International Journal on Semantic Web and Information Systems
Volume 20 • Issue 1

A Network Intrusion Detection Method for Information Systems Using Federated Learning and Improved Transformer

Qi Zhou, Guangdong Open University, China*
Zhoupu Wang, China Telecom Sichuan Branch, Chengdu, China

ABSTRACT

A network intrusion detection method for information systems using federated learning and improved transformer is proposed to address the problems of long detection time and low security and accuracy when analyzing massive data in most existing intrusion detection methods. Firstly, a network intrusion detection system is constructed based on a federated learning framework, and the transformer model is used as its universal detection model. Then, the dataset is divided and an improved generative adversarial network is used for data augmentation to generate a new sample set to overcome the influence of minority class samples. At the same time, the new samples are input into the transformer local model for network attack type detection and analysis. Finally, the authors aggregate the detection results of each local model and input them into the Softmax classifier to obtain the final classification prediction results.

KEYWORDS

Deep Learning, Federated Learning, Improve the Generation of Adversarial Networks, Network Intrusion Detection, Softmax Classifier, Transformer Model

1. INTRODUCTION

In the rapidly developing network environment, network security issues are constantly emerging. As an important measure to monitor potential network attacks, network intrusion detection (NID) needs to quickly and accurately identify attack events in a massive data environment (Vitorino, Praça, & Maia, 2023; Usoh, et al., 2023). Therefore, improving the accuracy and efficiency of network intrusion detection (NID) technology is of great practical significance (Krishna, et al. 2021).

Considering the complexity of network traffic and the development of computer technology, traditional ID methods have shortcomings in detecting attacks and have low detection efficiency (Wang, et al., 2023; Stergiou, et al., 2021; Devi, & Bharti, 2022). At present, various machine learning (ML) based NID methods have been proposed, and due to the ability of deep learning (DL) to learn complex patterns from high-dimensional data, it has become a suitable solution for detecting network attacks (Deore, & Bhosale, 2023; Mustafa, et al., 2023; Zhang, et al., 2023). ML and DL

DOI: 10.4018/IJSWIS.334845 *Corresponding Author

can be widely applied in ID, mainly due to the availability of collected network data, which can be used to train intrusion detection models. The development of technology has enhanced the computing power of devices, enabling faster training of data models while reducing costs, and the widespread application of DL ensures the accuracy of model optimization on the basis of self-learning. Although ML and DL have improved the detection accuracy, in reality, network intrusion data is limited and insufficient to train high-quality models with good performance (Yan, et al., 2023; Gaurav, et al. 2023). At the same time, there are still some issues with current intrusion detection methods: (1) users need to upload their data to a central entity to train the central model, but about 90% of the central entities will be attacked, resulting in poor security; (2) the performance of the system will decrease with the increase of user size, and single point of failure will be introduced, which will affect the integrity of services and the quality of the model; (3) traditional intrusion detection systems adopt a centralized processing mode, which is time-consuming and difficult to meet the current needs for fast and accurate detection.

The distributed machine learning framework - federated learning (FL), can effectively solve the above problems by implementing DL models in a distributed environment for training on datasets on different devices (Idrissi, et al., 2023; He, & Zhao, 2022). This can improve the efficiency of data feature extraction and learning while ensuring the privacy of terminal data for participants. To this end, a NID method for information systems is proposed based on FL and DL. The innovation of the proposed method is as follows:

1) To improve the processing efficiency and data security of massive data, the proposed method utilizes a FL framework for multi-server collaboration, which shortens training time.
2) Due to the small number of abnormal data samples, which directly affects the detection accuracy of the model, the proposed method utilizes an improved generative adversarial network for data augmentation to reduce the impact of minority class samples, while utilizing the Transformer model to ensure the reliability of detection.

2. RELATED RESEARCH

Traditional intrusion detection methods are based on fixed or dynamic rules to identify attacks on the network (Sawsan, et al., 2020). However, attackers use various techniques to disguise their attacks and disrupt the target's defense system (Xu, et al., 2021; Singh, & Gupta, 2022; Wang, et al., 2022). Therefore, ML algorithms were first widely used to detect anomalies in networks and have been proven to provide high detection rates. Supervised ML includes methods such as naive Bayesian classifiers (Ma., & Ding, 2022; Sharma, & Sharma, 2022). Supervised learning algorithms require classification and labeling of data, while unsupervised learning algorithms do not. Unsupervised algorithms include clustering, K-means, deep neural networks, etc. As shown in Zhang, & Wang, (2023), the use of feature engineering methods and synthetic minority class oversampling (SMOTE) technology to process network data can effectively reduce feature redundancy and alleviate the attack detection problem of class imbalance. At the same time, Catboos classifier is used to achieve network intrusion detection. Mohy, et al., (2023), propose a NID model for IoT environments based on KNN classifier and feature selection and utilize genetic algorithm for parameter optimization to ensure good detection performance. Maidamwar, et al., (2023), use a multi-layer perceptron classifier and a random forest algorithm to complete network intrusion detection. Layeghy, et al., (2023), propose using adversarial domains to extract domain invariant features from multiple network domains and apply unsupervised techniques for anomaly recognition, i.e. training One Class SVM (OSVM) models to detect network anomalies. The above methods, which rely solely on typical machine learning methods to construct intrusion detection methods, are no longer able to resist increasingly complex and diverse network threats. Therefore, there is a strong need for effective intrusion detection methods.

In the past few years, DL has also been widely used in the field of NID (Aryandoust, et al., 2020; Tembhurne, et al., 2022). DL based NID methods do not rely on feature engineering due to their deep structure. As proposed in Harini, et al., (2023), an intrusion detection technology with a three-layer structure is proposed, which includes Weighted Deep Neural Network (WDNN), Long-Short Term Memory Network (LSTM), and XGBoost algorithm. At the same time, a single side selection undersampling algorithm is used to remove noise samples from most class attacks to improve the detection rate of minority class attacks. The detection results are accurate, but the model is complex. The detection takes a long time. Yao, et al., (2023), propose a lightweight intelligent intrusion detection method using single class bidirectional GRU (BiGRU) autoencoder and ensemble learning (EL), in which soft voting is used to evaluate the results of various base classifiers, making anomaly classification more accurate. Priya, & Ponmagal, (2023) use genetic algorithms to optimize automatic encoders for classification and recognition of service based attacks in cloud systems. Song, et al., (2023), propose a NID model based on Time Convolutional Network (TCN), BiGRU. The feature vectors obtained from TCN and BiGRU are fused and inputted into the self-attention mechanism for analysis to further enhance the detection ability of the model. However, the model did not consider the influence of minority class samples, and its universality was poor. Wang., Xu., & Liu., (2023) propose an intrusion detection model (Res-Tran BiLSTM) using ResNet, Transformer, and BiLSTM, and synthesized the Small Overlap Technique (SMOTE) - Edit Nearest Neighbor (ENN) method to alleviate the degree of data imbalance. This method has high detection reliability, but the model parameters are large, making it difficult to quickly process massive network data. The comparison of different intrusion detection technologies is shown in Table 1.

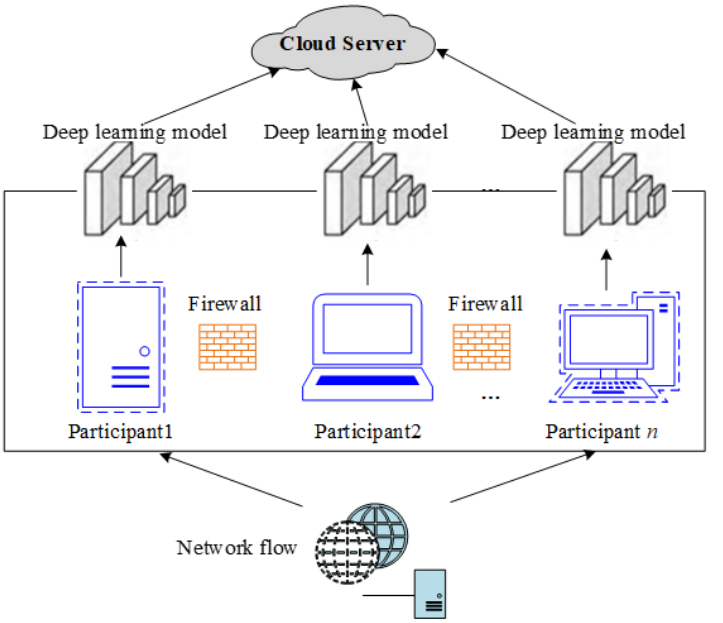
3. SYSTEM MODEL

Intrusion detection system is an active defense security tool that combines hardware and software (Zhang, & Zhang, 2022). It ensures security by disrupting attackers' access to information or preventing them from further accessing network systems, and can effectively resist network attacks (Lu, et al., 2021; Mishra, et al., 2022). Due to the large amount of data in intrusion detection and the extremely uneven distribution of network attack types, a federated deep learning based NID system is proposed to improve its detection efficiency and accuracy. The overall architecture of the system model is shown in Fig. 1, which includes a public cloud server and n learning participants. In this system, trusted institutions are used to distribute keys, and the cloud server and learning participants use additive homomorphic encryption during communication to protect the privacy of intermediate results and ensure the security of network data.

Table 1. Comparison of different intrusion detection technologies

Method	Technological Innovation	Limitation
SMOTE-Catboos (Zhang, & Wang, 2023)	Reduce feature redundancy and alleviate class imbalance.	The model is simple and has a small application range
OSVM (Layeghy, et al., 2023)	Extract domain invariant features from multiple network domains using adversarial domains and apply unsupervised techniques for anomaly recognition.	Lack of consideration for data imbalance and distributed analysis.
BiGRU -SL (Yao, et al., 2023)	Implement lightweight detection using BiGRU autoencoder and integrated learning.	The detection effect of small sample data is poor.
Res-Tran BiLSTM (Wang, Xu, & Liu, 2023)	Combining ResNet, Transformer, and BiLSTM models to address data imbalance issues.	The model has large parameters and long training time.

Figure 1. Overall framework of the proposed method



Among them, cloud servers can coordinate participants and aggregate their encrypted models, while leveraging their powerful computing power to achieve precise analysis of detection models. Through multiple rounds of interaction with participants, cloud servers can build more comprehensive network intrusion detection models.

Each learning participant is an entity trained by the system. Before training, a trusted institution generates a key pair and synchronizes it to all participants through a secure channel. The participants jointly established the public key pk and private key sk required in the additive homomorphic encryption scheme (Durga Prasa, et al., 2019). Then, each participant conducts DL and training on their local data to obtain a local NID model and encrypts the model with a public key pk before sending it to the server. The server aggregates the ciphertext and sends it to the participants, who then use sk to decrypt the aggregation model received from the server. The final learning participants can achieve network intrusion detection tasks locally through the obtained global model. The system parameter definitions are shown in Table 2.

Table 2. The system parameter definitions

Parameter	Define	Parameter	Define
M_{MHA}	Multi head attention function	L_d	Discriminator loss calculation function
ω	Weight	$\varphi_k\left(\tau\right)$	Local objective function
softmax	Multi class activation function	$\phi_k\left(\tau;\tau_t\right)$	Objective Function of Federated Learning

4. A GENERAL NETWORK INTRUSION DETECTION MODEL BASED ON DL

4.1 Transformer Model

The Transformer follows the commonly used encoder decoder structure, as shown in Fig. 2. Among them, in order to prevent network degradation and accelerate convergence, the Transformer encoder uses residual connections and layer normalization internally, while adding multi head attention and feedforward network (FFN) modules (Oliveira, H.S., & Oliveira, H.P., 2023).

4.1.1 Multiple Attention Mechanism

The calculation of the multi head attention function (M_{MHA}) is as follows:

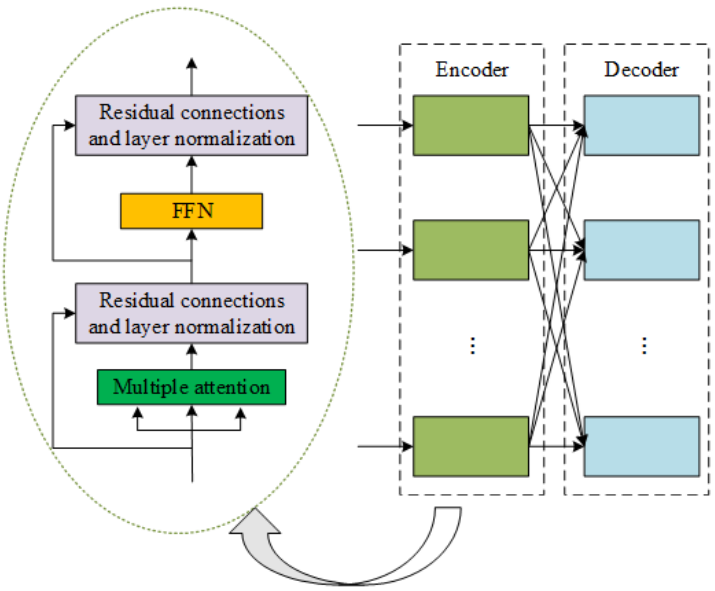
$$M_{MHA}\left(Q,K,V\right)=Concat\left(h_1,h_2,\cdots,h_{\Xi}\right)\omega^o \tag{1}$$

Where, Q, K, and V represent the query, key, and value of attention, respectively; Ξ represents the total number of heads; ω^o is the weight matrix; Concat is a vector concatenation operation; h_i is the feature of the i-th head, and $i\in\left\{1,2,\cdots,\Xi\right\}$. Based on the scaled dot product attention, h_i is obtained as follows:

$$h_i=A_{att}\left(Q\omega_i^Q,K\omega_i^K,V\omega_i^V\right) \tag{2}$$

$$A_{att}\left(Q,K,V\right)=soft\max\left(QK^T/\sqrt{d_k}\right)V \tag{3}$$

Figure 2. Architecture of the transformer model



Where, ω_i^Q, ω_i^K and ω_i^V are the weight matrices corresponding to the i-th input; d_k is the feature dimension; Softmax is the activation function used for multi classification. $\sqrt{d_k}$ can scale the dot product to prevent the resulting dot product from being too large.

4.1.2 FFN

The function of FFN is to prevent degradation of the Transformer model output, mainly composed of two linear transformations, which use the ReLU activation function (Zhang, Feng, & Huang, 2022). The calculation of the FFN operation function is as follows:

$$F_{FFN}\left(x\right) = \max\left(0, x\omega_1 + b_1\right)\omega_2 + b_2 \tag{4}$$

Where, ω_1 and ω_2 are weight matrices; b_1 and b_2 are bias terms; x is the input of FFN; max represents the maximum value operation.

4.2 Model Training

Due to the imbalance of data samples, it is necessary to use generative adversarial networks (GAN) to generate minority class samples and complete data sample expansion before using the Transformer model for network intrusion detection (Zhang, et al., 2021). In order to generate minority class samples more accurately, the proposed method uses an Exchange-GAN network.

The Exchange-GAN network generator includes feature extractors and feature synthesizers (Yang, et al., 2020). The feature extractor consists of a convolution module and a residual module, used to extract the features of the input image. The discriminator of Exchange GAN needs to distinguish the authenticity of samples and includes auxiliary classifiers. In addition to the shared convolutional module, the auxiliary classifier also includes convolutional layers and fully connected layers.

Through adversarial training between generators and discriminators in Exchange-GAN, the model's ability to extract features and the discriminator's recognition ability are improved. The adversarial loss L_{adv} calculation between the generator and discriminator is as follows:

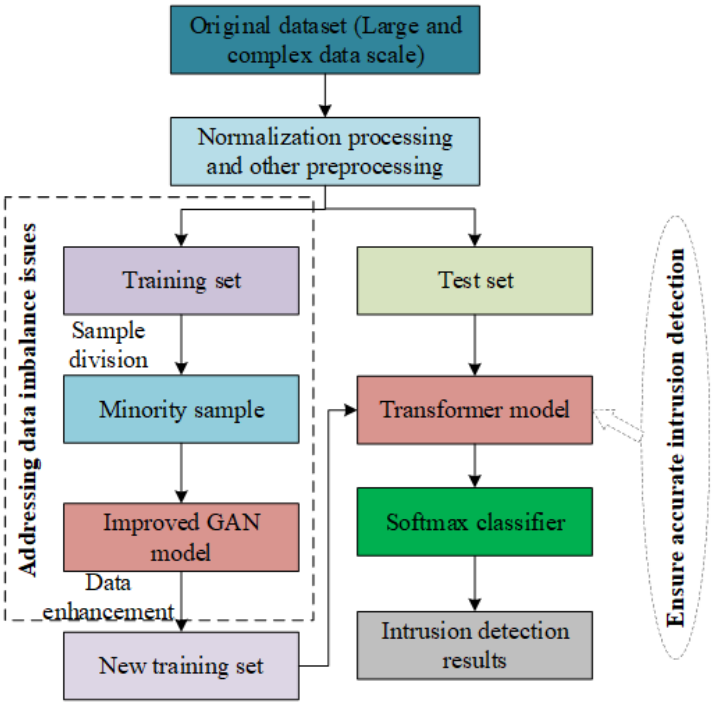
$$\begin{aligned} L_{adv}\left(\theta_f, \theta_h, \theta_d\right) = & \sum_{i=1,2,\cdots,N}\left(1 - L_d\left(G_d\left(x_i; \theta_d\right)\right)\right) \\ & \sum_{i=1,2,\cdots,N} L_d\left(G_d\left(G_h\left(G_f\left(x_i; \theta_f\right); \theta_h\right); \theta_d\right)\right) \end{aligned} \tag{5}$$

where, x_i is a real image; G_f, G_h and G_d are feature extractors, feature synthesizers, and discriminators, respectively; θ_f, θ_h and θ_d are parameters of G_f, G_h and G_d, respectively; L_d is the discriminator loss calculation function.

Input the expanded new dataset into the Transformer model for feature extraction and complete Softmax classification. The entire NID process is shown in Fig. 3.

(1) Normalize network intrusion data to obtain a standardized raw dataset and divide it into training and testing sets.
(2) Network intrusion data includes both normal and attack data, but in general, the amount of attack data is much smaller than the amount of normal behavior data. There is also a category imbalance in attack data. By data partitioning, the training set is divided into minority class samples and other class samples.

Figure 3. NID process based on improved transformer model



(3) A few class samples in the training dataset are enhanced through improved GAN. Integrate the new data samples generated by GAN with the original data samples to obtain a new balanced training set.
(4) Train the Transformer model using a new training set, and test the model using the test set to achieve feature extraction of intrusion data. Finally, the extracted features are classified using a Softmax classifier to obtain detection results, thereby achieving NID.

5. A NID METHOD BASED ON FL

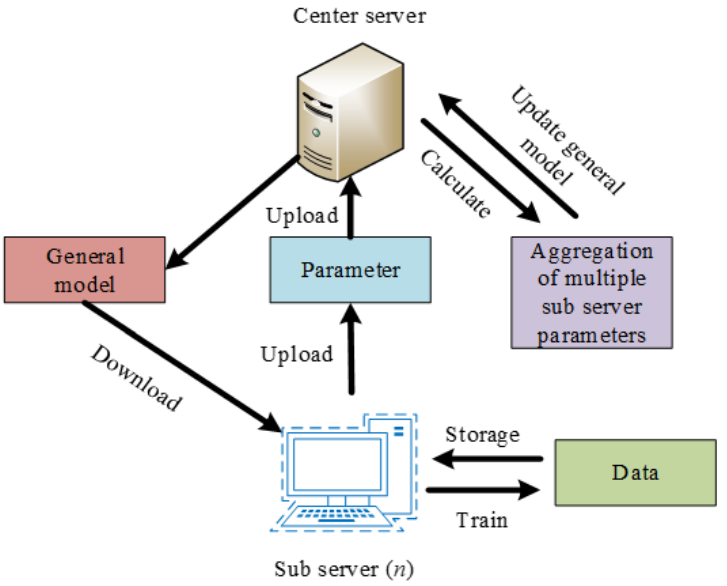
5.1 FL

FL is the establishment of ML models. Under the FL framework, each device does not send raw data for processing, only sending updated model parameters for aggregation. The FL framework is shown in Fig. 4.

In the FL framework, each server is relatively independent and does not interfere with each other, and the processing permissions for local data are not controlled by the central server. Each sub server independently trains and tests local data by downloading a universal model and then uploading the local parameter updates to the central server. Each data owner can train without providing their own data, effectively ensuring the data security of the data owner. Under the FL mechanism, multiple sub servers can jointly maintain the latest global model, which is the most important component module in the FL framework. The universal model used in the proposed method is the Transformer model.

Each selected device performs local calculations based on the current global model and its local dataset and sends updates to the central server. The central server utilizes these updates to obtain a new global model to minimize the loss function:

Figure 4. Architecture of federated learning



$$\min_{\tau} f(\tau) = \sum_{k=1}^{N} p_k \varphi_k(\tau) = E\left[\varphi_k(\tau)\right] \tag{6}$$

where, τ is the parameter of the given model; N is the number of devices; $p_k \geq 0$, and $\sum_k p_k = 1$; n_k is the local data sample corresponding to each device k, and n is the total number of samples $n = \sum_k n_k$, therefore $p_k = n_k / n$; $\varphi_k(\tau)$ is the local objective function of device k.

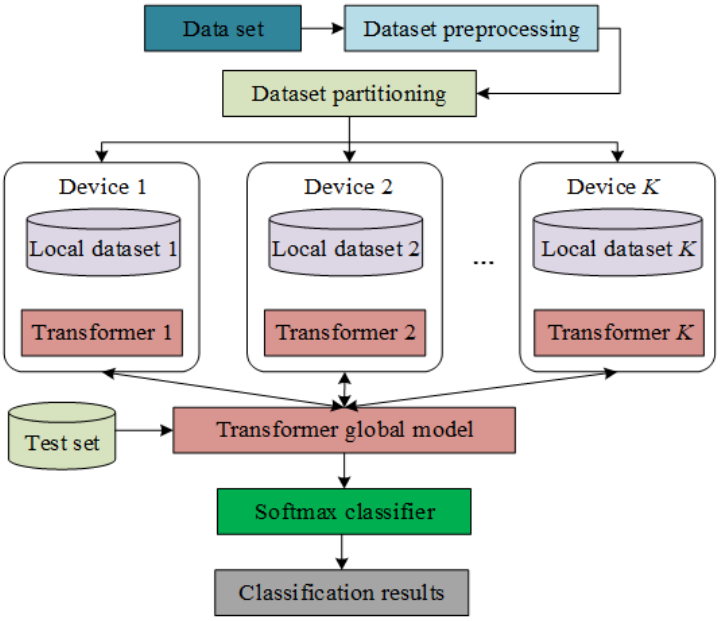
Usually, the federated average algorithm is used for solving, and the Stochastic Gradient Descent (SGD) method is chosen as the local solver. The learning rate and local iteration number on each device are set to be the same (Chen, et al., 2022; Wang, et al., 2023). In each round of local iteration, the number of selected devices is much smaller than the total number of devices K, and the local data of the selected devices is calculated. Then, the global model averages it and updates it.

5.2 FL Process for Intrusion Detection

Most existing intrusion detection methods are centralized learning methods, which input a large amount of data into the model for training, and the performance of the model is easily affected (Ling, & Hao, 2022; Ling, & Hao, 2022). To this end, a Transformer based federated deep learning network intrusion detection method is proposed, and its detection process is shown in Fig. 5.

The proposed model first divides the original dataset of intrusion detection into several local datasets, which correspond to the federated learning devices, one by one. The devices train the local Transformer model based on their local datasets, which not only avoids the impact of excessive data volume, but also reduces communication overhead to a certain extent. Secondly, the model also adopts a dynamic local iteration method, which can effectively avoid differences between models and improve the convergence speed of the model while ensuring accurate classification. In addition, in order to avoid the impact of excessive differences, a proximal term was added to the model during local training. The objective function containing the proximal term is $\phi_k(\tau;\tau_t)$, and its minimization calculation $\min_{\tau} \phi_k(\tau;\tau_t)$ is as follows:

Figure 5. Intrusion detection process based on transformer-federated deep learning



$$\min_{\tau} \phi_k\left(\tau;\tau_t\right) = \varphi_k\left(\tau\right) + \frac{\mu}{2}\left\|\tau - \tau_t\right\|^2 \tag{7}$$

$$\begin{aligned} &\left\|\nabla\phi\left(\tau^*;\tau_t\right)\right\| \le \gamma_k^t \left\|\nabla\phi\left(\tau_k;\tau_t\right)\right\| \\ &s.t. \nabla\phi\left(\tau;\tau_t\right) = \nabla\varphi_k\left(\tau\right) + \mu\left(\tau - \tau_t\right) \end{aligned} \tag{8}$$

where, μ is the scaling factor; $\gamma \in \left[0,1\right]$, γ_k^t is used to determine how many iterations device k needs to perform in the t-th cycle.

Next, the local model trained on the local dataset is transmitted to the central server, and the local models obtained from device training are aggregated to form the optimal Transformer global model. Finally, input the aggregated data into the Softmax classifier to obtain the intrusion detection results of the network. The Softmax classifier uses a cross entropy loss function to obtain the probability output of feature values and determine the intrusion type based on the probability value.

6. EXPERIMENT AND ANALYSIS

6.1 Experimental Environment

The experimental environment is the Win10 operating system, with a 4-core 8-thread Intel (R) Core (TM) i7-10510U, 12 GB memory, and programming language using Python 3.6.

This experiment uses TensorFlow gpu-1.15.0 to build the model. To ensure the training efficiency of the model, the time step is set to 4, and the model is trained a total of 150 times. At the same time, in order to improve the learning performance of the model, the learning rate is initially set to

0.001, and Adam is used as the optimizer, where Adam's parameter beta_ 1 and beta_ Set 2 to 0.9 and 0.999 respectively.

6.2 Experimental Dataset

Two datasets were selected for evaluation in the experiment, namely the NSL-KDD and the UNSW-NB15 dataset. The characteristics and attack identification of NSL-KDD and KDD99 are the same, but NSL-KDD cleared and organized some duplicate records in KDD99, containing a total of over 100000 pieces of data. The data distribution is shown in Table 3.

UNSW-NB15 is a dataset collected by the Canberra Network Range Laboratory in 2015 in a real network environment, with traffic data that is more in line with current real network activities and contemporary attack behavior. This dataset contains 9 attacks, 49 features, and 1 labeled feature. Table 4 shows the data distribution after dividing the UNSW-NB15 dataset into training and testing sets in a 3:2 ratio.

Table 3. Sample distribution of the NSL-KDD dataset

Attack Type	NSL-KDD	
	Training set	Test set
Normal	67343	9711
DoS	45927	7544
Probe	11656	2421
R2L	995	2754
U2R	52	200
Total	125973	22544

Table 4. Sample distribution of the UNSW-NB15 dataset

Attack Type	NSL-KDD	
	Training set	Test set
Normal	56000	37000
Generic	40000	18871
Exploits	33393	11132
Fuzzers	18184	6062
DoS	12264	4089
Reconnaissance	10491	3496
Analysis	2000	677
Backdoor	1746	583
Schellcode	1133	378
Worms	130	44
Total	175341	82332

6.3 Evaluation Index

The evaluation indexes used in the experiment include accuracy (Acc), precision (Pre), recall (Rec), and F1 score, calculated as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$\Pr e = \frac{TP}{TP + FP} \tag{10}$$

$$\mathrm{Re}\, c = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = \frac{2 * \Pr e * \mathrm{Re}\, c}{\Pr e + \mathrm{Re}\, c} \tag{12}$$

where, *TP* is the true case, *TN* is the true negative case, *FP* is the false positive case, and *FN* is the false negative case.

Meanwhile, the detection time represents the time required to train a batch of samples and export accuracy.

6.4 Model Training

6.4.1 Model Parameters Determination

The main parameters that need to be clarified in the system are *h*, d_k and Drop. The parameter values and corresponding experimental results used on the two datasets are shown in Tables 5 and 6.

Table 5. Test results for parameter values and NSL-KDD dataset

Group	Parameter			Acc/%	Time/s
	h	d_k	**Drop**		
1	1	32	0.1	99.23	10.17
	2	16	0.1	99.28	11.96
	4	8	0.1	99.31	12.54
	8	4	0.1	99.29	16.61
2	4	8	0.0	99.27	12.63
	4	8	0.2	99.35	12.60
	4	8	0.3	99.30	12.55

Table 6. Test results for parameter values and UNSW-NB15 dataset

Group	Parameter			Acc/%	Time/s
	h	d_k	**Drop**		
1	1	32	0.1	89.67	12.81
	2	16	0.1	89.72	13.49
	4	8	0.1	89.75	15.24
	8	4	0.1	89.71	19.03
2	4	8	0.0	89.74	15.52
	4	8	0.2	89.70	15.47
	4	8	0.3	89.69	15.31

From Table 5, it can be seen that according to the first group of experiments, as the number of heads h increases, the time spent on training a batch of samples also increases. When h=4, the model has strong generalization ability. According to the second set of experiments, the proposed method is more suitable when the internal Dropout of the encoder is set to 0.2, with an accuracy of 99.35%, which improves the generalization ability. However, if it is too large, the accuracy will decrease. Therefore, in the experiment of the NSL-KDD dataset, the parameters were set to h=4, d_k=8, and Drop=0.2.

Similarly, as shown in Table 6, when h, d_k, and Drop are 4, 8, and 0.1, respectively, the proposed method has the best detection performance, reaching 89.75%. Therefore, the experimental parameters for the UNSW-NB15 dataset are set to h=4, d_k=8, and Drop=0.1.

6.4.2 Analysis of Model Inspection Results

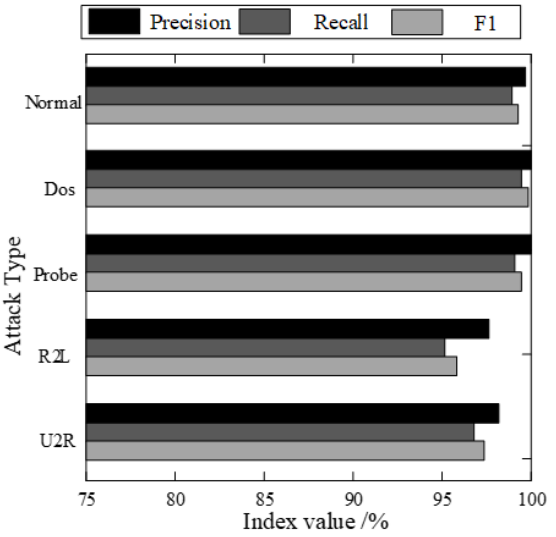
In order to simulate multi user participation in federated learning, assuming 12 users participate in training in the experiment, the detection index values of different attack types in the two datasets of the proposed method are shown in Fig. 6.

From Fig.6, the proposed method has a detection accuracy of nearly 100% for normal samples in the two datasets. However, for samples with small sample sizes, such as U2R and R2L in the NSL-KDD, and Worms and Analysis in the UNSW-NB15, the detection results are poor, with a detection recall of less than 95% for R2L and a detection accuracy of less than 90% for Worms. Although the proposed method uses an improved GAN algorithm to enhance network samples and can improve the impact of imbalanced sample distribution, it cannot be completely eliminated. Overall, the proposed method is effective for detecting two datasets.

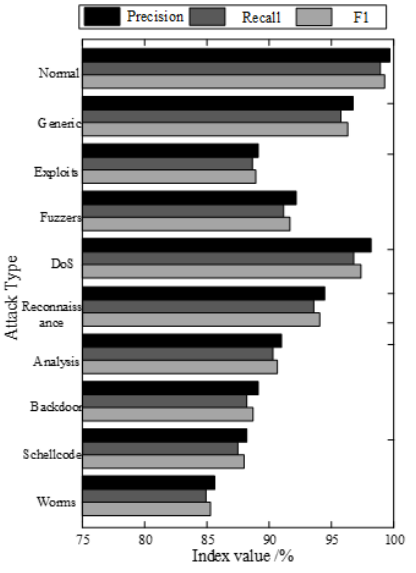
6.5 Comparison of Experimental Results

In the experiment, OSVM (Xu, et al., 2021), BiGRU-SL (Ma, & Ding, 2022), Res-Tran BiLSTM (Mohy, et al., 2023) are selected as comparative methods to demonstrate the detection performance of the proposed method. Among them, Xu, et al., (2021) utilize machine learning OSVM model to achieve network intrusion detection. Ma, & Ding, (2022) and Mohy, et al., (2023) both use deep learning model for intrusion detection. Ma, & Ding, (2022) use single class BiGRU autoencoder and EL to classify network attack types, and Mohy, et al., (2023) propose Res-Trans BiLSTM model for intrusion detection.

Figure 6. Classification performance on different datasets: (a) NSL-KDD, (b) UNSW-NB15



(a) NSL-KDD



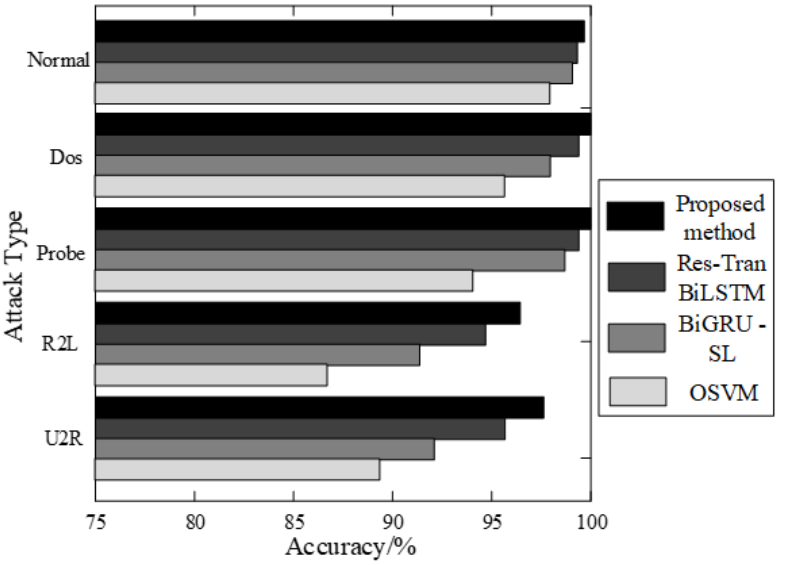
(b) UNSW-NB15

6.5.1 Comparison Results of NSL-KDD

The detection accuracy of the four methods on the NSL-KDD dataset for various attack types is shown in Fig. 7.

From Fig. 7, the proposed method has good detection results for various attack types in the NSL-KDD, with Acc rates exceeding 95%. This is because the proposed method utilizes an improved GAN for data augmentation and a Transformer model for federated learning, further ensuring detection reliability. The Res-Tran BiLSTM model has similar performance to the proposed method, but lacks

Figure 7. Detection results of the NSL-KDD



support from a federated learning environment, resulting in a significant decrease in detection accuracy for small samples. However, the BiGRU-SL and OSVM models lack data imbalance processing, resulting in poor detection performance for R2L and U2R, especially for the OSVM model, which has a detection accuracy of less than 90%.

After multiple experiments, the detection results of the four methods on the NSL-KDD are shown in Table 7.

From Table 7, the detection results of the proposed method are superior to other methods, with Acc, Rec, and F1 values of 99.45%, 98.31%, 95.72%, and 96.99%, respectively. The Res-Tran BiLSTM model implements intrusion detection, but lacks distributed learning for massive data, resulting in a 3.08% decrease in detection accuracy compared to the proposed method. The BiGRU-SL model combines BiGRU and EL for intrusion detection, without considering the impact of data imbalance, resulting in poor detection performance, with a Rec rate of only 91.94%. The OSVM model implements network intrusion detection. A single model is difficult to detect complex attack types, and the F1 value of the detection results is less than 90%.

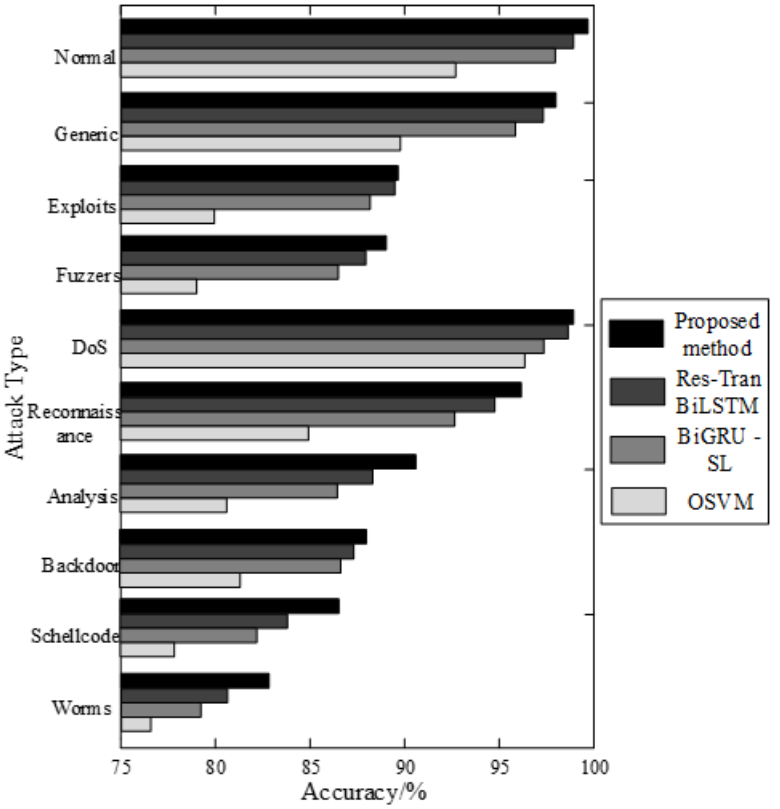
6.5.2 Comparison Results of UNSW-NB15

The detection accuracy of the four methods on the UNSW-NB15 for various attack types is shown in Fig. 8.

Table 7. Comparison of results between different methods on the NSL-KDD (%)

Method	OSVM	BiGRU -SL	Res-Tran BiLSTM	Proposed Method
Acc/%	91.89	95.21	96.37	99.45
Pre/%	90.92	94.56	95.68	98.31
Rec/%	87.44	90.05	91.94	95.72
F1/%	89.15	92.25	93.77	96.99

Figure 8. Detection results of the UNSW-NB15 dataset



From Fig. 8, the proposed method has higher detection accuracy for various attack types in the UNSW-NB15 dataset than other comparative methods, especially for attack types such as Worms and Schellcode with fewer samples, and its detection advantages are more obvious. Due to the fact that the UNSW-NB15 records real network environments, with more types of attacks and more complex features, the detection efficiency of all four methods has decreased. After multiple experiments, the detection results of the four methods on the UNSW-NB15 are shown in Table 8.

As shown in Table 8, compared to the NSL-KDD, the proposed method shows a significant decrease in detection performance on the UNSW-NB15, but still outperforms other comparison methods. Its accuracy, Acc, Rec, and F1 values are 89.83%, 89.45%, 87.29%, and 88.36%, respectively. Due to the lack of data imbalance and distributed analysis considerations, the OSVM model significantly reduces its detection performance in the face of massive network

Table 8. Comparison of results between different methods on the UNSW-NB15 (%)

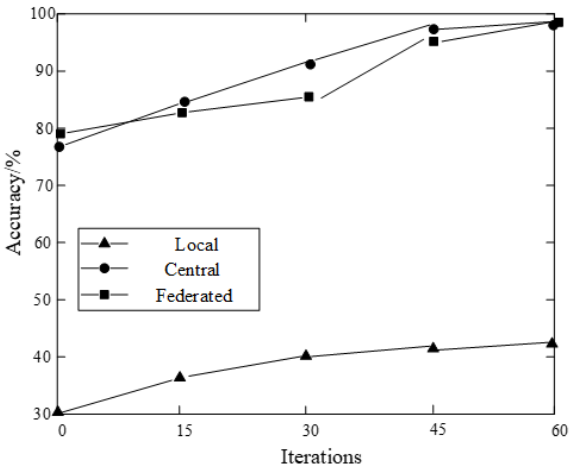
Method	OSVM	BiGRU -SL	Res-Tran BiLSTM	Proposed Method
Acc/%	79.27	83.92	86.65	89.83
Pre/%	78.81	83.05	85.93	89.45
Rec/%	75.53	79.74	82.21	87.29
F1/%	77.14	81.36	84.03	88.36

data, with a Rec rate of only 75.53%. The BiGRU-SL model utilizes double-layer learning to ensure the performance of intrusion detection, but it is not ideal for learning with few sample types, resulting in an overall Acc decrease of 6.40% compared to the proposed method. The Res-Tran BiLSTM model uses a hybrid model for intrusion detection, which can improve detection reliability. However, the centralized analysis mode leads to a decrease in computational reliability, resulting in an F1 value below 85%. The proposed method not only solves the small sample learning problem but also improves the reliability and efficiency of detection by introducing improved GAN networks and federated learning.

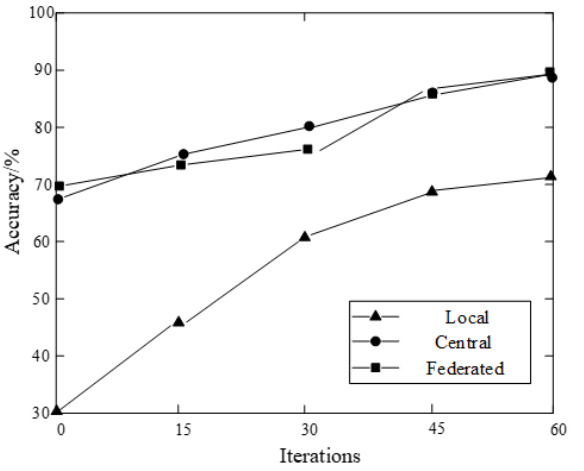
6.5.3 Comparison of FL Training

The detection accuracy of the proposed model is shown in Fig.9 after testing local learning, centralized learning, and FL environments on the two dataset, respectively.

Figure 9. Detection accuracy in different learning environments: (a) NSL-KDD, (b) UNSW-NB15



(a) NSL-KDD



(b) UNSW-NB15

From Fig. 9, when the local labels are insufficient, the detection accuracy on the KDD99 is less than 45%. Similarly, the accuracy on the UNSW-NB15 is only about 70%, which is not sufficient as a high-quality detection model. According to the results of centralized training and federated training, it can be found that the accuracy on the two datasets is almost the same, with only slight differences, and can even reach over 99% on the NSL-KDD. Compared to centralized learning, FL first saves the dataset locally to the participants, and the server cannot access any data from any participant, effectively protecting the privacy of participant data. Secondly, participants only need to train locally and submit the locally trained model to the central server, which greatly reduces communication traffic during the training process compared to submitting data to the central server.

7. CONCLUSION

DL has been widely applied, but the problem of a single user having data that cannot meet training needs in practical applications is ignored. Therefore, a NID method for information systems based on FL is proposed, which not only protects user privacy and security, but also has the ability to detect anomalies in network traffic data. In a NID system built on the FL framework, the Transformer model is used as its universal detection model. The FL characteristics are utilized to enable multiple participants to participate in training using a universal model. After training, the parameters are uploaded to the cloud server, and the updated universal model is continuously iterated. The final classification prediction results are obtained through the Softmax classifier. The experimental results based on the two datasets indicate that:

(1) By improving the GAN model, the problem of data imbalance has been improved, data quality has been improved, and the reliability of intrusion detection has been further enhanced. Its accuracy on the NSL-KDD and UNSW-NB15 was 99.45% and 89.83%, respectively.
(2) The proposed method utilizes FL for multiple nodes to participate in model training, and chooses the Transformer model with strong learning ability as its universal detection model, ensuring local data security while improving the accuracy of ID.

The paper focuses on the accuracy and efficiency of network intrusion detection, while there is less attention to network security. Therefore, in the following research, efficient and secure aggregation algorithms will be further explored, such as considering using differential privacy instead of homomorphic encryption to protect model parameters to reduce computational complexity, or using blockchain technology to ensure detection security. In addition, for the small sample detection problem, in-depth research will be conducted in the future to comprehensively improve the detection accuracy of the proposed method. Meanwhile, future research will consider more intrusion types to enhance the practical application value of the proposed method.

ACKNOWLEDGMENT

This work was supported by the Guangdong Provincial Major Research Platform Ordinary University Characteristic Innovation Project Fund (No.2022KTSCX204).

COMPETING INTERESTS STATEMENT

The author declares that there is no competing interest regarding the publication of this paper.

REFERENCES

Aryandoust, A., Patt, A., & Pfenninger, S. (2020). Enhanced spatio-temporal electric load forecasts using less data with active deep learning. *Nature Machine Intelligence*, 977–991.

Chen, R., Tang, X., & Li, X. (2022). Adaptive stochastic gradient descent method for convex and non-convex optimization. *Fractal and Fractional*, *6*(12), 709–709. doi:10.3390/fractalfract6120709

Deore, B., & Bhosale, S. (2023). Adaptive dolphin atom search optimization-based drnn for network intrusion detection system. *SN Computer Science*, *4*(5), 1–1. doi:10.1007/s42979-023-02006-6

Devi, S., & Bharti, T. S. (2022). A review on detection and mitigation analysis of distributed denial of service attacks and their effects on the cloud. *International Journal of Cloud Applications and Computing*, *12*(1), 1–21. doi:10.4018/IJCAC.311036

Durga Prasa, J. K., & Vasumathi, D. (2019). Privacy preserving data analysis using decision tree learning algorithm through additive homomorphic encryption. *International Journal of Innovative Technology and Exploring Engineering*, *8*(4), 267–272.

Gaurav, A., Gupta, B. B., & Panigrahi, P. K. (2023). A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system. *Enterprise Information Systems*, *17*(3), 2023764. doi:10.1080/17517575.2021.2023764

Harini, R., Maheswari, N., Ganapathy, S., & Sivagami, M. (2023). An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach. *Alexandria Engineering Journal*, *2023*(78), 469–482. doi:10.1016/j.aej.2023.07.063

He, W., & Zhao, L. (2022). Application of federated learning algorithm based on k-means in electric power data. *Journal of New Media*, *4*(4), 191–203. doi:10.32604/jnm.2022.032994

Idrissi, M. J., & Alami, H. (2023). Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems. *Expert Systems With Applications, 234*(30), 121000.

Layeghy, S., Baktash, M., & Portmann, M. (2023). Domain invariant network intrusion detection system. *Knowledge-Based Systems*, *273*, 110626. doi:10.1016/j.knosys.2023.110626

Ling, Z., & Hao, Z. J. (2022). An intrusion detection system based on normalized mutual information antibodies feature selection and adaptive quantum artificial immune system. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–25. doi:10.4018/IJSWIS.308469

Ling, Z., & Hao, Z. J. (2022). Intrusion detection using normalized mutual information feature selection and parallel quantum genetic algorithm. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–24. doi:10.4018/IJSWIS.307324

Lu, J., Shen, J., Vijayakumar, P., & Brij, B. (2021). Blockchain-based secure data storage protocol for sensors in the industrial internet of things. *IEEE Transactions on Industrial Informatics*, *18*(8), 5422–5431. doi:10.1109/TII.2021.3112601

Ma, G., & Ding, H. (2022). Semi-Supervised random forest methodology for fault diagnosis in air-handling units. *Buildings*, *13*(1), 14–14. doi:10.3390/buildings13010014

Maidamwar, P. R., Lokulwar, P. P., & Kumar, K. S. (2023). Ensemble learning approach for classification of network intrusion detection in IoT environment. *International Journal of Computer Network and Information Security*, *15*(3), 30–36. doi:10.5815/ijcnis.2023.03.03

Mishra, A., Joshi, B. K., Arya, V., Gupta, A. K., & Chui, K. T. (2022). Detection of distributed denial of service (ddos) attacks using computational intelligence and majority vote-based ensemble approach. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–10. doi:10.4018/IJSSCI.309707

Mohy, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. *Multimedia Tools and Applications, 82*(15), 23615-23633.

Mustafa, A. L. (2023). Machine learning for network intrusion detection—A comparative study. *Future Internet*, *15*(7), 243. doi:10.3390/fi15070243

Oliveira, H. S., & Oliveira, H. P. (2023). Transformers for energy forecast. *Sensors (Basel)*, *23*(15), 6840–6853. doi:10.3390/s23156840 PMID:37571622

Priya, S., & Ponmagal, R. S. (2023). Network intrusion detection system based security system for cloud services using novel recurrent neural network-autoencoder (NRNN-AE) and genetic. *Advances in Science and Technology (Owerri, Nigeria)*, *6630*, 729–737. doi:10.4028/p-076960

Rahman, S. A., Tout, H., Talhi, C., & Mourad, A. (2020). Internet of Things intrusion detection: Centralized, on-device, or federated learning? *IEEE Network*, *34*(6), 310–317. doi:10.1109/MNET.011.2000286

Sharma, R., & Sharma, N. (2022). Attacks on resource-constrained IoT devices and security solutions. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–21. doi:10.4018/IJSSCI.310943

Singh, A., & Gupta, B. B. (2022). Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: Issues, challenges, and future research directions. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–43. doi:10.4018/IJSWIS.297143

Song, Y., Luktarhan, N., Shi, Z., & Wu, H. (2023). TGA: A novel network intrusion detection method based on TCN, BiGRU and attention mechanism. *Electronics (Basel)*, *12*(13), 2849. doi:10.3390/electronics12132849

Stergiou, C. L., Psannis, K. E., & Gupta, B. B. (2021). InFeMo: Flexible big data management through a federated cloud system. *ACM Transactions on Internet Technology*, *22*(2), 1–22. doi:10.1145/3426972

Tembhurne, J. V., Almin, M. M., & Diwan, T. (2022). Mc-DNN: Fake news detection using multi-channel deep neural networks. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–20. doi:10.4018/IJSWIS.295553

Usoh, M., Asuquo, P., Ozuomba, S., Stephen, B., & Inyang, U. (2023). A hybrid machine learning model for detecting cybersecurity threats in IoT applications. *International Journal of Information Technology : an Official Journal of Bharati Vidyapeeth's Institute of Computer Applications and Management*, *15*(6), 3359–3370. doi:10.1007/s41870-023-01367-8

Vitorino, J., Praça, I., & Maia, E. (2023). SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection. *Computers & Security*, *134*, 1–10. doi:10.1016/j.cose.2023.103433

Wang, S., Xu, W., & Liu, Y. (2023). Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things. *Computer Networks*, *2023*(235), 1–16. doi:10.1016/j.comnet.2023.109982

Wang, T., Pan, Z., Hu, G., Duan, Y., & Pan, Y. (2022). Understanding universal adversarial attack and defense on graph. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–21. doi:10.4018/IJSWIS.308812

Wang, X., Zhang, H., Bilal, A., Long, H., & Liu, X. (2023). WGM-dSAGA: Federated learning strategies with byzantine robustness based on weighted geometric median. *Electronics (Basel)*, *12*(5), 1190–1190. doi:10.3390/electronics12051190

Wang, Y., Xu, L., Liu, W., Li, R., & Gu, J. (2023). Network intrusion detection based on explainable artificial intelligence. *Wireless Personal Communications*, *131*(2), 1115–1130. doi:10.1007/s11277-023-10472-7

Xu, Z., He, D., Vijayakumar, P., Gupta, B. B., & Shen, J. (2021). Certificateless public auditing scheme with data privacy and dynamics in group user model of cloud-assisted medical WSNs. *IEEE Journal of Biomedical and Health Informatics*, *27*(5), 2334–2344. doi:10.1109/JBHI.2021.3128775 PMID:34788225

Yadav, K., Gupta, B. B., Hsu, C. H., & Chui, K. T. (2021). *Unsupervised federated learning based IoT intrusion detection*. 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Kyoto, Japan.

Yan, F., Zhang, G., Zhang, D., Sun, X., Hou, B., & Yu, N. (2023). TL-CNN-IDS: Transfer learning-based intrusion detection system using convolutional neural network. *The Journal of Supercomputing*, *79*(15), 17562–17584. doi:10.1007/s11227-023-05347-4

Yang, L., Tian, Y., Song, Y., Yang, N., Ma, K., & Xie, L. (2020). A novel feature separation model exchange-GAN for facial expression recognition. *Knowledge-Based Systems, 204*, 1-13.

Yao, W., Hu, L., Hou, Y., & Li, X. (2023). A lightweight intelligent network intrusion detection system using one-class autoencoder and ensemble learning for IoT. *Sensors (Basel)*, *23*(8), 4141–4152. doi:10.3390/s23084141 PMID:37112482

Zhang, L., Jiang, S., Shen, X., Brij, B., Gupta, & Tian, Z. (2021). PWG-IDS: An intrusion detection model for solving class imbalance in IIoT networks using generative adversarial networks. *ArXiv.org, 6*, 1-1.

Zhang, L., & Zhang, J. H. (2022). Intrusion detection using normalized mutual information feature selection and parallel quantum genetic algorithm. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–24. doi:10.4018/IJSWIS.307908

Zhang, Q., Guo, Z., Zhu, Y., Vijayakumar, P., Castiglione, A., & Gupta, B. B. (2023). A deep learning-based fast fake news detection model for cyber-physical social services. *Pattern Recognition Letters*, *168*, 31–38. doi:10.1016/j.patrec.2023.02.026

Zhang, Y. J., & Wang, Z. H. (2023). Feature engineering and model optimization based classification method for network intrusion detection. *Applied Sciences (Basel, Switzerland)*, *13*(16), 1–1. doi:10.3390/app13169363

Zhang, Z., Feng, F., & Huang, T. (2022). FNNS: An effective feedforward neural network scheme with random weights for processing large-scale datasets. *Applied Sciences (Basel, Switzerland)*, *12*(23), 12478–12478. doi:10.3390/app122312478

Qi Zhou, Ph.D. in Computer Information Technology, Associate Professor. He graduated from Sun Yat-sen University in 2008. He currently works at Guangdong Open University. His research interests include network security and information technology security.

Zhoupu Wang, Master's degree in communication and information systems. He graduated from Sichuan University in 2019. He currently works at China Telecom Sichuan Branch. His research area is mainly in communication and information systems.