# Network Intrusion Detection Based on Hybrid Network Model and Federated Learning

Yuqing Kou[1,2](✉), Jieren Cheng[1,2], Yue Yang[1,3], Hao Wu[3], Yajing Li[1,2], and Victor S. Sheng[4]

[1] Hainan Blockchain Engineering Research Center, Hainan, China
1178575946@qq.com, cjr22@162.com

[2] School of Computer Science and Technology, Hainan University, Haikou 570228, China

[3] School of Cyberspace Security, Hainan University, Haikou 570228, China

[4] Department of Computer Science, Texas Tech University, Lubbock 79409, TX, USA

**Abstract.** Data is a valuable strategic resource for the development of modern society. However, with the increasingly complex network environment, privacy leaks and malicious attacks emerge in endlessly. For example, blockchain has also begun to become a new outlet for network black production, which poses a huge security threat to cryptocurrency. In this paper, we propose a hybrid network model (Cb Net), which uses Convolutional Neural Networks (CNN) and Bidirectional recurrent neural networks (BiGRU) to fully extract the space-time characteristics of network data traffic. Then, we propose an intrusion detection method (FLD), which introduces federated learning to collect traffic data from different network institutions, analyze network traffic and identify network attacks. We have fully evaluated the performance of the proposed model and method on the public dataset NSL-KDD. Experiments show that the proposed hybrid network model can achieve high detection accuracy, and the FLD method can effectively identify network attacks on the premise of ensuring the privacy of the local data of the users involved, and its performance is better than other methods.

**Keywords:** Federated learning · CNN · RNN · Network intrusion detection · NSL-KDD

## 1 Introduction

The vigorous development of the Internet has brought us great convenience as well as a great threat to network security. Network intrusion detection system (NIDS) is an important means of protection in the field of network security.

At present, deep learning technology is developing in full swing. Unlike traditional machine learning algorithms, which rely on manual design of features, deep learning uses hierarchical structures to unsupervised learn data to automatically extract features. However, the extensive application of artificial intelligence also

X. Qiu et al. (Eds.): ISCC 2023, SIST 350, pp. 152–166, 2024.
https://doi.org/10.1007/978-981-99-7161-9_12

brings us new difficulties. On the one hand, deep learning requires a lot of tag data as the basis of training, and obtaining such tag data is a time-consuming and laborious task. On the other hand, the problem of data privacy disclosure is becoming more and more serious. Data from different sources may have different user privacy, so it is difficult to share data from different sources, resulting in data islands. Especially in network intrusion detection, a single network domain can generate very limited attack behavior tags in a certain time, which means it is difficult to prevent large-scale network intrusion in time. Network traffic may leak sensitive information and user data in the network domain. Therefore, it is not possible to directly aggregate data sets from multiple network domains. How to expand the number of training data and conduct model training on large-scale network data to achieve good model training effect under the premise of ensuring data privacy and security is an important problem to be solved in the field of network intrusion detection.In such an environment where data privacy is crucial, federated learning emerged as a collaborative ML paradigm [1]. As a new distributed machine learning technology, federated learning can effectively solve the "data island problem". We expect the federated learning to play a distributed advantage in the field of network intrusion: multiple participants can effectively expand network attack data, and encrypt and exchange information and model parameters while maintaining data independence, Under the premise of ensuring user privacy, improve the detection performance of the intrusion detection system in the complex network situation.

Our contributions are summarized as follows:

1. CBNet hybrid network model is proposed, which uses CNN layer to extract local spatial hierarchical features and BiGRU layer to extract long-distance dependent features. And customized focal loss is to decrease the weight of data types that are easy to classify, so that the model can focus more on data types that are difficult to classify in the training process, thus reducing the impact of data imbalance.
2. We propose a network intrusion detection method FLD based on federated learning mechanism, which can phone traffic data from different network institutions. In the FLD, the training data of each institution is saved locally, and only parameters are transferred during the training process. In this way, data privacy and security are protected and network traffic anomalies are detected.
3. We use the NSL-KDD dataset [2] to evaluate the performance of the proposed CBNet model and FLD method, simulate the real scenario of "data island", and compare and analyze the detection performance of the centralized learning and federated methods.

## 2 Related Work

Deep learning organizes "learning algorithms" hierarchically in the form of "artificial neural networks" that can learn and make intelligent decisions on their own. Convolutional neural networks (CNN) and cyclic neural networks (RNN) The popularity of deep learning has made depth shine. In recent years, researchers have used CNN and RNN to identify cyber attacks.

Khan et al. [3] used CNN to extract two-dimensional features. Kan et al. [4] proposed the network intrusion detection method of APSO-CNN, which innovatively uses adaptive particle swarm optimization to optimize the convolutional neural network. Experiments show that this method can effectively detect network intrusion attacks. Al Turaiki et al. [5] combined dimensionality reduction with features and used CNN to extract features. Yu et al. [6] proposed a hierarchical CNN method based on packet bytes, which automatically extracts features from Pcap files. Andresini et al. [7] proposed an intrusion detection model combining GAN and CNN. In this model, the traffic is mapped to a two-dimensional image representation, and a new network attack image is generated through the network generation mode. However, CNN is not effective in dealing with long-term data dependence. RNN has the ability to extract time characteristics from input network traffic data. Alkahtani and others [8] developed the IoT intrusion detection framework. CNN, Long LSTM and CNN-LSTM are used to classify traffic. Khan [9] uses LSTM to detect time features, and AE detects global features more effectively, learning key feature representation efficiently and automatically from a large number of unmarked original network traffic data. Jothi [10] and others proposed the world integrated LSTM intelligent intrusion detection system, which has high accuracy in anomaly detection of the Internet of Things. Yang et al. [11] proposed a short-term memory network intrusion detection model based on attention mechanism, which preserves the long-term dependence between data through short-term memory network. Kurochkin et al. [12] proposed a GRU based method to detect abnormal traffic in the software definition network, and Singh et al. [13] proposed a TL based stacked GRU model with generalization and memory capabilities.

As mentioned above, researchers mainly focus on improving the deep learning algorithm to improve the accuracy of the deep learning model, but ignore the data privacy in the process of model training. In 2016, Google formally proposed federated learning [13]. Federated learning has shown a very vigorous application prospect and has been applied in many fields. Such as recommendation system field, medical image analysis,automatic driving field and so on.Based on this, we aim to study the applicability of federated learning in network intrusion detection.

## 3 Hybrid Network Model Based on CNN and BiGRU

The proposed hybrid network model has a multi-layer structure, including input layer, preprocessing layer, 1D convolution layer, BiGRU layer, attention layer, output layer, etc. The model structure is shown in Fig. 1.

### 3.1 Input Layer

Since the input of the model only accepts digital data, the NSL-KDD dataset contains non digital data, such as protocols, states, and services. Therefore, it needs to be pretreated [15]. One-hot coding: Since each network traffic data in the NSL-KDD dataset we use has character characteristics such as “protocol type”, “service” and “flag”, we need to perform numerical operations on them. Normalization: Normalization is the re-scaling of data to a specific range to reduce redundancy and shorten the training time of the model. We used the minimum-maximum normalization method to linearly transform the original data, mapping to the range of [0, 1]. The formula is:

$$X_{[i]} = \frac{X_{[i]} - X_{min}}{X_{max} - X_{min}} \tag{1}$$

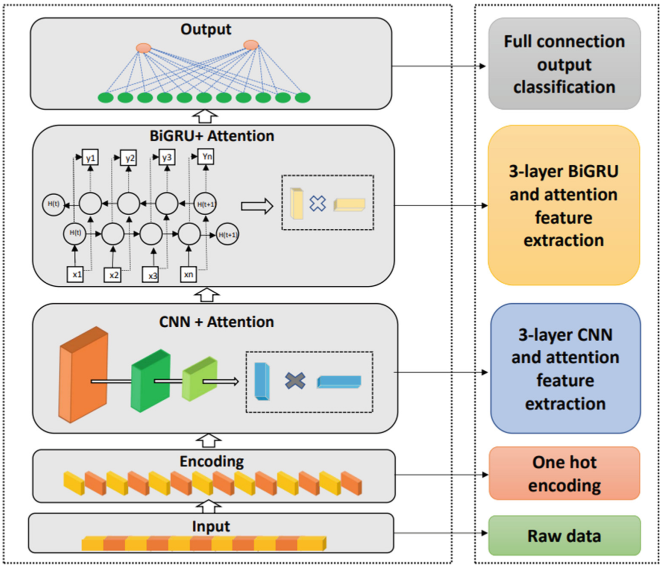
Output
BiGRU+ Attention
y1
y2
y3
Yn
H(t)
H(t+1)
x1
x2
x3
xn
CNN + Attention
Encoding
Input
Full connection output classification
3-layer BiGRU and attention feature extraction
3-layer CNN and attention feature extraction
One hot encoding
Raw data

**Fig. 1.** CBNet model diagram

### 3.2 CNN Layer

The one-dimensional data of network traffic time series input through the input layer first passes through the CNN layer, the main component of which is convolution. The feature mapping $f_m \in \mathcal{R}^{fd}$ is constructed by using the filter through the convolution operation, where f is a set of new features generated in the packet. $f_m$ is the feature mapping obtained from the feature group.

$$hl_i^{fm} = \tanh(\omega^{fm}\chi_{i:i+f-1} + b) \tag{2}$$

Filter hl generation characteristic graph representation $hl = [hl_1, hl_2, \ldots, hl_{n-f=1}]$ where $hl \in \mathcal{R}^{n-f+1}$. In order to reduce training time and prevent over-fitting, we mapped the maximum pool operation for each feature $\overrightarrow{hl} = maxhl$. Finally, the new features generated are fed to the fully connected layer containing the softmax function to obtain the probability distribution of network traffic types. The mathematical formula of the fully connected layer is:

$$o_t = softmax = (\omega_{ho} hl + b_o) \tag{3}$$

### 3.3 BiGRU Layer

The model adopts BiGRU network, and we capture the time series in CNN by passing the newly constructed feature vector to GRUs.In order to capture the time-series pattern of the newly formed feature across time steps from the maximum pool operation in CNN, the newly constructed feature mapping vector is passed to the GRUs.

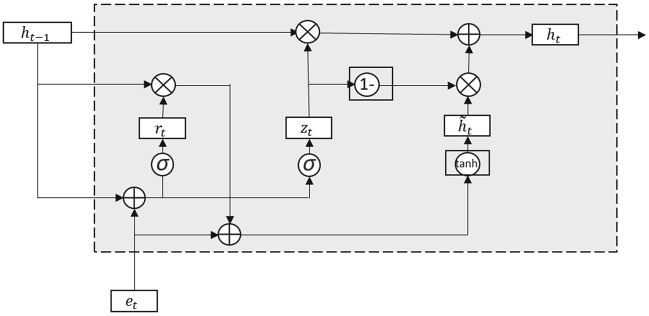

**Fig. 2.** GRU unit structure diagram

GRU network is a variant structure of RNN [16]. Compared with long-term short-term memory network LSTM, the structure is simpler and can reduce the time cost of model training. A single GRU unit is shown in Fig. 2. The state of each GRU unit is updated according to the calculation formula (4)–(7)

$$z_t = \sigma \left( W_z e_t + U_z h_t + b_z \right) \tag{4}$$

$$r_t = \sigma \left( W_r e_t + U_r h_{t-1} + b_r \right) \tag{5}$$

$$h^{'}{}_{t} = tanh \left( W_h e_t + U_h \left( r_t * h_{t-1} \right) + b_h \right) \tag{6}$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h}_t \tag{7}$$

where $h_t$ represents the output of GRU unit at the moment, and $h^{'}{}_{t}$ represents the candidate state. When $r_t$=0, $\widetilde{h}_t$ is independent of historical information $h_{t-1}$ and only related to the current input $e_t$; when $r_t$=1, $h_{t-1}$ is related to the current input and $h_{t-1}$. W, U, b and b are parameter matrices and vectors.

### 3.4 Attention Layer

Attention mechanism is introduced into 1DCNN network and BiLGRU network respectively. First, the hidden layer vector $h_t$ obtained by convolution block or BiGRU is implicitly expressed as $e_t$ through nonlinear transformation, and its expression is:

$$e_t = \ u \tan h \left( w_t h_t + b \right) \tag{8}$$

Then evaluate the importance of each flow at different time t. The normalized importance vector $\alpha_t$ is calculated by the softmax function, namely the attention weight. The fine-grained feature $s_t$ can be obtained by the weighted sum of the coarse-grained feature $\alpha_t$. According to the transformation of formula (9) and formula (10), the following is obtained:

$$\alpha_t = \frac{exp\left(e_t\right)}{\sum\limits_{j=1}^{t} exp\left(e_j\right)} \tag{9}$$

$$s_t = \sum_{t=l}^{i} \alpha_t h_t \tag{10}$$

Finally, softmax classifier is used to predict the traffic type. The formula is:

$$y = \ softmax\left(W_h s + b_k\right) \tag{11}$$

where $W_h$ and b represent classification weight and bias, and k is the number of types of network traffic.

### 3.5 Customized weight function focal loss

We use the sigmoid function as a binary classifier to distinguish normal traffic attack traffic.

$$sigmoid(h) = \frac{1}{1 + e^{-h}} \tag{12}$$

For the loss function of dichotomy, binary cross entropy BCE is selected, and its expression is:

$$BCELoss = -y_i log y_i^{'} - \left(1 - y_i\right) log\left(1 - y_i^{'}\right) \tag{13}$$

where $y_i$ indicates the real tag, and $y_i^{'}$ indicates the predicted tag value corresponding to the real tag. We use the softmax function as a multi-category classifier to distinguish attack traffic categories. For the unbalanced distribution of network intrusion samples, we used Focal loss [17] function to add modulation factor $\alpha$ and 1-$\alpha$ to reduce the loss weight of all samples in a easily classified category, and added modulation factor $(1 - y_i^{'})^{\beta}$ and $y_i^{'}$ to reduce the loss weight of a single sample in a easily classified category.

$$binary\ Loss = -y_i \alpha (1 - y_i^{'})^{\beta} log y_i^{'} - \left(1 - y_i\right)(1 - \alpha) log\left(1 - y_i^{'}\right) \tag{14}$$

$$multi\ Loss = -\frac{1}{N}\sum_{1}^{N}\sum_{q\epsilon Q} y_{s,q}\ \underset{s}{\alpha_q}(1-y^{'}_{\ s,q})^{\beta} logy^{'}_{\ s,q} \tag{15}$$

where $y_{s,q}$ indicates the real tag, and $y^{'}_{\ s,q}$ indicates the predicted tag value corresponding to the real tag. The predefined weights of class q are denoted as $\alpha_q$.

## 4 Intrusion Detection Method Based on Federated Learning

We assume that there are N distinct network institutions $\{M_1, M_2, \ldots, M_N\}$, respectively have their timing data $\{D_1, D_2 \ldots D_N\}$, and jointly train a machine learning model with its data. Traditional centralized way to collect data to the data center, and then to train the model and forecast, that is, the network data set into $D = \{D_1, D_2 \cup \ldots \cup D_N\}$ to train a model of a unified $M_{sum}$. Our proposed intrusion detection method FLD uses the federated learning paradigm to train and share the distributed cryptographic model to solve the data island problem in the real scenario of network intrusion. We use all the data of each network institution to train $M_{fl}$. In the process of model training, none of the network organizations will disclose their data to each other to ensure the security of the training process. We expect that the malicious traffic detection accuracy is close to or better than $M_{sum}$ to prove the applicability of federated learning in network intrusion detection.The learning objectives of the central server is

$$arg\ min\ L = \sum_{i=1}^{n} l\left(y_i, f_s\left(x_i\right)\right) \ \ (\omega, b) \tag{16}$$

where $\omega$ and b are the weights and bias to be learned by the central server, and $(x_i, y_i)$ represents the global data, n is the size of the central server summary data set. The learning objectives of the network institution is

$$arg\ min\ L_j = \sum_{i=1}^{n^j} l\left(y_i^j, f_s\left(x_i^j\right)\right) \ \ \left(\omega^j, y^j\right) \tag{17}$$

where $\omega^j$ represents the weight to be learned by the network mechanism; $\left(x_i^j, y_i^j\right)$ represents the time sequence data sequence of the jth network mechanism; $n^j$ is he size of the network organization data set. After each round of training, the central server aggregates the weight of model parameters of each network organization. We used FedAvg algorithm to optimize the training process. Randomly select m network mechanisms for sampling, and average the gradient updates of these m network mechanisms to form a global update. Meanwhile, the current global model is used to replace the unsampled network mechanisms with the FedAvg algorithm [18]. network

$$f_s\left(\omega, b\right) = \frac{1}{M}\sum_{m=1}^{M} f\left(\omega^j, b^j\right) \tag{18}$$

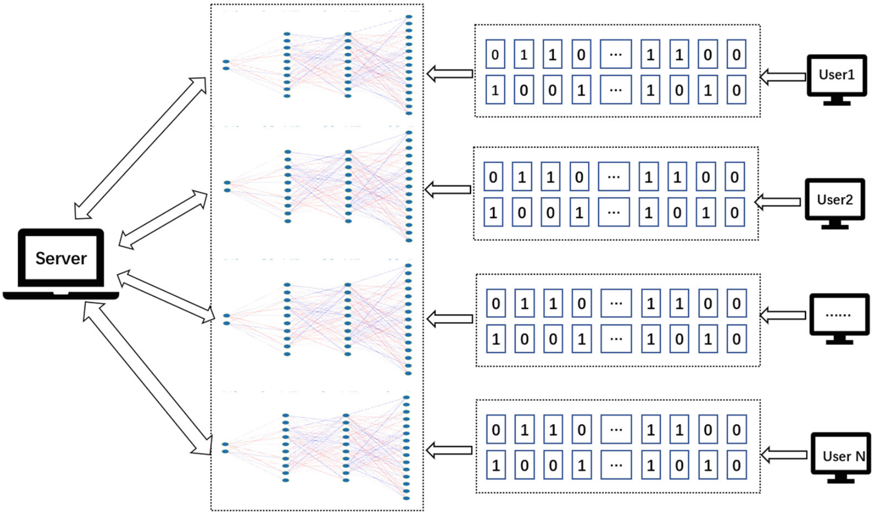

**Fig. 3.** Framework of the FLD method

The framework of the FLD method is shown in Fig. 3. It is divided into seven steps.

Step 1: The proposed hybrid network model CBNet is used as the cloud and each network institution model to identify traffic anomalies in network data, and CNN and BiGRU extract traffic sequence from the input network organization data. Step 2: Prospective network organizations download the model and do their homework. Step 3: The local data of the selected training node belong to the private state, and they are used locally to train the model. Step 4: Update the local model parameters and pass the model parameters to the central server. Step 5: After receiving updates from all network institutions, the central server aggregates model weights and creates a new update model. In this step we use the federated average algorithm to aggregate, weighting the parameters according to the local data set size. Step 6: The server returns parameters. At this time, the updated model parameters are sent back to the network institutions involved in the first round of training. Step 7: Each network organization continues to improve the model by replacing local parameters with updated global parameters.

## 5 Experiment and Evaluation

This section is divided into two parts: the implementation and evaluation process of the hybrid network model CBNet and the Federated Learning-based Intrusion detection Method (FLD). We used Keras with Tensorflow as the back end to build the model.In the future, we will take the blockchain environment as a further experimental verification environment.

### 5.1 Dataset

Public dataset NSL-KDD [19] is adopted as the experimental dataset,there are four attack types in the NSL-KDD: Dos, Probe, R2L, and U2R. Dataset information is shown in Table 1.

**Table 1.** Training set and test set information

| | Training | | Test | |
|---|---|---|---|---|
| Data category | Quantity | Proportion(%) | Quantity | Proportion(%) |
| Normal | 67343 | 53.46 | 9711 | 43.08 |
| Dos | 45927 | 36.46 | 7458 | 33.08 |
| Probe | 11656 | 9.25 | 2421 | 10.74 |
| R2L | 995 | 0.79 | 2754 | 12.22 |
| U2R | 52 | 0.04 | 200 | 0.89 |

### 5.2 Evaluation Method

At present, the evaluation indexes in the field of network intrusion detection mainly include accuracy, precision, recall rate and F1-score. These indices can be calculated according to the four basic criteria of the confusion matrix: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Several indicators are calculated as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$P = \frac{TP}{TP + FP} \tag{20}$$

$$R = \frac{TP}{TP + FN} \tag{21}$$

$$F_1 - score = \frac{2 \times P \times R}{P + R} \tag{22}$$

### 5.3 Experimental Results and Discussion

**Hybrid Network Model (CBNet) Performance Evaluation** Figure 4 shows the confusion matrix of CB-Net, a hybrid network model with two categories. We calculated the accuracy, precision, recall rate and F1-score of the hybrid network model detection by the confusion matrix, and the values of the four evaluation criteria were all up to 98.5%. Experimental results show that this model has excellent performance for binary classification. Similary, the five

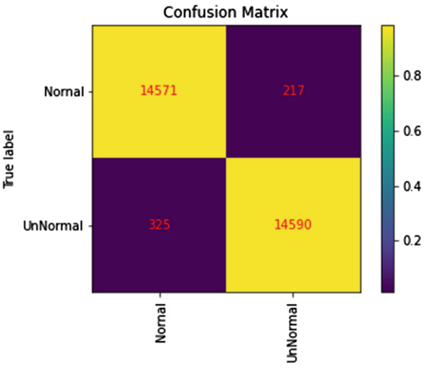

**Fig. 4.** Binary confusion matrix of CBNet

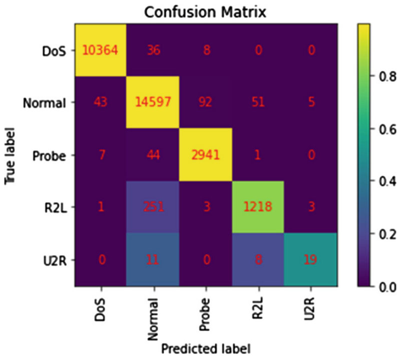

**Fig. 5.** Five classification confusion matrix of CBNet

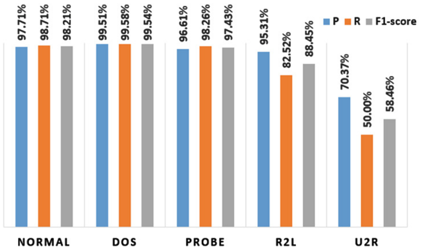

**Fig. 6.** Value of various data indicators

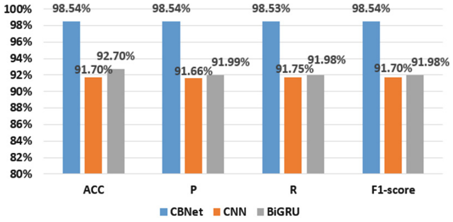

**Fig. 7.** Results of ablation experiment

categories confusion matrix of Fig. 5 shows the same excellent performance. The accuracy rate was 98.1%. Precision, recall rate and F1-score were 91.9, 85.81 and 88.42%, respectively.

In Fig. 6 , we show the accuracy, recall rate and F1-score of each category to analyze the detection effect of each data category. Despite the unbalance of class distribution in the data set, the proposed CBNet model still achieves attractive results, which proves the validity and robustness of the proposed model.

And we performed ablation experiments with CBNet model and neural network algorithm model (CNN, GRU). The comparison results are shown in In Fig. 7. Experiments show that the hybrid network model we adopted is superior to the single network model in terms of various indicators.

**Performance Evaluation of Intrusion Detection Method FLD** To simulate the situation of “data island”, we randomly divide the data set into multiple parts to represent the local data owned by each user. This data processing method can ensure that each local user lives in a completely independent network environment, that is, the distribution of intrusion attack types is completely random and independent, and different users are subject to different

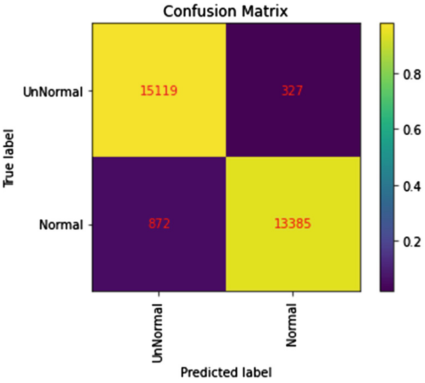

**Fig. 8.** Binary confusion matrix of FLD method

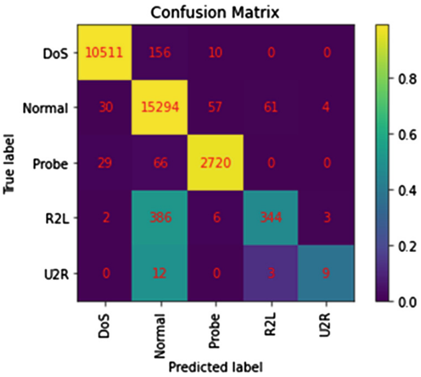

**Fig. 9.** Five class confusion matrix of FLD method

types of intrusion attacks and attacks, even some users may not have all types of intrusion attacks. If the dataset contains M pieces of data in total, and it is necessary to generate local data for N local users, then M/N pieces are directly and randomly extracted from the total dataset to form a user's local dataset.

Figures 8 and 9 show the confusion matrix obtained by the FLD method in the second and fifth classification experiments. After 20 epochs, most of the detection results basically hit the correct classification. In the second classification experiment, the accuracy rate, accuracy rate and F1 score value all reached more than 97%, which is only about 1% different from the centralized CBNet model. We believe that with the increase of epoch, the difference between the two will be smaller and smaller.

We compare the proposed CBNet model and FLD method with the classical work and recent work based on NSL-KDD dataset. The comparison information is shown in Tables 2 and 3. The detection accuracy of our CBNet model and FLD method is much higher than C4 5. Decision tree, random forest and other classical machine learning methods. Li et al. [20] used the improved Bat algorithm and random forest to detect malicious traffic. The generalization ability is not high, and the model performance trained by this method is not good enough compared with our model and method. Although Diro et al. [21] also used the deep learning method, the detection accuracy was less than 94% because they did not process high-dimensional data. Yang et al. [22] and Tian et al. [23] used deep confidence networks and did not explicitly deal with time related learning of observed variables, so the accuracy of traffic classification is not as high as our models and methods. shan Kumar et al. [24] used genetic algorithm to improve the neural network to detect traffic, but it only optimized the weight of the neural network. Our hybrid network model CBNet is an extension of NN, which can extract more effective feature information. From the experimental data, the accuracy and F1 value of CBNet model and FLD method are higher than this method.

**Table 2.** Comparison of secondary classification results

| | Comparison of secondary classification results | | | |
|---|---|---|---|---|
| | ACC | P | R | F(%) |
| C4.5 decision tree | 74.6 | / | / | / |
| Random forest | 74 | / | / | / |
| Random tree | 72.8 | / | / | / |
| SVM | 74 | / | / | / |
| Tian et al. [23] | 97.2 | 94.6 | 98.5 | 96.5 |
| shan Kumar et al. [24] | 95.5 | 97.5 | 89.4 | 93.3 |
| CBNet | 98.5 | 98.5 | 98.5 | 98.5 |
| FLD (N = 10) | 97.5 | 97.1 | 97.2 | 97.1 |

**Table 3.** Comparison of five classification results

| | Comparison of five classification results | | | |
|---|---|---|---|---|
| | ACC | P | R | F(%) |
| Li et al. [20] | 93.96 | / | / | / |
| Diro et al. [21] | 92.77 | / | / | / |
| Yang et al. [22] | 84.98 | / | / | / |
| Tian et al. [23] | 96.06 | 87 | 71.8 | 76.2 |
| shan Kumar et al. [24] | 95.6 | 90.5 | 65.2 | 69.4 |
| CBNet | 98.1 | 91.9 | 85.81 | 88.42 |
| FLD (N = 10) | 97.22 | 86.8 | 75.6 | 78 |

We use the FLD intrusion detection method to compare it with the hybrid network model (CBNet) that each user only uses its local dataset for training. We set the number of local users to N = 10, 50, 100, that is, 1/10 dataset, 1/50 dataset and 1/100 dataset are used respectively. Only the local dataset is used to train a group of users of the mixed model. Since each user generates its own model, the performance of each client is averaged to compare the results. Figure 10 shows the comparison of accuracy indicators of FLD method and CBNet model under different data set sizes. It can be clearly seen that under the same data scale, the FLD method has a higher accuracy rate for identifying abnormal traffic, and the difference is more and more obvious as the data scale decreases.

We use the F1 score value, the harmonic value of accuracy and recall, to draw a comparison chart between FLD method and CBNet model under different data scales, as shown in Fig. 11. Similar to the rule of accuracy, under the same data size, the F1 score value of FLD method is higher, and the difference will become more obvious as the data set size decreases. This shows that the FLD method with federal learning mechanism has better performance.

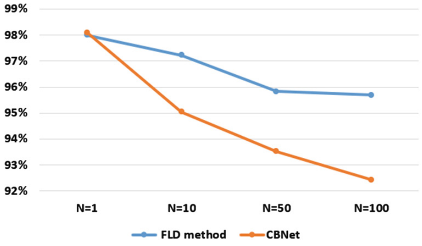

**Fig. 10.** Comparison of accuracy under different data set sizes

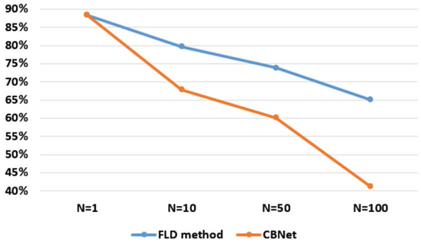

**Fig. 11.** Comparison of F1-score under different dataset sizes

As mentioned earlier, our FLD method has better detection performance than CBNet only when the data sets are of the same size. In order to find out the specific reason for the large difference in detection accuracy between the two, we further compared the specific F1 score values of each data type between the two. When N = 10, F1 score values of various data types are shown in Fig. 12. The results of Probe and Probe are basically the same, while the F1-score values of the other four data types of CBNet model centralized learning are lower than those of FLD method, and the comparison is obvious in small sample U2R and R2L.

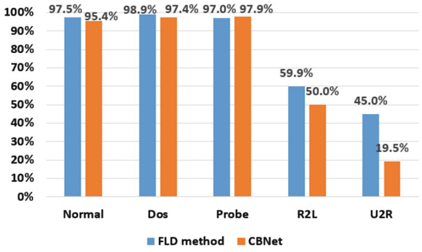

**Fig. 12.** F1-score value of various data when N = 10

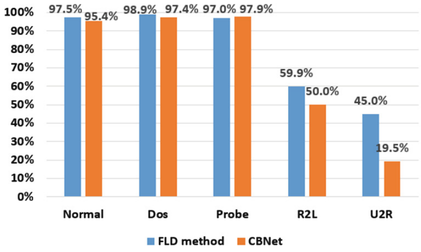

**Fig. 13.** F1-score value of various data when N = 50

As the size of the dataset decreases, when N = 50, the F1 score values of various data types are shown in Fig. 13. Our comparison results show that the F1 score of each data type using FLD method is significantly lower than CBNet centralized learning, and the F1 score of small samples U2L and U2R are more than twice as high.The reason is that in the case of random data grouping, as the data size decreases, the number of small samples R2L and U2R owned by some users is extremely low or even zero, which makes it extremely difficult for such users to detect such attacks. And with the reduction of the data size, the detection difficulty of small samples becomes more and more difficult.

## 6 Conclusion

In this paper, a hybrid network model CBNet is proposed, and the experimental results verify its excellent intrusion detection performance. In particular, we further propose a network intrusion detection method FLD based on federated learning, and the experiment proves that our FLD method can break the "data island" problem faced by current network institutions, and is feasible in the real network intrusion scenario. In the future, we will use real network traffic data such as the internal server traffic of the State Grid to further experiment.And we will conduct cross chain anomaly detection in the blockchain environment.

**Acknowledgment.** This work was supported by Hainan Provincial Natural Science Foundation of China(Grant No. 620MS021, 621QN211), the Key Research and Development Program of Hainan Province (Grant No. ZDYF2021GXJS003, ZDYF2020040), the Major science and technology project of Hainan Province (Grant No. ZDKJ2020012), National Natural Science Foundation of China (NSFC) (Grant No. 62162022, 62162024), Science and Technology Development Center of the Ministry of Education Industry-university-Research Innovation Fund (2021JQR017), Innovative research project for Graduate students in Hainan Province (Grant No.Qhyb2022-93).

## References

1. Qiang, Y.: Federal learning: the last mile of ai. CAAI Trans. Intell. Syst. **15**(1), 183–186 (2020)
2. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A: A detailed analysis of the kdd cup 99 data set. In: IEEE International Conference on Computational Intelligence for Security Defense Applications (2009)
3. Khan, M.A., Karim, M.R., Kim, Y.: A scalable and hybrid intrusion detection system based on the convolutional-lstm network. Symmetry **11**(4) (2019)
4. Kan, X., Fan, Y., Fang, Z., Cao, L., Li, X.: A novel iot network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network. Inf. Sci. (2021)
5. Al-Turaiki, I., Altwaijry, N.: A convolutional neural network for improved anomaly-based network intrusion detection. Big Data **9**(3), 233–252 (2021)
6. Yu, L., Dong, J., Chen, L., Li, M., Xu, B., Li, Z., Qiao, L., Liu, L., Zhao, B., Zhang, C.: PBCNN: packet bytes-based convolutional neural network for network intrusion detection. Comput. Netw. **194**, 108117 (2021). [Online]. Available: https://doi.org/10.1016/j.comnet.2021.108117
7. Andresini, G., Appice, A., Rose, L.D., Malerba, D.: GAN augmentation to deal with imbalance in imaging-based intrusion detection. Future Gener. Comput. Syst. **123**, 108–127 (2021). [Online]. Available: https://doi.org/10.1016/j.future.2021.04.017
8. Alkahtani, H., Aldhyani, T.H.H.: Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms. Complex 5 579 851:1–5 579 851:18 (2021). [Online]. Available: https://doi.org/10.1155/2021/5579851
9. Khan, M.A., Kim, Y.: Deep learning-based hybrid intelligent intrusion detection system. Comput. Mater. Continua **7**, 17 (2021)

10. Jothi, B., Pushpalatha, M.: Wils-trs—a novel optimized deep learning based intrusion detection framework for iot networks. In: Personal and Ubiquitous Computing, pp. 1–17
11. Yang, S., Tan, M., Xia, S., Liu, F.: A method of intrusion detection based on attention-lstm neural network. In: ICMLT 2020: 2020 5th International Conference on Machine Learning Technologies (2020)
12. Kurochkin, I.I., Volkov, S.S.: Using gru based deep neural network for intrusion detection in software-defined networks. IOP Conf. Ser. Mater. Sci. Eng. **927**(1), 012035 (2020)
13. Singh, N.B., Singh, M.M., Sarkar, A., Mandal, J.K.: A novel wide deep transfer learning stacked gru framework for network intrusion detection. J. Inf. Secur. Appl. 61 (2021)
14. Cheng, K., Fan, T., Jin, Y., Liu, Y., Yang, Q.: Secureboost: a lossless federated learning framework. Intell. Syst. IEEE (99), 1 (2021)
15. Paulauskas, N., Auskalnis, J.: Analysis of data pre-processing influence on intrusion detection using nsl-kdd dataset. In: Electrical, Electronic Information Sciences, pp. 1–5 (2017)
16. Salem, F.M.: Gated RNN: The Gated Recurrent Unit (GRU) RNN. Springer International Publishing, pp. 85–100 (2022). https://doi.org/10.1007/978-3-030-89929-5_5
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. IEEE (2), (2020)
18. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: Scaffold: stochastic controlled averaging for federated learning (2019)
19. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A: A detailed analysis of the kdd cup 99 data set. In: IEEE International Conference on Computational Intelligence for Security Defense Applications (2009)
20. Li, J., Zhao, Z., Li, R., Zhang, H.: Ai-based two-stage intrusion detection for software defined iot networks. IEEE Internet Things J. 6(2), 2093–2102 (2019). [Online]. Available: https://doi.org/10.1109/JIOT.2018.2883344
21. Diro, A.A., Chilamkurti, N.K.: Distributed attack detection scheme using deep learning approach for internet of things. Future Gener. Comput. Syst. **82**, 761–768 (2018). [Online]. Available: https://doi.org/10.1016/j.future.2017.08.043
22. Yang, Y., Zheng, K., Wu, C., Niu, X., Yang, Y.: Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Appl. Sci. 9(2) (2019)
23. Tian, Q., Han, D., Li, K., Liu, X., Duan, L., Castiglione, A.: An intrusion detection approach based on improved deep belief network. Appl. Intell. **50**(10), 3162–3178 (2020). [Online]. Available: https://doi.org/10.1007/s10489-020-01694-4
24. Kumar, G.: An improved ensemble approach for effective intrusion detection. J. Supercomput. **76**(1), 275–291 (2020). [Online]. Available: https://doi.org/10.1007/s11227-019-03035-w