IoT Malicious Traffic Detection Based on FSKDE and Federated DIOT-Pysyft

Ke Zhang[1], Guanghua Zhang[1(✉)], Zhenguo Chen[2], and Xiaojun Zuo[3]

[1] School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China
zhanggh@hebust.edu.cn

[2] Hebei IoT Monitoring Engineering Technology Research Center, North China Institute of Science and Technology, Langfang 065201, China

[3] State Grid Hebei Electric Power Research Institute, Shijiazhuang 050021, China

Abstract. In order to solve the limitations of existing malicious traffic detection methods in the Internet of Things (IoT) environment, such as resources, heterogeneous devices, scarce traffic, and dynamic threats, this paper proposes the Feature Selection based on Kernel Density Estimation (FSKDE) and the federated learning method Detection Internet of Things based on Pysyft (DIOT-Pysyft). First, IoT devices perform data preprocessing operations on the collected network traffic data; Second, the FSKDE is used to calculate the probability density of each column of features and selects features according to a preset abnormal threshold; Third, the DIOT-Pysyft is build. It initializes the server that the federated convolutional neural network (CNN) is sent to the IoT devices. The IoT devices use the processed data to train the federated CNN and send them to server secretly. After that, the improved FedAvg algorithm is used to average the gradient of the federated CNN model, which for training and transmitting the encrypted and averaged gradient to the server to build a new global model to participate in the next round of training. Finally, this paper uses the UNSW-NB15 dataset to verify the proposed method for detecting malicious traffic in the IoT environment. The experimental results show that the identification accuracy of the IoT malicious traffic detection based on FSKDE and federated DIOT-Pysyft reaches 91.78%, which can detect potential malicious traffic in the IoT environment. The improved FedAvg method further protects the privacy and security of IoT data and ensures the accuracy while protecting the data.

Keywords: IoT · FSKDE · Federated Learning · DIOT-Pysyft · Malicious Traffic Detection

1 Introduction

With the continuous emergence of application scenarios such as smart homes, smart medical care, smart transportation, and smart cities, Internet of Things (IoT) devices and their technologies have brought convenience to human life. However, a large number of IoT devices and cloud servers are directly exposed to the Internet at present. If

S. Goel et al. (Eds.): ICDF2C 2022, LNICST 508, pp. 203–221, 2023.
https://doi.org/10.1007/978-3-031-36574-4_12

it is exploited, it will lead to security risks such as equipment loss of control, user privacy leakage, cloud server data theft, etc., and even have a serious impact on the basic communication network. The security challenges of the IoT have not been considered in-depth, resulting that the IoT environment has always been the hardest hit area for various vulnerability attacks [1]. Various security threats in the IoT emerge in an endless stream. The botnet malware controlled 100,000 IoT devices to launch Distributed Denial of Service (DDoS) attacks on the domain name servers managed by Dyn, which directly caused the complete paralysis of most of the Internet in the United States [2]. The existence of 250 vulnerabilities in IoT devices, including open ports, outdated firmware, and unencrypted transmission of sensitive data [3, 4] results in the lack of key security functions of IoT. Once the IoT devices with vulnerabilities enter the network, which can be easily exploited by attackers to launch Denial of Service (DoS) attacks. Therefore, malicious traffic detection for attack behaviors generated by IoT devices can mitigate network security risks.

Most of the detection methods based on malicious traffic are designed for servers or personal computers with sufficient resources. There are relatively few abnormal traffic detection methods and technologies for the characteristics of the IoT. Compared with the traditional Internet, the IoT has the following characteristics [5]: 1) Resources are limited. The functions of IoT devices are limited, so the available memory and computing resources are insufficient. 2) The devices are heterogeneous. There are many IoT devices, the behaviors of different IoT devices are very diverse, the traffic patterns in different time periods are quite different, so the attacks on different devices are also different. 3) Communication traffic is scarce. Compared to the Internet, some IoT devices generate less traffic which often triggered by infrequent user interactions. 4) Dynamic threats increase. New IoT devices are released every day with security vulnerabilities, and attackers develop attacks against these devices at the same high rate. Threats targeting IoT devices are increasing dynamically. The above characteristics of IoT devices make traditional Internet-based intrusion detection methods not directly applicable to the IoT, which makes IoT terminal devices more vulnerable to malware attacks.

Considering the above problems, this paper proposes the Feature Selection based on Kernel Density Estimation (FSKDE) and federated learning method Detection Internet of Things based on Pysyft (DIOT-Pysyft) to detect IoT malicious traffic. The main contributions of this method are:

- A feature selection method based on FSKDE is proposed. This method considers the problem of limited memory and computing resources when the model is calculated in IoT devices, so a feature selection method based on FSKDE is proposed to select the optimal subset of features. The algorithm uses the Gaussian kernel density estimation method without the need for the distribution of the data samples to be assumed in advance. The features that contribute more to the classification are initially screened.
- A traffic detection method based on federated DIOT-Pysyft is proposed, which utilizes the characteristics of federated learning and secure multi-party computation, uses the dataset of IoT devices to train the local federated convolutional neural network (CNN) model, and encrypts the gradient of the model during training with the improved FedAvg algorithm after the aggregation is averaged, it is uploaded to the server to

update the model, and it participates in the next round of training. This method realizes the detection of malicious traffic and protects the privacy of user data.
– Combined with the feature selection method based on FSKDE and the traffic detection method based on federated DIOT-Pysyft, the UNSW-NB15 dataset is multi-classified for malicious traffic, the federated CNN method is used for model training, and the performance and time of different algorithms is compared. The method proposed in this paper are evaluated.

2 Related Work

Most of the detection methods currently don't take into account the characteristics of the IoT. Literature [6] used different Back Propagation (BP) neural networks on the gateway to establish different anomaly detection models for different types of IoT devices, but this strategy increased the calculation and storage of the gateway. It was not suitable for the application environment of the home IoT; Literature [7] proposed a lightweight detection method for DDoS malicious traffic, which effectively solved the problem of limited resources of IoT devices, but did not consider the heterogeneity between devices; In addition, most of the current experiments use public network traffic data sets, and some of the data are collected in the Internet environment, so the amount of traffic is huge. But the problem of the small amount of actual IoT traffic is not considered. Signature-based detection methods match specific patterns or strings in known attacks or threats to detect malicious traffic. By building a knowledge base to identify known threats in the network environment, unknown attacks and potential threats cannot be detected.

In recent years, Machine Learning (ML) and Deep Learning (DL) technologies have been widely used in IoT scenarios to mitigate malicious attacks, and related research has also proposed various improved algorithms to improve the detection ability of Intrusion Detection Systems (IDS). The experiments show that ML and DL have great advantages in enhancing the accuracy of IDS. Reference [8] proposed a mirai botnet attack detection method based on a bidirectional long memory Recurrent Neural Network (RNN). Simulation experiments showed that it had achieved good experimental results, but the model was too complex and was not suitable for the IoT network environment with insufficient computing and storage resources. Reference [7] deployed the abnormal traffic detection model on the IoT gateway, which can prevent network attacks to the greatest extent. Reference [9] proposed an IoT anomaly detection algorithm, which was effective and sensitive against DoS. It can quickly detect high-traffic situations, and it can also handle abnormal problems caused by system failures and human operations, but this method relied on serial communication protocols and the network attacks detected were relatively simple. Reference [10] proposed a security scheme by analyzing the architecture of the IoT layer and proposed a security measure against the vulnerabilities existing in other network layers.

Although ML and DL can improve the detection performance of IDS, the training of the model needs to upload the data from the client to the server. The data is managed centrally. It is difficult to apply to distributed network traffic data from different devices through a single entity. In the IoT environment, this entity can access the communication data and network traffic of different devices participating in the training process, which may have the problem of privacy leakage [11]. Therefore, this paper adopts a more

secure data management scheme, federated learning, to alleviate the problems caused by malicious traffic in the IoT environment.

Federated learning (FL) was proposed in 2016 [12]. In federated learning, the client does not share its data, but jointly trains the model under the coordination of the central server. The client only needs to transmit encrypted gradient-related data, and use multi-source data to collaborate and train a unified model [13], which can keep the training data decentralized. In recent years, the development of federated learning-based intrusion detection systems in IoT scenarios has attracted more and more researchers' interest [14–16]. Reference [17] proposed a federated learning-based framework that covered Multilayer Perceptron (MLP) and Auto-Encoders (AE) to detect network attacks affecting IoT devices in a way that protected data privacy, but it can only perform binary classification, dividing traffic into normal traffic and malicious traffic, while there was no further distinction between the types of malicious traffic. Reference [18] proposed a cloud intrusion detection scheme based on blockchain federated learning. The proposed method was based on unrealistic data distribution, inappropriate datasets, and unsuitable settings among the various parties. Reference [19] discussed the challenges and future development directions of federated learning-based intrusion detection systems but did not provide experiments and evaluation results to support their contributions.

This paper proposes an IoT malicious traffic detection method based on FSKDE and federated DIOT-Pysyft. First, a feature selection method based on FSKDE is proposed to reduce redundant features on IoT devices, find optimal feature subsets, and reduce the impact of IoT resource consumption and irrelevant features on malicious traffic detection performance during model training. The traffic detection based on federated DIOT-Pysyft enables IoT devices to use local datasets to train models separately, and use the improved FedAvg algorithm to encrypt and aggregate the federated gradients generated during training. Then, the gradient is uploaded to the server to update the model which can be added to the next round of training. When the model converges, the training end. The method alleviates the problem of inaccurate identification caused by insufficient traffic of some IoT devices and realizes multi-classification by combining the knowledge learned from each IoT device. Finally, combining the FSKDE-based feature selection method and the federated DIOT-Pysyft-based traffic detection method, the experiments are carried out on the UNSW-NB15 dataset. The experimental results show that the proposed method not only guarantees accuracy but also pays attention to protecting the privacy of user data. In addition, it can alleviate the problems existing in traditional methods such as device heterogeneity, and less communication traffic, and can effectively detect unknown attacks.

3 The IoT Malicious Traffic Detection Based on FSKDE and Federated DIOT-Pysyft

FSKDE is a feature selection algorithm based on gaussian kernel density estimation, and DIOT-Pysyft is a federated learning-based IoT malicious traffic detection method, as shown in Fig. 1. First, IoT devices perform data preprocessing operations on their respective network traffic data. Then, FSKDE is used to calculate the probability density of each column of features, and feature selection is performed according to a preset anomaly threshold to reduce computing and communication resources in federated

learning training. Second, build the central server as a model aggregator, the IoT devices receive the federated CNN model transferred by the server, train the model by using the federated CNN and local data, encrypt the parameter weights of the trained model, and transmit them to the server. The aggregator uses the FedAvg aggregation algorithm to combine all parameter weights to build a new global model to participate in the next round of training. Finally, this paper uses the UNSW-NB15 dataset to perform the IoT malicious traffic detection based on FSKDE and federated DIOT-Pysyft. The method is verified to realize the identification of various malicious traffic in the IoT environment and protect the privacy of users.

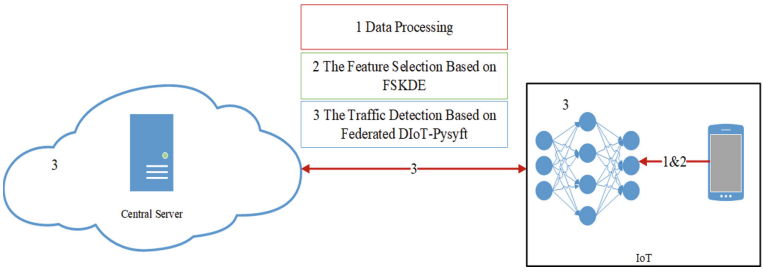

Fig. 1. The framework of IoT malicious traffic detection based on FSKDE and federated DIOT-Pysyft

The system architecture used in this solution is divided into two subsystems: the central server-side and the IoT client side.

Each client contains multiple IoT devices. For simplicity, only one smart device is used for the architecture and experiments. The client is responsible for collecting traffic data from IoT devices, which uses port mirroring through the switch connecting the IoT devices, as described in [20]. Firstly, because the collected data is unstructured, it cannot be used directly, and the data needs to be structured; secondly, in the data preprocessing part, missing values and data normalization need to be processed. Then, considering the practical problem of insufficient malicious traffic resources in the IoT environment, a feature selection algorithm based on gaussian kernel density estimation (Feature Selection Based on Kernel Density Estimation, FSKDE) is proposed. In the network traffic data, the features that contribute prominently to malicious traffic detection are screened out according to the FSKDE, and the optimal subset is selected to reduce the dimension of the sample features.

On the central server-side, the traffic detection method based on federated DIOT-Pysyft transmits the federated CNN model to the client to be trained on the server-side. After the client receives the model, it's trained for one or more rounds using local standardized data. Then, the encrypted model gradients upload to the server after training. The server aggregates the model gradients from each client using improved FedAvg to update the model parameters, which completes an iterative process of the model. Finally, When obtaining the optimal model parameters after several iterations, the server sends the optimized model to the client for final malicious traffic detection. The local training method of this method can speed up the detection time. In addition,

because the optimized model aggregates the parameters of all training data from multiple IoT devices, knowledge sharing is realized. The devices participating in the training can detect the malicious traffic categories owned by different participants in a private way.

3.1 Data Preprocessing

The quality of the data determines the prediction and generalization ability of the model, as does the actual network traffic data. The UNSW-NB15 dataset used in this paper contains missing values, invalid values, categorical and numerical features. The quantitative units are also different. In order to ensure that the accuracy of the model is not affected, this paper preprocesses the data before training.

(1) **Handling of missing and invalid values:** The UNSW-NB15 dataset contains a small amount of missing data (e.g. np.nan). For a small number of missing values, the strategy of directly deleting the row of data is adopted. For the categorical type of the 'service' column, the data contains 57% of invalid data '-'. The method of mode filling will make the data out of reality and the method of predicting using the correlation between each feature will increase computational cost. Therefore, this paper directly deletes the column data containing '-'.

(2) **Data normalization:** The advantage of data normalization is that it can eliminate the differences between data of different dimensions and make the data of different dimensions comparable after normalization. Each feature in the UNSW-NB15 dataset has different value ranges. To ensure the reliability of the training results, each feature must be normalized to the [0, 1] range. This paper uses the Min-Max Normalization method to normalize the data, as shown in Eq. (1).

$$x_{ij}^{'} = \frac{x_{ij} - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Among them, x_{ij} is the original data, $x_{\min}$ is the minimum value of the eigenvalues of the column, $x_{\max}$ is the maximum value of the eigenvalues of the column, and $x_{ij}^{'}$ is the normalized result.

(3) **Categorical feature processing:** For the 'state' feature in the UNSW-NB15 dataset, its value is limited to ['FIN', 'INT', 'CON', 'REQ', 'ACC', 'CLO', 'RST'], this paper adopts the form of one-hot encoding. For the 'proto' feature, the corresponding value of its variables is too much, this paper adopts the weight of evidence. This method considers the relationship between the independent variable and the target, which is the natural logarithm of the odds ratio in mathematics. By calculating the WOE value of 'proto', the 'proto' column can be replaced by the WOE.

3.2 The Feature Selection Based on FSKDE

Considering the limited computing and communication resources of IoT devices, this paper proposes a feature selection method based on FSKDE. Kernel density estimation (KDE), as a non-parametric estimation method in statistics, is very suitable for estimating the probability of unknown sample sets. Compared with parameter estimation, the advantage of the KDE method is that it does not need to assume the distribution model of the data samples in advance. The test sample set is used for density estimation of unknown probabilities.

Assuming that there are n samples independently distributed in the data sample set $\{X|x_1, x_2, \cdots, x_n\}$, and the probability density function of sample $x_i \in X$ is $f(x_i)$, the kernel density estimation formula is shown in Eq. (2):

$$f_h(x_i) = \frac{1}{n}\sum_{j=1}^{n} K_h\left(x_i - x_j\right) = \frac{1}{nh}\sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h}\right) \tag{2}$$

Among them, $K(\cdot)$ is the kernel function, which usually has symmetry and satisfies $\int K(x_i)dx = 1$; the parameter h is the bandwidth of the kernel function, which is used to balance the deviation and variance of the kernel density estimation. This paper uses the kernel parameter estimation method proposed in the literature [20] to estimate h. In addition, there are many options for the kernel function. In theory, all smooth peak functions can be used as the kernel function for kernel density estimation. This paper uses the gaussian function as the kernel function.

Assuming that the set of normal traffic samples in the training set is $R^{n_1 \times m}$, n_1 is the number of samples, m is the sample dimension. The sample $x_i \in R^{n_1 \times m}$ represents the $i-th$ sample of $R^{n_1 \times m}$, let $x_i = \{x_i^1, x_i^2, \cdots, x_i^d, \cdots, x_i^m\}$, then x_i^d represents the value of the $d-th$ dimension feature x_i, $a^d = \{x_1^d, x_2^d, \cdots, x_{n_1}^d\}$ denotes the all feature of the $d-th$ dimension, then the set of all eigenvalues can be expressed as $A = \{a^1, a^2, \cdots, a^d, \cdots, a^m\}$. When $a^d \in A$, using the gaussian kernel density estimation method to train, the probability density function f^d corresponding to the eigenvalues of the $d-th$ dimension features of all samples in $R^{n_1 \times m}$ can be obtained, and the probability density function corresponding to the eigenvalues of the $d-th$ dimension features in the sample $x_i \in R^{n_1 \times m}$ is $f^d(x_i^d)$. The value $f^d(x_i^d)$ is larger, the more it can show that x_i^d is more in line with the eigenvalue distribution of the $d-th$ dimension feature in normal traffic samples. Based on the above research, this paper proposes the following feature selection method based on FSKDE, and its pseudocode is shown in Algorithm 1.

Algorithm 1. The feature selection based on FSKDE-normal traffic sample set

Input:

$R^{n_1 \times m}$: Normal traffic sample sets

Output:

F : The probability density function of all normal traffic sample features is saved in F

$F(A)$: Save the probability density of all normal sample eigenvalues in $F(A)$

1: $A \leftarrow (R^{n_1 \times m})^T$ // Each row in A represents a feature

2: **For** a^d in A **do**

3: Use Eq. (1) to calculate f^d // Calculate the probability density function corresponding to the d-dimensional eigenvalue

4: $F \leftarrow f^d$

5: **End For**

6: **For** a^d in A **do**

7: **For** x_i^d in a^d **do**

8: $f^d(a^d) \leftarrow f^d(x_i^d)$

9: **End For**

10: $F(A) \leftarrow f^d(a^d)$

11: **End For**

12: **Return** F , $F(A)$

The more features of the sample, the greater the amount of computation required, which will increase the computational overhead of IoT devices. In addition, not all features contribute to malicious traffic detection, so this paper adopts the FSKDE method for feature selection. First, according to the kernel density estimation formula, each feature column is traversed. we calculate the probability density function of the feature F, bring the eigenvalues belonging to the feature in turn, and calculate the probability density $f^d(x_i^d)$ of the eigenvalues of each feature. The probability density set $f^d(a^d)$ of the feature is obtained. Then, this paper sorts the results $F(A)$ obtained by Algorithm 1 in ascending order, and the sorting result is recorded as $sort(f^d(a^d))$. Pick the value t^d at a percentage of τ locations from $sort(f^d(a^d))$ as the filter threshold for the $d-th$ dimension feature. τ is an empirical parameter, which is usually taken $\tau = 0.1\%$. When the eigenvalue probability density of the $d-th$ dimensional feature of the sample x_j is $f^d(x_j^d) < t^d$, it means that the eigenvalues of the $d-th$ dimensional feature don't match the eigenvalue distribution of normal traffic samples, and the more likely it is malicious traffic data. Therefore, the screening threshold of all features is set as T, and the set is expressed as $T = \{t^1, t^2, \cdots, t^d, \cdots, t^m\}$.

Finally, the eigenvalues of each dimension feature of all samples in the malicious sample set $R^{'n_2 \times m}$ in the training set are brought into F for probability density calculation to obtain $f^d(x_i^d)$, the number of $f^d(x_i^d) < t^d$ in the $d-th$ dimension is counted as z^d. z^d represents the number of abnormal data detected by the feature value of the $d-th$ dimension feature. The number of abnormal data of m dimensional eigenvalues is recorded as $Z = \{z^1, z^2, \cdots, z^d, \cdots z^m\}$. Assuming that the number of malicious samples in the training set is n_2, when $\frac{z^d}{n_2} \geq \alpha$, it means that the $d-th$ dimension feature

$feature_d$ is added to the selected feature subset. All features are traversed, where α is the set threshold, usually 0.8. Its pseudocode is shown in Algorithm 2.

Algorithm 2. The feature selection based on FSKDE-malicious traffic sample set

Input:

F : The probability density function of all normal traffic sample features is saved in F

$F(A)$: Save the probability density of all normal sample eigenvalues in $F(A)$

$R^{'n_2 \times m}$: Normal traffic sample sets

Output:

S : Feature subsets using FSKDE

1: **For** $f^d(a^d)$ in $F(A)$ **do** // Calculate the filter threshold for each dimension of data

2: $t^d \leftarrow \tau \times sort(f^d(a^d))$

3: **End For**

4: $A^{'} \leftarrow (R^{'n_2 \times m})^T$

5: $Z = [0, 0, \cdots, 0]_{1 \times m}$

6: **For** $a^{'d}$ in $A^{'}$ **do**

7: **For** x_i^d in $a^{'d}$ **do**

8: **If** $f^d(x_i^d) < t^d$ **then**

9: $z^d = z^d + 1$

10: **End If**

11: **End For**

12: **End For**

13: **For** z^d in Z **do** //select features

14: **If** $\frac{z^d}{n_2} \geq \alpha$ **then**

15: $S \leftarrow feature_d$

16: **End If**

17: **End For**

18: **Return** S

3.3 The Traffic Detection Based on Federated DIOT-Pysyft

With the rapid growth of IoT-connected devices, traditional malicious traffic detection methods are prone to introduce a single point of failure and compromise data privacy, which has a negative impact on the development of new industries such as smart homes and smart healthcare. Considering the resource, communication, and privacy security issues of IoT devices, this paper proposes an IoT traffic detection method based on federated DIOT-Pysyft. The method performs local training on the detection model to

maintain data privacy. At the same time, the trained gradient encryption is uploaded to the server to update the model, and the improved detection model is shared with the participating devices, which can benefit from the knowledge of the participants. The architecture of this solution is shown in Fig. 2 below, which describes the architecture of the client and the interaction with the server after data collection. IoT data preprocessing, FSKDE-based feature selection, and model training and evaluation are all done on the client-side, and the server participates in model gradient aggregation and model update, acting as a model aggregator.

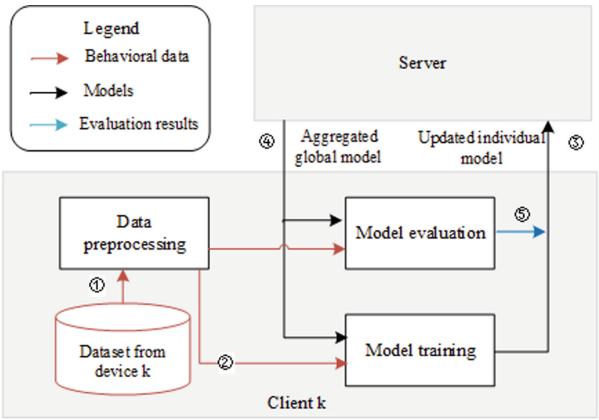

Fig. 2. Detailed view of client architecture in training evaluation

The DIOT-Pysyft Model Training

This solution consists of a federated DIOT-Pysyft learning system composed of a server and N IoT devices, and uses the FedAvg algorithm to jointly train a global model. Considering that the existing federated learning has many participants and is widely distributed, the federated learning scheme discussed in this paper is limited to the IoT network scenario. Assuming that each IoT device client $k \in N$ has a local dataset D_k, for any data sample $\{x_k, y_k\}$, x_k represents the input of the model, the learning task of the device is to find a model parameter w to describe y_k, and minimize the loss function $f_k(w)$ of the model. The loss function is used to evaluate the gap between the model prediction results and the real situation. The smaller the gap, the better the model. This paper uses $|D_i|$ to denote the dataset size of the $i-th$ device and defines the total size of the data involved in learning by $D = \sum_{i=1}^{N} D_i$. Therefore, the loss function of the $i-th$ terminal device on its dataset is defined as Eq. (3):

$$F_i(w) = \frac{1}{|D_i|} \sum_{j \in D_i} f_j(w) \tag{3}$$

According to the FedAvg algorithm, the server's global loss function can be defined as Eq. (4):

$$F(w) = \sum_{i=1}^{N} \frac{D_i}{D} F_i(w) \tag{4}$$

When DIOT-Pysyft starts training, the server initializes a global model parameter, and the terminal device optimizes this parameter. The convergence of the loss function is achieved after T global iterations. Similarly, the terminal device i needs to find the best model parameters through multiple rounds of local training on its local dataset D_i in each global iteration process, as shown in Eq. (5):

$$w_i^{(t)} = \arg\min F(w) \tag{5}$$

Due to the inherent complexity of most machine learning models, formula (5) is usually solved using Stochastic Gradient Descent (SGD) [21]. The optimal local update parameters $w_i^{(t)}$ trained on the local dataset by these terminal devices need to be uploaded to the server to participate in global aggregation. According to the FedAvg algorithm, the global aggregation process can be expressed as Eq. (6):

$$w^{(t+1)} = \sum_{i=1}^{N} \frac{D_i}{D} w_i^{(t)} \tag{6}$$

The goal of global aggregation is to minimize the loss function in formula (4), and then the server broadcasts $w^{(t+1)}$ to all end devices as the global model parameters for the next iteration. After many global iterations, the global model converges and a stable global model accuracy is finally obtained.

Encrypted Transmission of Gradients

During federated learning with DIOT-Pysyft, the server-side aggregates the gradients of the models of multiple clients, then updates the global model, and sends it to these clients. This process iterates again until the model converges to some extent. The gradient of the model update is essentially a function, which is calculated based on the initial model and local data. So this function is related to the local data to a certain extent. Although it is difficult to calculate the local data through the gradient, for the training model is a simple model, such as logistic regression. The relationship between the gradient and the local data is closely related, the local data can be obtained by solving the equations to solve the unknowns. For complex models, the optimization method of machine learning can be used to reverse, an approximate solution can be obtained from the optimization. An accurate result may not be obtained, but a roughly similar result can be obtained. In essence, the original data can still be inferred. Therefore, this paper proposes to use secure Multi-Party Computation (SMPC) to improve the FedAvg gradient aggregation algorithm to protect this gradient.

The core of SMPC is to realize the collaborative completion of model training and prediction by exchanging information on the premise of protecting that multiple

client data doesn't leave the local area. Therefore, this paper uses SMPC to improve the aggregation algorithm FedAvg method to implement secure transfer of gradients.

SMPC does not use public key and private key to encrypt gradient variables, but it splits each value into multiple parts, as shown in Fig. 3, which describes the process of obtaining the average gradient of the gradient data encryption transmission of 3 IoT clients that do not trust each other. The model gradients obtained by training the model according to the local data are W_1, W_2 and W_3. Each IoT device is set with two additional random numbers respectively, and the intermediate data is calculated by the remainder operation of the two random numbers. Each IoT device has only a fraction of the gradient-related data, the corresponding value can only be obtained when the three data participate in the operation at the same time. The final required gradient sum cannot be obtained only through a part of the gradient. Each part of the data operates like a

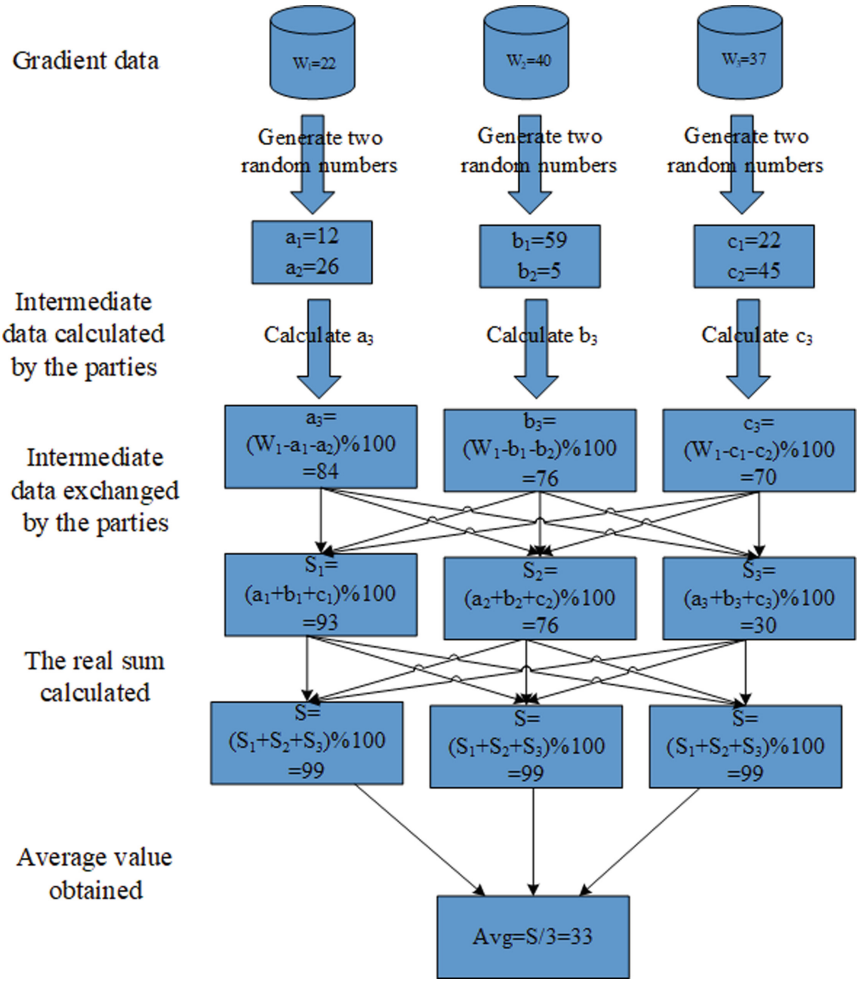

Fig. 3. Encryption process of gradient data

private key, S_1, S_2 and S_3 is the encrypted data of the three devices. SMPC can perform operations on the encrypted data. The operation results are also encrypted values, and the corresponding real values can be obtained by decrypting the operation results.

The SMPC is used to improve FedAvg aggregation algorithm to realize the encryption operation of the gradient in this paper. It does not require a trusted third party to collect the original data from all participating nodes, but only needs to exchange data which is encrypted between each participating IoT device. It is guaranteed that other participating devices cannot reverse the original plaintext data after obtaining the encrypted data, which ensures the privacy of the data of each participating IoT device.

4 Experiment and Result Analysis

4.1 Experimental Design

This paper designs and implements an IoT malicious traffic detection method based on FSKDE and federated DIOT-Pysyft, uses the public dataset UNSW-NB15 [22], which contains 9 malicious traffic generated by attacks and 1 normal traffic. The malicious attack traffic includes Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The dataset has a total of 49 features, and the training set and test set have been divided. There are 175,341 records in the training set and 82,332 records in the test set. This paper selects 40% of the data set as the experimental data.

Data loading, processing, training and testing are all done in a virtual machine. The virtual machine uses a Centos7 system with 60G storage space and 4G memory, and the processor is Intel(R) Core(TM) i5-6300HQ CPU @ 2.30 GHz 2.30 GHz, the experiment is implemented based on python3.9 programming, combined with keras, pytorch, pysyft and other libraries, the programming software used is Jupyter Notebook.

4.2 Evaluation Metrics

To evaluate the classification performance of IoT malicious traffic detection based on FSKDE and federated DIOT-Pysyft, four evaluation indicators such as accuracy are defined.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{8}$$

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{9}$$

$$F1 - Score = \frac{2 * \mathrm{Pr}\,ecision * \mathrm{Re}call}{\mathrm{Pr}\,ecision + \mathrm{Re}call} \tag{10}$$

where True Positive (TP) represents the number of correctly identified positive samples, True Negative (TN) represents the number of correctly identified negative samples, False

Positive (FP) denotes the number of wrongly identified positive samples, False Negative (FN) denotes the number of wrongly identified negative samples. The accuracy rate is the percentage of correct prediction results in the total samples; the recall rate is the probability of being predicted to be malicious traffic samples among the samples that are actually malicious traffic; the precision rate is the percentage of all samples that are predicted to be malicious traffic samples that are actually malicious traffic samples; The $F1-Score$ considers both the precision rate and the recall rate, so that the two can reach the highest at the same time and achieve a balance.

4.3 Experimental Evaluation

Effectiveness of Feature Selection Method Based on FSKDE

In the first experiment. We evaluate the effectiveness of the feature selection algorithm based on FSKDE proposed in this paper, the performance and time of malicious traffic detection using the FSKDE feature selection method and the feature selection algorithm without FSKDE (noFSKDE) on the CNN model and the MLP model are analyzed for comparison in this paper. As shown in Figs. 4 and 5, the four detection models are the CNN detection model without the FSKDE feature selection algorithm (noFSKDE_CNN), the MLP detection model without the FSKDE feature selection algorithm (noFSKDE_MLP), The CNN detection model (FSKDE_CNN) with the FSKDE feature selection algorithm and the MLP detection model with the FSKDE feature selection algorithm (FSKDE_MLP). The performance comparison is shown in Fig. 4, and the time comparison is shown in Fig. 5:

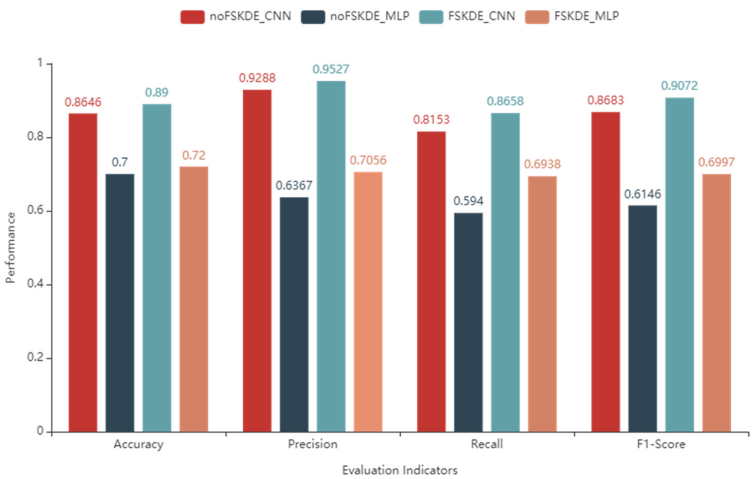

Fig. 4. Performance effectiveness for FSKDE

In terms of performance, the accuracy, precision, recall and F1-score of the noFSKDE_CNN model are 0.8646, 0.9288, 0.8153, and 0.9072 respectively, which

are 0.1646, 0.2921, 0.2213, 0.2537 higher than the detection efficiency of the noFSKDE_MLP model in terms of accuracy, precision, recall, and F1-score respectively. Therefore, this paper concludes that CNN has more advantages than MLP in detecting malicious traffic on the Internet of Things. Compared with the noFSKDE_CNN model, the FSKDE_CNN model has better accuracy, precision, and recall rate. And F1-scores. It improves 0.0254, 0.0239,0.0505, and 0.0389 than noFSKDE_CNN model, while FSKDE_MLP compared with noFSKDE_MLP improves 0.0200, 0.0689, 0.0998, and 0.0851 in accuracy, precision, recall, and F1-score, the results show that the model based on the FSKDE feature selection method proposed in this paper can remove redundant features in the dataset, find the optimal feature subset, and improve the detection indicators of the model.

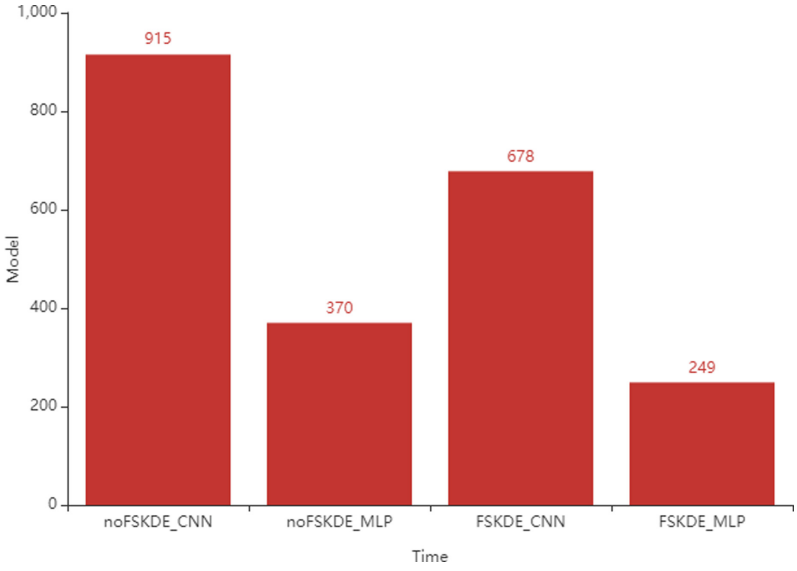

Fig. 5. Time effectiveness for FSKDE

In terms of time, as shown in Fig. 5 above, the FSKDE_CNN model saves 237s in time compared with the noFSKDE_CNN model, and the FSKDE_MLP saves 121s compared to the noFSKDE_MLP. This result shows that the FSKDE-based feature selection method proposed in this paper can reduce the running time of the model. When detecting malicious traffic, it can quickly identify malicious traffic to take effective defense measures. This method reduces the feature dimension, retains important features for malicious traffic detection, and reduces the difficulty of model learning tasks, so the running time is reduced.

In the second experiment, we compare our proposed feature selection approach with other feature selection ones. Table 1 shows the detection accuracy using the FSKDE and other proposed methods such as the greedy algorithm [23] and the CFS-DE [24]. The greedy algorithm makes use of all features and computes the detection accuracy as the initial detection accuracy, then one feature is deleted at each stage as the decremental

Table 1. The detection accuracy of the FSKDE and other feature selection methods

Method	Accuracy
The FSKDE	**0.8900**
The Greedy Algorithm [23]	0.7040
The CFS-DE [24]	0.8730

learning, and the detection accuracy is calculated for each feature removed. Table 1 shows that our proposed feature selection method can obtain 0.0890 accuracy, and the greedy algorithm just reaches to 0.7040, which is 0.186 lower than the FSKDE. The CFS-DE method calculates the variance of each feature and removes features with variance below a threshold to reduce the dimension of the features and search for the best feature subset. This method is usually used for feature selection, but it isn't created for IDS. We can get the accuracy 0.8730, which is lower than our proposed method FSKDE. The results show that the FSKDE method has stronger ability to capture malicious features in the IDS field.

Effectiveness of Traffic Detection Based on Federated DIOT-Pysyft

In order to verify the effectiveness of traffic detection based on federated DIOT-Pysyft, This paper uses federated DIoT_CNN (CNN model with DIOT-Pysyft) algorithm and federated DIoT_DNN (DNN model with DIOT-Pysyft) algorithm to conduct experiments under the federated DIOT-Pysyft system architecture. As shown in Fig. 6 below, the two detection models use the FSKDE feature selection. The experiment uses the federated DIoT_CNN algorithm with FSKDE and uses the federated DIoT_DNN algorithm with FSKDE.

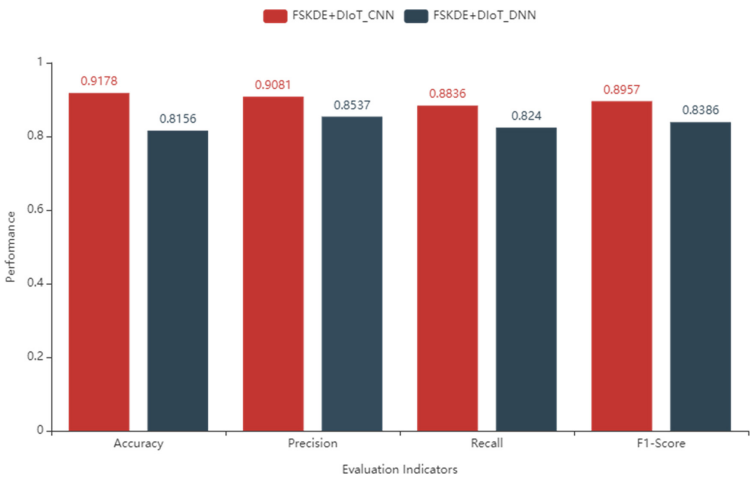

Fig. 6. Effectiveness for DIOT-Pysyft

As can be seen from Fig. 6 above, the accuracy rate of the FSKDE+DIoT_CNN algorithm is 0.9178, the precision rate is 0.9081, the recall rate is 0.8836, and the F1-score is 0.8957. Compared with the federated DIoT_DNN under the DIOT-Pysyft system architecture, the accuracy, precision, recall and F1-score are 0.1022, 0.0544, 0.0596 and 0.0564 higher respectively. The federated DIoT_CNN algorithm has more advantages than the federated DIoT_DNN in detecting malicious traffic and can learn traffic data more accurately. Therefore, this paper selects the federated DIoT_CNN as the malicious traffic detection algorithm under the DIOT-Pysyft framework.

Table 2. The performance of FSKDE+DIoT_CNN and FSKDE+CNN

Method	Accuracy	Precision	Recall	F1-Score	Time
FSKDE+CNN	0.8900	0.9527	0.8658	0.9072	678 s
FSKDE+DIoT_CNN	**0.9178**	**0.9081**	**0.8836**	**0.8957**	**247 s**

From Table 2, it can be concluded that compared with deep learning CNN, the accuracy, precision, recall rate and F1-score of federated DIoT_CNN have been improved to a certain extent when detecting malicious traffic in the IoT. Different from centralized model training, DIoT_CNN uses the respective data to train on their own devices, and each IoT device is trained in parallel. The training time is determined by the longest training time. Compared with FSKDE+CNN, the training time of FSKDE+DIoT_CNN is about 1/3 of it. we can infer that the method can achieve effective detection of malicious traffic, and reduce the model training time. Therefore, we combine FSKDE and DIoT_CNN as our final model of the federated DIOT-Pysyft system architecture.

Table 3. The detection accuracy of the DIOT-Pysyft and other federated learning approaches

Method	Accuracy
DIOT-Pysyft	**0.9178**
FILCNN [25]	0.8685
FL [26]	0.9026

Table 3 shows that the experimental results of our proposed DIOT-Pysyft approach, FILCNN [25], and FL [26] proposed by other papers are analyzed and compared to verify the advantages of the algorithm. The final experimental results show that DIOT-Pysyft has higher detection accuracy than FILCNN and FL. The detection accuracy is respectively 0.9178, 0.8685, and 0.9026. Our proposed method has more advantages in detecting malicious traffic. Besides, the improved FedAvg algorithm with SMPC further protects user privacy.

5 Conclusion

In view of the existing problems in the current Internet of Things malicious traffic detection technology, including insufficient computing power due to resource constraints, differences in attack types due to heterogeneous equipment, inaccurate detection due to less communication traffic, and insufficient data privacy protection, a malicious traffic detection method based on FSKDE and federated DIOT-Pysyft is proposed, this method combines the feature selection algorithm based on FSKDE and the traffic detection based on federated DIOT-Pysyft. The FSKDE algorithm is used on the IoT device side to filter features, reduces feature redundancy, and makes up for the problem of insufficient computing power of the IoT device. After that, the traffic detection based on federated DIOT-Pysyft algorithm is used to train each IoT device and data on their respective clients, and the gradients were encrypted and transmitted, which is used to protect the data privacy of IoT devices. At the same time, it can detect the attack type of each IoT device and reduces the problem of inaccurate detection caused by insufficient communication traffic. Compared with traditional machine learning algorithms, it has higher accuracy, precision, recall and F1-score. In the future work, each IoT device will be used to resolve the differences in bandwidth and processing speed, and reduce the differences in device performance and time.

References

1. Andrea, I., Chrysostomou, C., Hadjichristofi, G.: Internet of things: security vulnerabilities and challenges. In: 2015 IEEE symposium on computers and communication (ISCC), pp. 180–187. IEEE, Cyprus (2015)
2. Kolias, C., Kambourakis, G., Stavrou, A., et al.: DDoS in the IoT: Mirai and other botnets. Computer **50**(7), 80–84 (2017)
3. CALERO. 3 Ways the Internet of Things will Impact Enterprise Security. https://www.calero.com/mobility-service-support/3-ways-the-internet-of-things-will-impact-enterprise-security/. Accessed 17 June 2018/27 Feb 2022
4. Stankovic, J.A.: Research directions for the internet of things. IEEE Internet Things J. **1**(1), 3–9 (2014)
5. Nguyen, T.D., Marchal, S., Miettinen, M., et al.: A. DÏoT: a crowdsourced self-learning approach for detecting compromised IoT devices. ArXiv, abs/1804.07474 (2018)
6. Yang, W.C., Guo, Y.B., Zhong, Y., et al.: Anomaly detection of internet of things traffic based on device model classification and BP neural network. Inf. Netw. Secur. **11**(12) (2019)
7. Mendonça, G., Santos, G.H., e Silva, E.D.S., Leao, R.M., Menasché, D.S., Towsley, D.: An extremely lightweight approach for DDOS detection at home gateways. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 5012–5021. IEEE, USA (2019)
8. McDermott, C.D., Majdani, F., Petrovski, A.V.: Botnet detection in the internet of things using deep learning approaches. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE, Brazil (2018)
9. Elkhadir, Z., Mohammed, B.: A cyber network attack detection based on GM Median Nearest Neighbors LDA. Comput. Secur. **86**, 63–74 (2019)
10. Palmieri, F.: Network anomaly detection based on logistic regression of nonlinear chaotic invariants. J. Netw. Comput. Appl. **148**, 102460 (2019)
11. Ding, W., Jing, X., Yan, Z., et al.: A survey on data fusion in internet of things: towards secure and privacy-preserving fusion. Inf. Fusion **51**, 129–144 (2019)

12. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B. A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
13. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
14. Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.R.: DÏoT: a federated self-learning anomaly detection system for IoT. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 756–767. IEEE, Dallas (2019)
15. Al-Marri, N.A.A.A., Ciftler, B.S., Abdallah, M.M.: Federated mimic learning for privacy preserving intrusion detection. In: 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1–6. IEEE (2020)
16. Huong, T.T., et al.: Lockedge: Low-complexity cyberattack detection in IoT edge computing. IEEE Access **9**, 29696–29710 (2021)
17. Rey, V., Sánchez, P.M.S., Celdrán, A.H., Bovet, G.: Federated learning for malware detection in IoT devices. Comput. Netw. **204**, 108693 (2022)
18. Hei, X., Yin, X., Wang, Y., Ren, J., Zhu, L.: A trusted feature aggregator federated learning for distributed malicious attack detection. Comput. Secur. **99**, 102033 (2020)
19. Agrawal, S., Sarkar S, Aouedi O, et al.: Federated learning for intrusion detection system: concepts, challenges and future directions. Comput. Commun. (2022). https://doi.org/10.48550/arXiv.2106.09527
20. Meidan, Y., Bohadana, M., Mathov, Y., et al.: N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. **17**(3), 12–22 (2018)
21. Wang, S., Tuor, T., Salonidis, T., et al.: When edge meets learning: adaptive control for resource-constrained distributed machine learning. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 63–71. IEEE, USA (2018)
22. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE, Australia (2015)
23. Qin, Y., Masaaki, K.: Federated learning-based network intrusion detection with a feature selection approach. In: 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp.1–6. IEEE, Kuala Lumpur (2021). https://doi.org/10.1109/ICECCE52056.2021.9514222
24. Zhao, R., Mu, Y., Zou, L.: A hybrid intrusion detection system based on feature selection and weighted stacking classifier. IEEE Access **10**, 71414–71426 (2022). https://doi.org/10.1109/ACCESS.2022.3186975
25. Ji, X., Zhang, H., Ma, X.: A novel method of intrusion detection based on federated transfer learning and convolutional neural network. In: 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), vol. 10, pp. 338–343. IEEE, Chongqing (2022). https://doi.org/10.1109/ITAIC54216.2022.9836871
26. Mothukuri, V., Khare, P., Parizi, R.M., Pouriyeh, S., Dehghantanha, A., Srivastava, G.: Federated-learning-based anomaly detection for IoT security attacks. IEEE Internet Things J. **9**(4), 2545–2554 (2021). https://doi.org/10.1109/JIOT.2021.3077803