LSTM and RNN, and proposed a voting algorithm to determine whether the traffic is an attack, and achieved high accuracy in terms of detection rate. Koniki et al. [17] introduced a new deep learning intrusion detection model based on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks in 2022. Their proposed model achieved high accuracy and detection rates in identifying various types of attacks on the NSL-KDD dataset. In another study, Hu et al. [18] proposed an improved Convolutional Neural Network (CNN) architecture called Split Convolutional CNN (SPCCNN) on the NSL-KDD dataset. The proposed SPCCNN model outperformed traditional CNN and RNN models by 4.60% and 2.79%, respectively, in terms of detection accuracy.

Compared with the traditional method of centrally uploading the data collected by IoT devices to the cloud service center for processing, the local training model of Federated learning can not only relieve the computing resources of the cloud server, effectively adapt to some changing model systems, but also protect the privacy of local data. It solves the problem of privacy protection and improves the effect of intrusion detection. In 2016, Google proposed the concept of Federated learning [19] and make the TensorFlow Federated framework open-sourced. Domestically, WeBank makes the federated learning framework FATE open-sourced and Baidu makes the PadleFL framework open-sourced. With the rapid development of federated learning, more and more researchers in enterprises and universities have applied federated learning to intrusion detection and made great progress.

Abdel-Basset et al. [20] proposed FED-IDS, a deep learning-based federated learning intrusion detection framework that allows for efficient attack detection by transferring the learning process to distributed edge nodes without central authorization. By implementing FED-IDS on a public dataset (TON_IoT), the authors showed that it can provide secure, reliable, and distributed training. Similarly, Mothukuri et al. [21] proposed a federated learning scheme that uses GRU to detect anomalies and attacks from locally dispersed vehicle data and sends gradients to a central server to aggregate the final model. To preserve privacy, Li et al. [22] proposed the DeepFed Federated learning framework, which encrypts local upload parameters to prevent attackers from inferring local data. Cui et al. [23] introduced a GAN-based private FL approach that adds noise to local gradients to prevent adversaries from inferring any local information. Finally, Liu et al. [24] proposed a gradient compression scheme that reduces communication overhead between federated trainers and cloud servers by reducing parameter communication.

3 Proposed Methods

This section presents a detailed explanation of the method for fusing a Convolutional Neural Network for intrusion detection based on Federated learning and self-Attention. The proposed intrusion detection method is divided into two modules, the federated learning module and the client self-Attention mechanism fused with the convolutional neural network. As shown in Fig. 1 below.

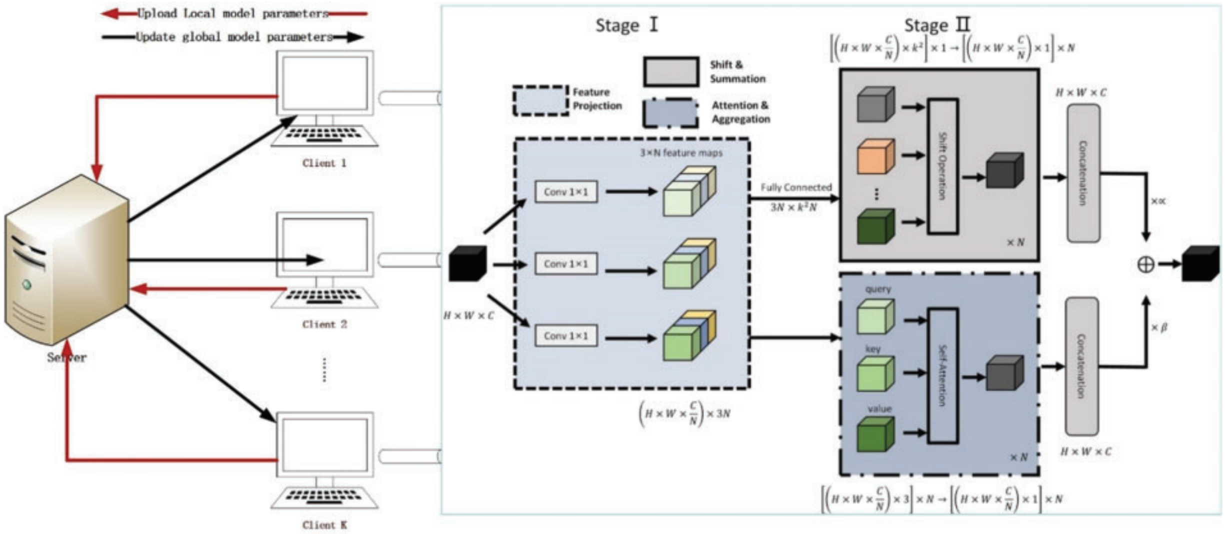
Upload Local model parameters
Update global model parameters
Client 1
Client 2
Client K
Server
Stage I
Stage II
Shift & Summation
Feature Projection
Attention & Aggregation
3×N feature maps
Conv 1×1
Conv 1×1
Conv 1×1
H × W × C
Fully Connected
3N × k²N
$\left(H \times W \times \frac{C}{N}\right) \times 3N$
$\left[\left(H \times W \times \frac{C}{N}\right) \times k^2\right] \times 1 \to \left[\left(H \times W \times \frac{C}{N}\right) \times 1\right] \times N$
Shift Operation
× N
H × W × C
Concatenation
× α
⊕
× β
query
key
value
Self-Attention
× N
Concatenation
H × W × C
$\left[\left(H \times W \times \frac{C}{N}\right) \times 3\right] \times N \to \left[\left(H \times W \times \frac{C}{N}\right) \times 1\right] \times N$

Figure 1: Architecture of CS-FL

3.1 *Self-Attention Mechanism Fusion Convolutional Neural Network*

To address the issues of suboptimal detection and false alarm rates in traditional deep learning models for intrusion detection, this paper proposes a hybrid approach that combines self-attention and convolutional neural networks. Specifically, the data set is preprocessed to form a grayscale image, and the Nth power grayscale image is divided into 1 × 1 input features, and then the output weights are combined by self-Attention mechanism and convolutional neural network respectively. Thereby improving the performance of the detection rate and false positive rate.

3.1.1 Decomposition of Convolutional Neural Network

The standard Convolutional Neural Network formula can be defined by Eq. (1):

$$o_{ab} = \sum\nolimits_{p,q} K_{p,q} i_{a+p-\lfloor k/2 \rfloor, b+q-\lfloor k/2 \rfloor} \tag{1}$$

where $K_{p,q} \in \mathcal{R}^{C_{out} \times C_{in} \times k \times k}$ represents the kernel weights, C_{in}, C_{out} represents the size of the input and output of the channel, k is the kernel size, $p, q \in \{0, 1, 2, 3, \cdots, k-1\}$ indicates where the kernel weights are located, i_{ab} represents input, o_{ab} represents output.

For convenience, Eq. (1) can be simplified as:

$$o_{ab} = \sum\nolimits_{p,q} o_{ab}^{(p,q)} \tag{2}$$

With:

$$o_{ab}^{(p,q)} = K_{p,q} i_{a+p-\left\lfloor \frac{k}{2} \right\rfloor, b+q-\left\lfloor \frac{k}{2} \right\rfloor} \tag{3}$$

To further simplify the formulation, the *Shift* operation is defined, $\tilde{i} \triangleq Shift(i, \Delta x, \Delta y)$, as:

$$\widetilde{i_{a,b}} = i_{a+\Delta x, b+\Delta y}, \forall a, b \tag{4}$$

where Δx, Δy correspond to displacements in the horizontal and vertical directions, respectively, Eq. (3) can be rewritten as:

$$o_{ab}^{(p,q)} = K_{p,q} i_{a+p-\left\lfloor \frac{k}{2} \right\rfloor, b+q-\left\lfloor \frac{k}{2} \right\rfloor} = Shift\left(K_{p,q} i_{ab}, p - \left\lfloor \frac{k}{2} \right\rfloor, q - \left\lfloor \frac{k}{2} \right\rfloor\right) \tag{5}$$

Therefore, the standard Convolutional Neural Network formulation can be divided into two stages:

Stage I:

$$\widetilde{o_{ab}^{(p,q)}} = K_{p,q} i_{ab} \tag{6}$$

Stage II:

$$o_{ab}^{(p,q)} = Shift\left(o_{ab}^{(p,q)}, p - \left\lfloor \frac{k}{2} \right\rfloor, q - \left\lfloor \frac{k}{2} \right\rfloor\right) \tag{7}$$

$$o_{ab} = \sum\nolimits_{p,q} o_{ab}^{(p,q)} \tag{8}$$

Stage I: projects the input features from a specific position (p, q) to the kernel weights, and Stage II shifts them according to the kernel position and finally clusters them together.

3.1.2 Decomposition of Self-Attention Network

The standard Self-Attention Network formula can be defined by Eq. (9):

$$o_{ab} = \Big\|_{l=1}^{N} \sum\nolimits_{m,n \in N_k(a,b)} \left(A\left(W_q^{(l)} i_{ab}, W_k^{(l)} i_{mn}\right) W_v^{(l)} i_{mn}\right) \tag{9}$$

where $\|$ is the concatenation of the outputs of N attention heads, and $W_q^{(l)}, W_k^{(l)}, W_v^{(l)}$are the projection matrices for queries, keys, and values. $N_k(a, b)$ represents a local region of pixels with spatial extent k centered around (a, b), and $A(W_q^{(l)} i_{ab}, W_k^{(l)} i_{mn})$is the corresponding attention weight with regards to the features within $N_k(a, b)$.

The attention weights are computed as:

$$A\left(W_q^{(l)} i_{ab}, W_k^{(l)} i_{mn}\right) = softmax_{N_k(a,b)}\left(\frac{(W_q^{(l)} i_{ab})^T (W_k^{(l)} i_{mn})}{\sqrt{d}}\right) \tag{10}$$

where d is the feature dimension of $W_q^{(l)} i_{ab}$.

Stage I:

$$q_{ab}^{(l)} = W_q^{(l)} i_{ab}, k_{ab}^{(l)} = W_k^{(l)} i_{ab}, v_{ab}^{(l)} = W_v^{(l)} i_{ab} \tag{11}$$

Stage II:

$$o_{ab} = {}_{l=1}^{N}\Bigg\| \sum\nolimits_{m,n\epsilon N_k(a,b)} \left(A\left(q_{ab}^{(l)}, k_{ab}^{(l)}\right) v_{ab}^{(l)}\right) \tag{12}$$

Similar with the decomposed Convolutional Neural Network, in the first stage, convolution is performed first, and the input feature projection is used as queries, keys, and values. In the second stage, the calculation of the attention weight and the aggregation of the mechanism matrix collects local features.

Specifically, the hybrid approach proposed in this paper consists of two stages. In the first stage, the input features are projected through 3 convolutions and re-segmented into N segments respectively. Therefore, we can get $3 \times N$ feature maps. The second stage aggregates weights according to a different paradigm. For the Self-Attention module, $3 \times N$ features are divided into N groups on average, where each group contains 3 features, and each feature comes from 1×1 convolution, corresponding to three feature map queries, keys, and values, following the tradition The multi-head self-attention (Eq. (12)) generates corresponding weights. For the convolutional module, new feature weights are generated by transformation and aggregation via Formulas (7) and (8). The sum of the outputs of the last two paths is given by Eq. (13).

$$F_{out} = \alpha F_{att} + \beta F_{conv} \tag{13}$$

3.2 Federated Learning Modules

Federated averaging (FedAvg) is a widely used optimization algorithm in Federated learning. It is the standard algorithm that employs local stochastic gradient descent to achieve model aggregation in a central federated server using an average update method.

The Federated Averaging (FedAvg) algorithm starts with the federated server distributing the initial model parameters to each client. Each client then performs local training using the model parameters and uploads the trained local model parameters to the federated server. The federated server aggregates and updates the model using these local model parameters and downloads the updated model for further training. This process continues iteratively until convergence is achieved.

The local client $k$calculates the batch gradient descent method g_k during the r round of global model parameter update. The local model parameters W_{r+1}^{k} are obtained from Eq. (14), where η is the learning rate.

$$W_{r+1}^{k} = W_r - \eta g_k \tag{14}$$

After obtaining the local model parameters, each participant uploads them to the federation server for aggregation. The global model parameters for the next round are obtained by using Eq. (15), where N represents the total amount of data of all local devices, and n_k represents the data amount of local device.

$$\overline{W}_{r+1} = \frac{n_k}{N} \sum\nolimits_{k=1}^{K} (W_{r+1}) \tag{15}$$

3.3 Based on Federated Learning and Self-Attention Fusion Convolutional Neural Network

This paper proposes a Federated learning and Self-Attention based fusion method for intrusion detection using Convolutional Neural Networks. The algorithm flow is depicted in Fig. 2, considering the characteristics of the two modules.

The algorithm steps are as follows:

Input: the quantity of clients N, the quantity of global iterations K, the starting global model parameters W_g^0 and the predetermined number of communication rounds R.

Output: Global model parameters W_g obtained from each round of training.

Step 1: Set the communication round counters $r = 0$ and begin training.

Step 2: Distribute the global model parameters W_g^r of this round to N local clients for training their local classifier model.

Step 3: N local clients combine their local model parameters according to the global model parameters and output the new local model parameters. Then, the local model parameters are sent to the server by the N participants through two paths.

Step 4: Update global model parameters using aggregation functions.

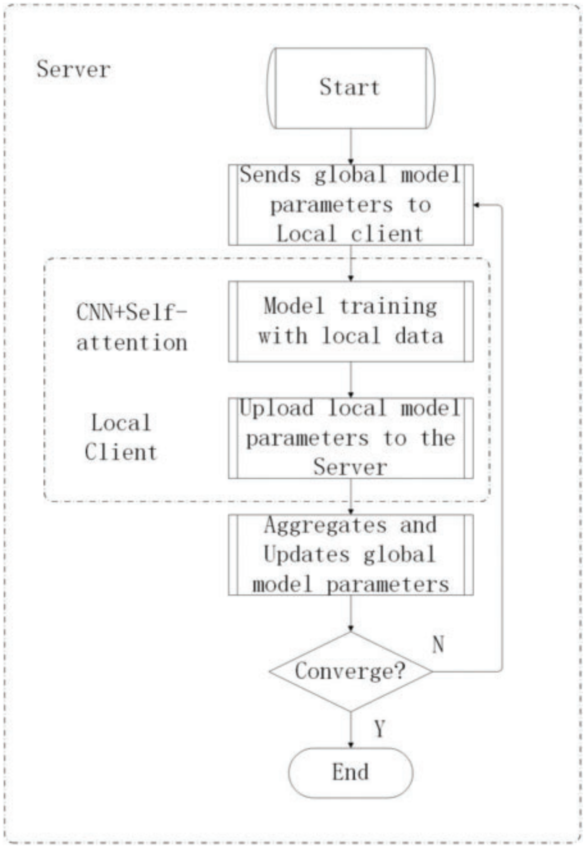
Server
Start
Sends global model parameters to Local client
CNN+Self-attention
Model training with local data
Local Client
Upload local model parameters to the Server
Aggregates and Updates global model parameters
Converge?
N
Y
End

Figure 2: The algorithm of CS-FN

4 Experimental Analyses

4.1 Datasets

The experiments described in this paper were conducted using the NSL-KDD dataset as the experimental dataset, which includes a training set and a test set. The training set contains 125973 data, while the test set contains 22544 data. The NSL-KDD dataset includes 22 attack types in the training set, 39 attack types in the test set, and 17 network attack types that are not present in the training set. Each piece of network traffic data in the NSL-KDD dataset contains 42-dimensional features, including 38-dimensional numerical features, 3-dimensional character features, and 1-dimensional label features. The attack types in the NSL-KDD dataset are divided into four types: DoS, Probe, R2L, and U2R. Tables 1 and 2 show the data category, quantity, and proportion of the training set and test set used in the experiment. The different types of attacks are briefly described as follows:

Table 1: Information of the training set

Category	Quantity	Proportion%
Normal	67343	53.46
DoS	45927	36.46
Probe	11656	9.25
R2L	995	0.79
U2L	52	0.04

Table 2: Information of the test set

Category	Quantity	Proportion%
Normal	9711	43.08
DoS	7458	33.08
Probe	2421	10.74
R2L	2754	12.22
U2L	200	0.89

DoS (Denial of Service) is a type of attack that aims to disrupt the normal traffic flow to and from a targeted system. The IDS may become overloaded by abnormal traffic that the system cannot handle, leading to a shutdown to protect itself. This blocks normal traffic from accessing the network. For example, an online retailer may be flooded with online orders on a busy sale day, and if the network cannot handle all the requests, it may shut down, preventing legitimate customers from making purchases. This is the most prevalent attack type in the NSL-KDD dataset.

The probe is an attack that attempts to obtain information from a network. The goal here is to gather information about the targeted system or network without disrupting its operation. This type of attack is similar to a thief casing a building to gather information before attempting a break-in. The attacker tries to identify vulnerabilities or weaknesses that can be exploited in the later stages of an attack.

R2L (Remote-to-Local) attacks are attempts to gain unauthorized access to a remote machine in a network by exploiting vulnerabilities in the remote access mechanisms. This type of attack involves an attacker who does not have local access to the targeted machine but tries to gain access by exploiting vulnerabilities in the network communication protocols, services, or applications. Once the attacker gains access, they can potentially perform other types of attacks, such as stealing sensitive data or installing malware on the system.

U2R is a type of attack that begins with a regular user account and aims to obtain superuser/root access to a system or network. Attackers exploit system vulnerabilities to gain elevated privileges and access.

4.2 Evaluation Measure

In this paper, the experimental results are evaluated and analyzed by the indicators commonly used in network anomaly detection such as accuracy, precision, and recall. These indicators can be used for true positive (TP), false positive (FP), true negative (TN), and false negative (FN) 4 metrics to represent.

- TP: Indicates traffic that is correctly classified as attack.
- TN: Indicates traffic that is correctly classified as normal.
- FP: Indicates traffic that is incorrectly as attack.
- FN: Indicates traffic that is incorrectly as normal.

Accuracy, Precision and Recall are defined as follows:

$$Accuracy = \frac{\mathrm{n_{TP}} + \mathrm{n_{TN}}}{\mathrm{n_{TP}} + \mathrm{n_{FP}} + \mathrm{n_{TN}} + \mathrm{n_{FN}}} \tag{16}$$

$$Precision = \frac{n_{TP}}{n_{TP} + n_{FP}} \tag{17}$$

$$Recall = \frac{n_{TP}}{n_{TP} + n_{FN}} \tag{18}$$

n_{TP}, n_{TN}, n_{FP}, n_{FN} in Eqs. (16)–(18) respectively represent the number of true positive data, the number of true negative data, and false positive data The number of records and the number of false negative data records.

4.3 Model Training and Testing Results

The experiment in this paper explores the performance of the CS-FN model proposed in this paper on the data set. At the same time, in order to compare with other models, a comparative experiment is also done, comparing the performance of the CS-FN model with random forest, C.45 decision tree, vector Machine, RNN and other models were compared.

Fig. 3 shows the classification effect of the CS-FN model on the test set and training set after 50 epochs. Obviously, the CS-FN model has a high detection accuracy rate, among which the accuracy rate on the training set is 97.52%, and the accuracy rate on the test set is 92.21%. Compared with the random forest, C.45 decision tree, vector machine, RNN, and other models proposed by previous researchers, the comparison results on the same data set are shown in Fig. 4. Obviously, in this experiment, the performance of the CS-FN model is better than other classification algorithms.

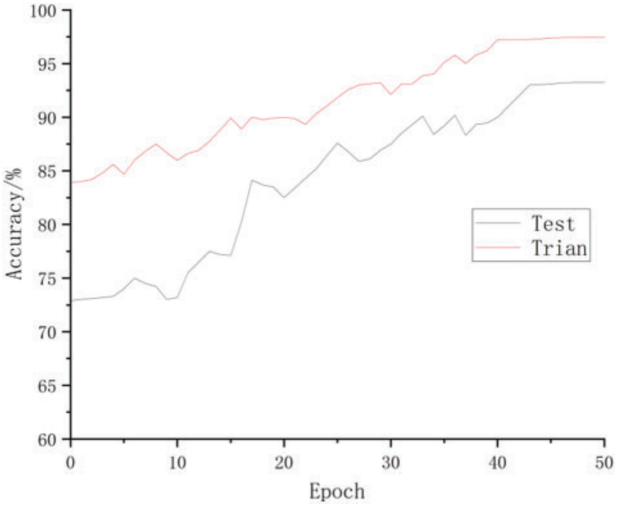

Figure 3: Detection accuracy on training set and test set

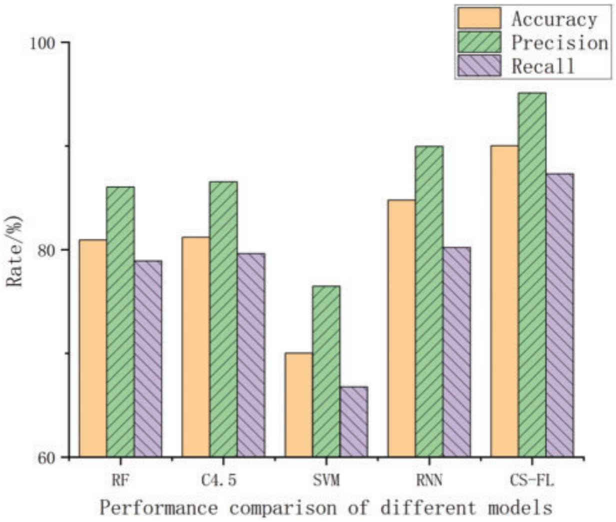

Figure 4: Performance comparison of different models

5 Conclusion and Future Work

Aiming at the problems of insufficient cloud storage, data privacy leakage, and low detection rate faced by current network intrusion detection methods, an intrusion detection method based on federated learning and Self-Attention fusion convolutional neural network is proposed. In the CS-FN model, the local client does not need to share the local training data, but only needs to upload the result parameters of the local classifier training and update the global model under the federated server. In the experiments, it performs well on the test set and training set respectively. Compared with other traditional intrusion detection methods, it has higher Accuracy, Precision, and Recall under the NSL-KDD data set. Its performance is better than that of traditional classification methods. The detection accuracy does not decrease while improving the performance of the model. The proposed CS-FN model in this study exhibits robust capabilities in intrusion detection, demonstrating excellent performance in the classification and recognition of network attacks. Furthermore, it effectively addresses the "data island" challenge in attack detection, while preserving the data privacy of local clients. This study introduces a promising research direction in the field of detection technology.

Although the model proposed in this paper has made great progress in intrusion detection, there are still many shortcomings. In future work, we will further in-depth research on federated learning, improve the high precision and generalization capabilities of federated learning, reduce the communication delay between local clients, and improve the performance of federated learning. At the same time, there is ongoing research to explore more deep learning algorithms combined with federated learning models to further enhance performance in the field of intrusion detection. It is also hoped that intrusion detection experiments can be carried out in real environments in the future so that the model can achieve better results in real-time attack detection.

Funding Statement: This work was sponsored by the National Natural Science Foundation of China under Grants 62271264, 61972207, and 42175194, and the Project through the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institution.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. J. Yu, J. P. Queralta, J. Heikkonen and T. Westerlund, "Federated learning in robotic and autonomous systems," *Procedia Computer Science*, vol. 191, no. 6, pp. 135–147, 2021.

[2] L. Cui, Y. Y. Qu and G. Xie, "Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3492–3500, 2022.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li *et al.,* "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[4] J. P. Anderson, "Computer security threat monitoring and surveillance," Technical Report, James P. Anderson Company, 1980.

[5] W. Tener, "Detection of control deterioration using decision support systems," *Computers & Security*, vol. 5, no. 4, pp. 290–295, 1986.

[6] D. E. Denning, "An intrusion-detection model," in *IEEE Symp. on Security and Privacy*, Oakland, CA, USA, pp. 118, 1986.

[7] K. Houck, S. Calo and A. Finkel, "Towards a practical alarm correlation system," in *Integrated Network Management IV: Proc. of the Fourth Int. Symp. on Integrated Network Management*, US, Springer, pp. 226–237, 1995.

[8] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," *ACM Transactions on Information & System Security*, vol. 6, no. 4, pp. 443–471, 2003.

[9] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection," in *IEEE Int. Conf. on Communications*, Istanbul, Turkey, pp. 2388–2393, 2006.

[10] L. C. Yann, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] N. Gao, L. Gao, Q. L. Gao and H. Wang, "An intrusion detection model based on deep belief networks," in *Second Int. Conf. on Advanced Cloud and Big Data*, Huangshan, China, pp. 247–252, 2014.

[12] A. Javaid, Q. Niyaz, W. Sun and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. of the Int. Conf. on Bio-Inspired Information and Communications Technologies*, Toledo, USA, University of Toledo, pp. 21–26, 2016.

[13] Z. Li, Z. Qin, K. Huang, X. Yang and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Proc. of the Int. Conf. on Neural Information Processing*, Guangzhou, China, pp. 14–18, 2017.

[14] M. Al-Qatf, L. H. Yu, M. Al-Habib and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[15] S. Z. Zhang, X. Y. Xie and Y. Xu, "An intrusion detection method based on a deep convolutional neural network," *Journal of Tsinghua University*, vol. 59, no. 1, pp. 44–52, 2019.

[16] F. Jiang, Y. Fu, B. B. Gupta, Y. Liang, S. Rho *et al.,* "Deep learning based multi-channel intelligent attack detection for data security," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 204–212, 2020.

[17] R. Koniki, M. D. Ampapurapu and P. K. Kollu, "An anomaly based network intrusion detection system using LSTM and GRU," in *2022 Int. Conf. on Electronic Systems and Intelligent Computing (ICESIC)*, Chennai, India, pp. 79–84, 2022.

[18] Z. Q. Hu, L. J. Wang, Q. Li, Y. Li and W. Z. Yang, "A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network," *IEEE Access*, vol. 8, pp. 195741–195751, 2020.

[19] J. Konecný, H. B. McMahan, D. Ramage and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *ACM Transactions on Intelligent Systems and Technology*, 2016. https://doi.org/10.48550/arXiv.1610.02527

[20] M. Abdel-Basset, N. Moustafa, H. Hawash, I. Razzak, K. M. Sallam *et al.,* "Federated intrusion detection in blockchain-based smart transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2523–2537, 2022.

[21] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha *et al.,* "Federated learning-based anomaly detection for IoT security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021.

[22] B. Li, Y. Wu, J. Song, R. Lu, T. Li *et al.,* "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[23] L. Cui, Y. Qu, G. Xie, D. Zeng, R. Li *et al.,* "Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3492–3500, 2022.

[24] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong *et al.,* "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.