2021 17th International Conference on Mobility, Sensing and Networking (MSN) | 978-1-6654-0668-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/MSN53354.2021.00047

FNEL:An Evolving Intrusion Detection System Based on Federated Never-Ending Learning

1st Tian Qin
School of Cyber Science and Engineering
Southeast University
Nanjing, China
230208959@seu.edu.cn

2nd Guang Cheng
School of Cyber Science and Engineering
Southeast University
Nanjing, China
gcheng@njnet.edu.cn

3rd Wenchao Chen
School of Cyber Science and Engineering
Southeast University
Nanjing, China
wcchen@njnet.edu.cn

4th Xuan Lei
School of Cyber Science and Engineering
Southeast University
Nanjing, China
xlei@njnet.edu.cn

Abstract—Existing intrusion detection models trained by machine learning all need reliable datasets. However, the update of the public dataset is basically long after the occurrence of the new attack, which makes the update speed of the intrusion detection model relatively slow. In this paper,we proposed a Never-Ending learning framework for intrusion detection. In this framework, the neural network model can constantly absorb the knowledge of the public/private datasets using multi-task learning and transfer learning. Meanwhile, the framework also drew on the idea of serendipitous learning, updating the model by isolating the suspected traffic from the device under attack and classifying it as a new attack category. In order to protect the privacy of users and private datasets, this paper improves various training methods of continuous learning based on the idea of federated learning. As a result, users' data will not be transmitted directly, so as to protect users' privacy.

Index Terms—Intrusion Detection; Malicious Traffic Detection; Never-Ending Learning; Federated Learning.

I. Introduction

Intrusion detection system (IDS) is an important part of system protection. With the encryption technology, access control and other technologies, malicious attacks can be intercepted [1]. However, the core of intrusion detection system still lies in its ability to distinguish malicious behavior from normal behavior [2]. Since then, researchers have been striving to build datasets for detection of different aggressive behaviors.

Through different datasets, an appropriate machine learning model can be trained to detect specific attacks in this dataset. At present, there are a large number of relevant literatures and different datasets. However, it is not timely to classify attacks by building new datasets due to the increasing emergence and implementation speed of new attack methods. In addition, various datasets target different types of attacks, and there is currently a lack of effective ways to combine the knowledge of different datasets.

Never-Ending Learning was a machine learning paradigm, and was first applied in the field of natural language processing [3]. In such framework, different learning tasks can be trained as different perspectives of learning. These tasks can be trained equivalent to each other, or multiple auxiliary tasks can be used to assist the training of the mast task. Also, this kind of framework can constantly add new learning goals to itself using self-reflection mechanism.

Drawing on the idea of never-ending Learning, the FNEL architecture designed for network intrusion detection system in this paper is shown in Figure 1.

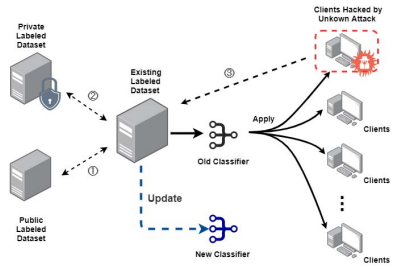

Fig. 1. Federated Never-Ending Learning Structure.

There are three ways to update this architecture as shown in the figure: ① Directly collaborate with newly released datasets; ② Collaborate with private labeled datasets based on federated learning; ③ Cluster the potential malicious traffic from the clients, and classify some traffic into a new type of attack and then retrain the old classifier.

For the part of cooperation with the new dataset, if there is little difference in features between datasets, we use transfer learning to carry out cooperative training [4]; otherwise, there is great difference in features, using multi-task learning to co-training [5]. The idea of updating from

978-1-6654-0668-0/21/$31.00 ©2021 IEEE
DOI 10.1109/MSN53354.2021.00047

attacked users borrows the idea of serendipitous learning [6]. The above three training methods are described in detail in part III of this paper.

Since traditional machine learning joint training requires aggregating data, this approach is not conducive to protecting users' privacy. Therefore, the training method used is improved based on federated learning to avoid data interaction.

This paper presents a framework for network intrusion detection system which can be updated continuously.The framework can not only update and identify new attacks through other different labelled datasets, but also identify potential unknown attacks through self-reflection mechanism. At the same time, the framework is updated with federated learning patterns to protect the privacy of the relevant data owners. The second chapter mainly introduces the related research of network intrusion detection system. The third chapter introduces the specific implementation methods of multi-tasking, transfer and serendipitous learning in federated learning scenarios. The fourth chapter introduces the datasets of CICIDS2017 [7] and CICDDoS2019 [8] used in the experiment, and introduces the neural network structures and parameters adopted. In chapter 5, we introduce the specific experimental process, including federated multitasking collaboration and federated transfer learning collaboration between labeled datasets and serendipitous learning of unknown attacks. The accuracy of attack identification was compared before and after the collaboration.

II. Related Research

Intrusion detection system has evolved into two forms: host - based and network - based since 1990 .Dewa and Maglaras proposed to apply data mining technology to the design of intrusion detection system [9], and proposes to collect data from abnormal and normal behaviors and construct statistical features to characterize them. This theory enables researchers to collect different labeled datasets through simulated attacks, so as to realize the identification of abnormal behavior.

After the construction of different datasets is successful, researchers try to train and apply different machine learning methods to detect or classify malicious behaviors: G.Zhu and J.Liao used SVM algorithm to obtain the classifier and deploy it on the network intrusion detection system [10], both misuse detection and anomaly detection are achieved successfully.S.Seo, S.Park and J.Kim used restricted boltzmann machine to classify the malicious traffic behavior in KDD99 dataset [11].A.Phadke, M.Kulkarni et al. compared a variety of machine learning algorithms and their performance on their respective datasets [12].

The artificial neural network algorithm is used to realize the classification of malicious traffic [13], but only using statistical features for input cannot play the role of deep learning feature construction.Wei Wang, Ming Zhu et al. processed PCAP messages directly and input into the convolutional neural network [14], which realizes the high-precision classification of malicious traffic. Although the above methods can realize the classification of traffic behavior, these models all rely on high-quality datasets, and only achieve the training of a single dataset.

Y.Fan, Y.Li et al. used federated transfer learning to enable the labeled datasets in private devices to assist the central server to train a better intrusion detection model [15]. This approach can be used for the collaboration of multiple datasets, but common devices rarely have the ability to analyze the type of attacks, that is, if the device suffered unknown attack, it can only determine the approximate time of attack at most. As a result, the dataset it owns should be unlabeled.

The FNEL framework proposed in this paper has two different mechanisms (multitasking learning and transfer learning) to cooperate with different datasets, and can be used to optimize the existing classifier by generalizing potential new attacks from the unlabeled data that potentially exist unknown attacks in the attacked devices.

III. Proposed collaboration pattern

In this paper, three learning modes are used to conduct joint training on different types of data. The network structure is roughly divided into three parts:

1. Alignment layer: it is used to integrate the data structure and ensure that the data is input to the network in the same dimension and similar distribution.

2. Sharing layer: Different tasks share several layers of neural networks. This part of the network uses federated learning update mode to ensure that no direct data exchange is carried out while the network is updated.

3. Task layer: different classification tasks of traffic are realized by connecting different loss calculation functions, including mast task (dichotomy of malicious and benign traffic) and auxiliary task (multiple categories of specific traffic attack types).

The overall network structure is shown in Figure 2

A. Collaborative structure of multi-tasking learning

The idea of multi-tasking learning originates from parameter sharing among several different neural networks [16]. Multi-tasking learning among the datasets of network intrusion detection can expand the total number of samples and learn the differences between different attacks and normal samples.

Although the gap between task domains may lead to negative effects of multi-tasking learning [17], in this scenario, tasks between multiple samples are all to distinguish specific malicious samples from normal samples, and the optimization objectives are roughly similar. So, directly sharing parameter is suitable in our framework.

The specific deep neural network is divided into three parts: alignment layers, sharing layers and task layers. The specific structure is shown in Figure 3.

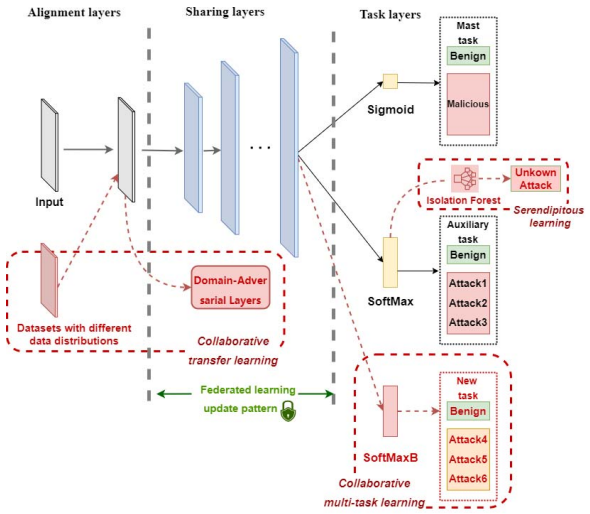

Fig. 2. The Overall Network Structure FNEL.

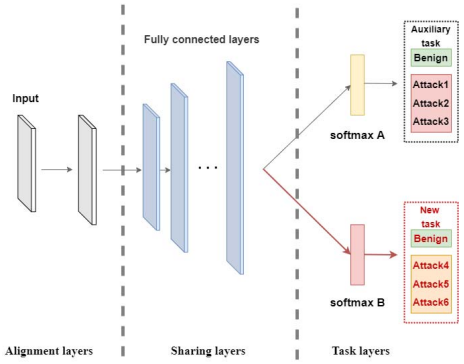

Fig. 3. Multi-task Learning Structure in FNEL.

The alignment layer is private for each data owner and is used to align input data of different dimensions. This section is not required if the data features used by the two datasets are exactly the same. If the features of two datasets are too different, transfer learning strategy should be used for further processing.

B. Collaborative structure of transfer learning

Transfer learning mainly solves the training problem in different data domains. Data features in the existing data domain are regarded as the source domain, and data features of the prospective partner are regarded as the target domain. The primary task of transfer learning is to reduce the distribution difference of data between these two domains.

Transference learning has been widely applied in computer vision and other fields. In [15], the idea of DDC [18] are adopted by using MMD (Maximum Mean Discrepancy) value to minimize the distance between domains. However, this method needs to compare the output of the middle layer of the neural network to calculate the MMD value. According to [19],if the collaborator obtains the changes of the output of the middle layer of others during collaborative learning, the original input could be deduced in some certain ways. So DDC is a good approach for working with public datasets, but not so suitable for federated learning with private datasets.

In this paper, the idea of domain adaptive neural network is used to train transfer learning. The specific structure is shown in Figure 4.

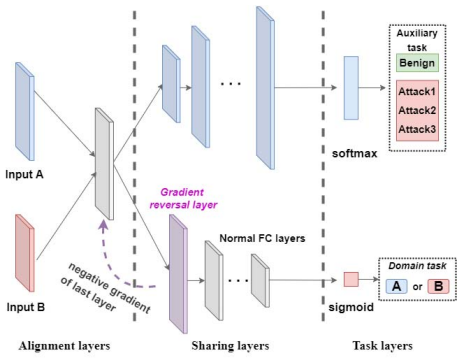

Fig. 4. Domain Adaption Transfer Learning Structure in FNEL.

When the first half of the network is updated, it receives not only the gradient transmitted from the label classification network, but also the inversion gradient from the domain classification network. In other words, the first half of the network should not only ensure the normal process of label classification, but also try to confuse the domain classifier. Finally, the inputs of different collaborators behind the gradient inversion layer can be considered to be uniformly distributed.

C. Collaborative structure using federated learning

When cooperating with private datasets, direct transmission of data should be avoided. In the training process of multi-tasking learning and transfer learning, the network layer with shared parameters should be updated by using the averaged gradient with the help of FedAvg algorithm [20]. The overall model should be integrated in the last stage.

The pseudo-code for the algorithm is given below:

In addition, the reliability of data from private datasets is difficult to distinguish. According to [21], the training of adversarial examples participation will have a great impact on the accuracy of IDE model. Even if the data partner is not malicious, it is difficult to guarantee the quality of the other party's dataset, and different degrees of label noise and sample noise will have unknown impacts on the classification effect of the neural network. Therefore, after cooperating with the private dataset, it is necessary to conduct tests on the own dataset to ensure that the model classification effect will not decline.

Algorithm 1 Classifier updating with federated leaning structure, where F:classifier for updating, F^*:updated classifier, S:Standardization processing, n:total number of data owners, L_{bw}:Gradient back propagation of loss function, G:Gradients and bias of each layer, X:data of each owner

Require: F, S, X

for $i = 1$ to n do
  $\bar{X}_i = S(X_i)$
  $G_i = L_{bw}(F(\bar{X}_i))$
  return G_i
end for
$\bar{G} = \frac{1}{n}\sum_{i=1}^{n} G_i$
Use $\bar{G}$ to update classifier
return updated classifier F^*

D. Serendipitous learning structure for unlabeled data

The purpose of serendipitous learning is to identify the data of unknown category by using unsupervised clustering method on a certain layer of output of the existing neural network model.

In the FNEL framework, this part is a self-reflection mechanism, and the process is shown in Figure 5.

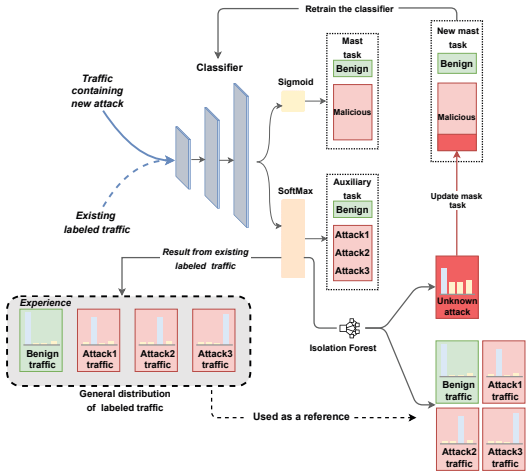

Fig. 5. Serendipitous Learning Structure in FNEL.

We assume that the device is subject to only one new type of attack over time. First, we input the unlabeled datasets that may have new attack behaviors into the existing model, and conduct cluster analysis on the data output of the previous layer of the network classification layer (SoftMax layer). Then the two types of output data are compared with the output of the same layer of the existing benign traffic data, so as to obtain a new class of unknown attack traffic. After the attack is labeled as unknown, the original neural network is retrained with the expanded dataset.

In addition, if the data labeled as unknown attacks can be accurately classified in the new task branch after the model conducts multi-task learning cooperation with other datasets, then such attacks could be identified.

IV. Dataset and neural network structure

A. Introduction to CICIDS2017 dataset

We use CICIDS2017 as our initial dataset, which contains types of attack traffic. In order to facilitate the comparison of the effects of expanding the dataset (some new DDoS attacks), we select DDoS attacks, DoS attacks and port scanning behaviors in the dataset. The number of streams of these attacks in the selected data is shown in Table III.

TABLE I
The number of different traffic in CICIDS2017

Flow Types	Flow Numbers	
	Train Set	Test Set
Benign	863248	215812
DDoS	146712	36678
PortScan	15000	3750
DoS	28804	7201

In addition, the dataset also counted a large number of statistical characteristics of traffic. In order to prevent over-fitting, we selected all features except IP address, port number and protocol.

B. Introduction to CICDDoS2019 dataset

We use the CICDDoS2019 dataset as a potential collaboration dataset. The dataset has similar traffic characteristics to CICIDS2017, but contains a large amount of new and typical DDoS attack traffic. In order to facilitate the comparison, we selected seven kinds of attack traffic and normal traffic. The specific flow numbers of these flows are shown in Table IV. In addition, we selected the flooding DDoS attack using the UDP protocol as the unknown new attack in the third experiment. The experiment in this paper also does not use IP address, port number and protocol characteristics in this dataset.

TABLE II
The number of different traffic in CICDDoS2019

Flow Types		Flow Numbers	
		Train Set	Test Set
Exp.1	Benign	29612	6548
	Syn	119984	29996
	NetBIOS	119964	29996
	SNMP	119236	29991
	MSSQL	118900	29725
	UDP	119320	29830
	TFTP	119988	29997
	NTP	105100	26275
Exp.2	Benign	29612	6548
	Syn	119984	29996
	UDP	119320	29830
Exp.3	Benign	29612	6548
	UDP	863248	215812

C. Structure and hyperparameters adopted

We used CICIDS2017 as our initial dataset, which contains types of attack traffic. In order to facilitate the comparison of the effects of expanding the dataset (some new DDoS attacks), we selected DDoS attacks, DoS attacks and port scanning behaviors in the dataset. The number of streams of these attacks in the selected data is shown in Table III.

TABLE III
The number of different traffic in CICIDS2017

Flow Types	Flow Numbers	
	Train Set	Test Set
Benign	863248	215812
DDoS	146712	36678
PortScan	15000	3750
DoS	28804	7201

In addition, the dataset also counted a large number of statistical characteristics of traffic. In order to prevent over-fitting, we selected all features except IP address, port number and protocol.

D. Introduction to CICDDoS2019 dataset

We used the CICDDoS2019 dataset as a potential collaboration dataset. The dataset has similar traffic characteristics to CICIDS2017, but contains a large amount of new and typical DDoS attack traffic. In order to facilitate the comparison, we selected seven kinds of attack traffic and normal traffic. The specific flow numbers of these flows are shown in Table IV. In addition, we selected the flooding DDoS attack using the UDP protocol as the unknown new attack in the third experiment. The experiment in this paper also does not use IP address, port number and protocol characteristics in this dataset.

TABLE IV
The number of different traffic in CICDDoS2019

Flow Types		Flow Numbers	
		Train Set	Test Set
Exp.1	Benign	29612	6548
	Syn	119984	29996
	NetBIOS	119964	29996
	SNMP	119236	29991
	MSSQL	118900	29725
	UDP	119320	29830
	TFTP	119988	29997
	NTP	105100	26275
Exp.2	Benign	29612	6548
	Syn	119984	29996
	UDP	119320	29830
Exp.3	Benign	29612	6548
	UDP	863248	215812

E. Structure and hyperparameters adopted

The purpose of this paper is to explore a cooperative training framework and only use traffic statistical characteristics as input, so the common neural network part of the main body uses the most basic three-layer multi-layer perceptron structure. The last layer uses SoftMax function to classify types of traffic and uses sigmoid function to distinguish between malicious and benign traffic. The specific number of neurons and other super parameters are shown in Table V.

TABLE V
Hyperparameters adopted

Hyperparameters	Values
Optimizer	Adam
Loss Function	Sigmoid(for Mast Task)
	SoftMax(for Auxiliary Task)
Batchsize	64
Activation Function	ReLU
Learning Rate	0.01
Neurons	[256 512 1024]

V. Experiment

As network technology continues to evolve, so do the devices, protocols and methods used to carry out attacks. A single public dataset is difficult to recognize new attack accurately. This paper aims to establish a framework which can keep model evolving with more and more partners by constantly training collectively. According to the diverse characteristics and the different labels of datasets, we put forward three different kinds of cooperation model: multitasking learning, transfer learning and serendipitous learning. In addition, in order to protect the privacy of the partners, all three models in this article have been improved using the update model of federated learning.

In this paper, partial data of CICIDS2017 and CICDDoS2019 datasets are used to simulate the above three cooperation models. Then we compared the improvement effect of the three cooperation models on the previous model to identify new attacks. The following experiments all used federated learning model for cooperation. The process of non-federated model is more concise and will not be described here.

This paper gives priority to the detection effect of attacks, that is, distinguishing attack and normal traffic is the mast task of the framework of this paper. The multi-classification tasks for specific attack types are auxiliary tasks. In this paper, the normal traffic is regarded as positive and the malicious traffic as negative. In the text, we used three metrics to evaluate the performance of our model.

1. For the dichotomy of malicious traffic and normal traffic of the main task, we considered the alarm omission rate (False Positive Rate: FPR) and false alarm rate (False Negative Rate: FNR), the detailed calculation method is shown below:

$$FPR = \frac{FP}{FP + TN} \qquad FNR = \frac{FN}{TP + FN} \tag{1}$$

2. For the subsidiary task, namely, the attack traffic classification problem, we considered the macro average

of each traffic identification precision ($macro\_P$), and the calculation formula is as follows:

$$macro\_P = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i} \tag{2}$$

Our model should minimize false positives while ensuring a minimum false positives. The average accuracy of the model should be as high as possible in predicting the related auxiliary tasks of the corresponding dataset.

A. Collaborative effect of multi-tasking learning

In this experiment, CICIDS2017 was regarded as the initial dataset and CICDDoS2019 as the dataset to be cooperated. The data sampled from the CICIDS2017 dataset (including three kinds of malicious traffic, DDoS, PortScan and DoS) were used to train the neural network in different rounds, and the model was applied respectively to the traffic sampled from the CICDDoS2019 dataset and the test set of CICIDS2017. The variation of detection accuracy is shown in Figure 6.

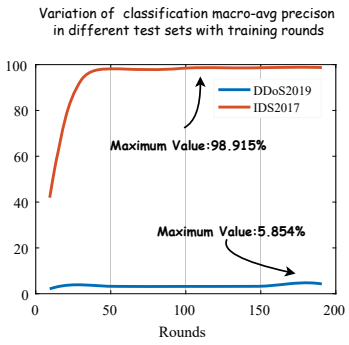

Fig. 6. Variable Graph Of Macro Average Precision Among Different Test Sets.

It can be found that the performance of the model in distinguishing the traffic in CICIDS2017 is gradually improved, but the model trained by the original dataset has a low accuracy in detecting the new attacks mentioned in the new dataset. Moreover, the effect of recognizing the unknown attacks would even decrease after the model is trained to the later stage.

The model parameters obtained in the 200th round are best detected when predicting different types of sampled data in the new dataset, and the detection results are displayed using our auxiliary multiclassification task. They were shown in Figure 7.

It can be seen that except for several attacks that can be partially detected as DDoS or DoS attacks, the probability of other attack traffic detection is almost zero. This reflects the fact that current malicious traffic identification models may not be as effective when there is attack traffic that is not contained in the original dataset.

To conduct cooperative training with the original CIC2017 training set in the way of federated multi-task

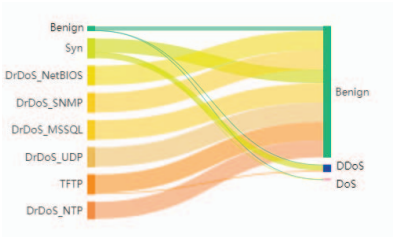

Fig. 7. Classification Results Before Multi-Task Cooperation.

learning, we sampled part of the traffic of attack types and some normal traffic of CICDDoS2019 in this paper, and divided into training set and test set according to the ratio of 0.8 and 0.2.

After the same training 200 rounds, we used the joint training model to detect the test set of CICDDoS2019. In auxiliary task, the same traffic would have two predictions simultaneously (prediction based on the attack type of CICIDS2017 and prediction based on the attack type of CICDDoS2019). We also demarcated the attacks in CICDDoS2019 dataset as malicious traffic and trained the mast task. The result based on CICDDoS2019 is shown in Figure 8.

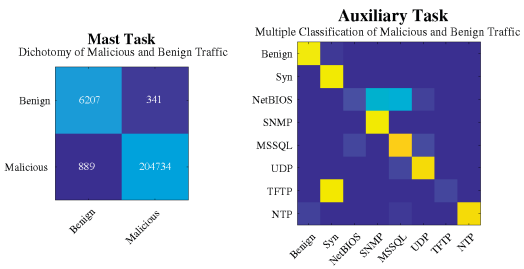

Fig. 8. DDoS Attack Classification Results After Multi-Tasking Cooperative Training.

In the prediction label based on CICDDoS2019, there are many misclassifications between attack categories. In view of the fact that multiple classifications of attacks based on traffic characteristics of CICDDoS2019 may indeed have some errors [8], this is regarded as a normal phenomenon. Even with the above misclassification among attack classes, the $macro\_P$ of multi-classification of overall traffic is still greatly improved, from 5.854% to 74.13%. Focusing only on the mast task, the original model has almost no detection effect against the new DDoS attack.After multi-task learning collaboration, the FPR and FNR of the same attack traffic identified by the model are 0.432% and 5.207% respectively.

B. Collaborative effect of transfer learning

In the training of machine learning model, data standardization can often greatly improve the effect of the model, but the overall distribution of data from different sources is not always similar. As an example, the mean

values after standardization of all features of benign flow of CICIDS2017 and CICDDoS2019 in their respective datasets are shown in Figure 9.

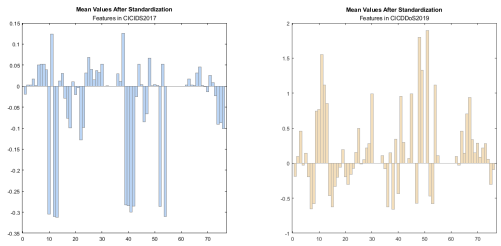

Fig. 9. Mean Values After Standardization of All Features in Different Datasets.

The DDoS attack in the CICIDS2017 dataset belongs to the flooding DDoS attack. We demoted SYN and UDP attacks in CICDDoS2019 train set as DDoS, and the model detection effect obtained based on the training of CICIDS2017 is shown in Figure 10.

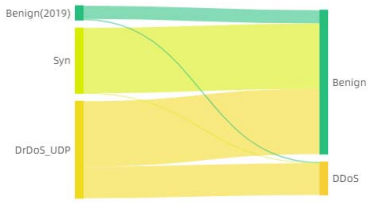

Fig. 10. Classification Results Before Transfer Learning Cooperation.

It can be seen that even if the traffic of the same kind is standardized in different datasets, the values obtained are greatly different, and this difference will directly lead to poor classification effect. In addition, each dataset may have a different kind of feature in a different order. Therefore, even if the classification target is the same, the transfer learning method is needed to unify the distribution of the dataset and then serve as the input joint training model.

In this paper, the method of domain adaptation transfer learning was used to conduct joint training for CICIDS2017 and CICDDoS2019. After 200 rounds, the accuracy of identifying UDP and SYN attacks as DDoS in CICDDoS2019 is shown in Figure 11.

Therefore, through the joint training of transfer learning, the traffic identification accuracy of the known categories in the new data distribution is greatly improved, the $macro\_P$ of multi-classification of Syn and UDP DDoS attack traffic is improved from 54.63% to 98.40%. Focusing only on the mast task, the FPR decreased significantly from 75.53% to 0.330% and the FNR also dropped from 6.26% to 0.804%.

C. Collaborative effect of serendipitous learning

In the model update scenario, it is most common that there are some new unknown attacks in the device and

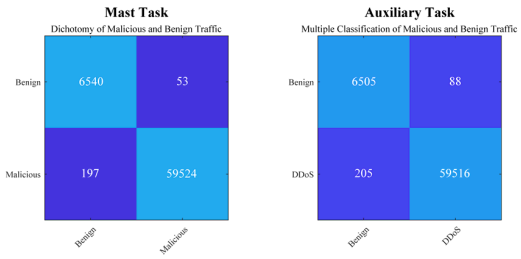

Fig. 11. DDoS Attack Classification Results After Transfer Learning Cooperative Training.

these types of traffic do not have a specific label. In this paper, we used the idea of serendipitous learning to cluster new attacks, known attacks and benign traffic by using the existing models and the output distribution of labeled data on the SoftMax layer.

Usually in the intrusion detection, only one new attack occurs in a specific period of time, and it accounts for a small proportion of the benign traffic. We used the unsupervised algorithm isolated forest to cluster the output results of the traffic in the SoftMax layer after passing the model. By comparing the isolated outliers with the distribution of known attacks, some unknown attack data can be obtained. These data can be labeled as malicious traffic and retrained as the mast task of binary classification to improve the success rate of detection of malicious traffic.

For the model trained by CICIDS2017, we used UDP DDoS attack in CICDDoS2019 as a new attack. The effect of the original model on the identification of such attacks is very unsatisfactory, as shown in the previous experiment, the missing alarm rate(FPR) is up to 75.53%. Although it is difficult for the mast mission to identify the new attacks, there are still objective differences between the new attack variants and normal traffic. New attack variants are often partially similar to one or more known attacks, so that the confidence level of classifying them as normal traffic is not particularly high.In other words, in the Softmax layer of the neural network, there will be differences between the data distribution and the normal traffic.

To try to isolate this unknown attack from other traffic, we used this part of the traffic and some of the benign traffic sending into the auxiliary task. Then we used the isolated forest algorithm to isolate the output of SoftMax layer from the outliers. We separated 10% of the traffic during this operation and the separation result is shown in Figure 12.

It can be seen that some UDP DDoS attacks are labeled as unknown attacks. This part of data was added to the training set to train the mast task, and then the UDP DDoS test set data in CICDDoS2019 was used to detected. The classification effect was shown in Figure 13.

The False Positive Rate(FPR) is dropped to 50.23%, but at the same time, the false alarm rate(FNR) is also

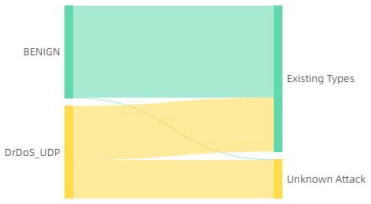

Fig. 12. Separation Effect of Traffic Containing Unknown Attack.

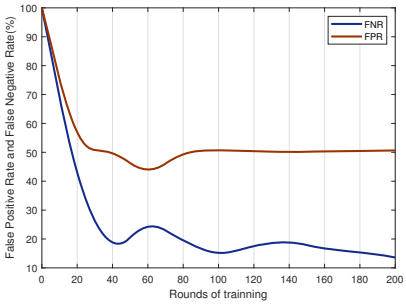

Fig. 13. Variable Graph of FPR and FNR After Serendipitous Learning.

slightly increased to around 13.60%. The reason is that the small amount of benign traffic in the traffic labeled as unknown attacks. In this part of the model should consider using some improved loss functions for the case of label noise in further research.

Discussion and Conclusion

In this paper, we provided a federated learning framework for continuous collaboration from unlabeled datasets potentially having some new categories of data, datasets containing different labels, and dataset with different distributions. In the experiment, CICIDS2017 dataset and CICDDoS2019 dataset were used to carry out the updating framework of multi-tasking, transfer and serendipitous learning respectively, which all improved the effect of the original model in detecting new attacks.

However, the initial input of the neural network in this paper is the flow statistical feature of the traffic. If the PCAP file traffic slice is used as the input, the model will be more portable. Related contents and experimental effects require further research.

Acknowledgment

This work was supported by the joint fund of the Ministry of Education of China and China Mobile (MCM20180506)

References

[1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," IEEE Communications Surveys and Tutorials, vol. 16, no. 1, pp. 303–336, 2014.

[2] F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework," in Proceedings 2002 IEEE Symposium on Security and Privacy, 2002, pp. 202–215.

[3] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, "Never-ending learning," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), 2015.

[4] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," IEEE Intelligent Systems, vol. 35, no. 4, pp. 70–82, 2020.

[5] C. T. Dinh, T. T. Vu, N. H. Tran, M. N. Dao, and H. Zhang, "Fedu: A unified framework for federated multi-task learning with laplacian regularization," 2021.

[6] Z. Dan, L. Yan, and S. Luo, "Learning beyond the predefined label space," 2011.

[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in International Conference on Information Systems Security and Privacy, 2018.

[8] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in 2019 International Carnahan Conference on Security Technology (ICCST), 2019.

[9] Z. Dewa and L. Maglaras, "Data mining and intrusion detection systems," International Journal of Advanced Computer Science and Applications, vol. 7, no. 1, 2016.

[10] G. Zhu and J. Liao, "Research of intrusion detection based on support vector machine," in International Conference on Advanced Computer Theory and Engineering, 2008.

[11] S. Seo, S. Park, and J. Kim, "Improvement of network intrusion detection accuracy by using restricted boltzmann machine," in International Conference on Computational Intelligence and Communication Networks, 2017.

[12] A. Phadke, M. Kulkarni, P. Bhawalkar, and R. Bhattad, "A review of machine learning methodologies for network intrusion detection," in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019.

[13] V. Jaiganesh, P. Sumathi, and S. Mangayarkarasi, "An analysis of intrusion detection system using back propagation neural network," in International Conference on Information Communication and Embedded Systems, 2013.

[14] W. Wei, Z. Ming, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in 2017 International Conference on Information Networking (ICOIN), 2017.

[15] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "Iotdefender: A federated transfer learning intrusion detection framework for 5g iot," in 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), 2020, pp. 88–95.

[16] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in Machine Learning, Proceedings of the Tenth International Conference, University of Massachusetts, Amherst, MA, USA, June 27-29, 1993, 1993.

[17] Y. Lu, A. Kumar, S. Zhai, C. Yu, and R. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," IEEE Computer Society, 2016.

[18] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," Computer Science, 2014.

[19] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," 2020.

[20] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," CoRR, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[21] A. Warzynski and G. Kolaczek, "Intrusion detection systems vulnerability on adversarial examples," 2018, pp. 1–4.