CONVERTER MAGAZINE
Volume 2021, No. 5

Network Abnormal Data Detection Based on Deep Learning Model

Yang Dong
China Mobile Information & Telecommunication Technology Co. Ltd., Chengdu, China

Abstract

To improve intrusion detection system performance,many algorithms are used to improve the performance of IDS systems, especially deep learning models. This paper presents an algorithm based on the model MLP, the training data set is the KDD99 data set, and the original data of the data set is vectorized by one-hot encoding, and the feature data is processed by Z-Score, and then the feature vector is encoded, and then the multi-layer perception is used The machine network performs feature learning, and finally trains the classifier model for detection. Traditional network anomaly detection algorithm models mainly use manual selection methods, and the accuracy and efficiency of classification problems are not high. This article first proposed the role of multilayer perceptron in Adam optimizer. The test of the KDD99 data set has been completed. The algorithm accuracy rate can reach 99%. For future network abnormal data detection work, an algorithm model that can realize real-time online detection is provided, which will have higher accuracy and better real-time performance.

Keywords:*Deep learning, anomaly detection, multilayer perceptron,adam algorithm*

I.Introduction

With the widespread application of computer networks in people's lives, Network security has become big problem. Network security including confidentiality of information, integrity and availability (CIA). Any attempt to undermine the CIA or bypass network security mechanisms can be regarded as a network intrusion. IDS is a security management system for monitoring network intrusion, and today's network security system is an integral part.According to the main detection technologies, IDS can be divided into two categories: one based on signature and one based on abnormal data. The signature-based detection system identifies intrusions through matching rules.However, it needs to establish a signature database, it can’t detect unknown attacks. On the contrary, Abnormality detection need a particular network, without the need for invasive characteristics priori knowledge. Therefore, although the false positive rate may be high, it can detect unknown attacks. Currently, the network structure more complex, invasive methods are increasingly diverse and complex, more difficult to detect these abnormal data intrusion detection system has brought more challenges.

1.1 Traditional machine learning detection methods

In the detection model first proposed by [1], many intrusion detection methods have been proposed. In recent years, attempts have been made to extract features from network intrusions from relatively simple statistical methods to deep learning, etc. In previous research, many methods based on traditional machine learning, including SVM[2-3], K-nearest neighbor(KNN)[4], ANN[5], random forest(RF)[6,7] and Other methods have been proposed and have been successfully used in intrusion detection systems. However, traditional methods require experts to manually design data features based on experience.

1.2 Deep learning detection methods

Deep learning technology [8] has rapidly emerged in machine learning and intrusion detection system has been applied.Research shows that this method can bring good results. In [9], deep learning can be applied to IDS. In [10], the author used a self-learning method and proved its effectiveness through experiments. However, the above methods focus on the arrangement of feature quantities and ignore the pre-training and classification capabilities of the deep learning network.

In [11], the author uses a three-layer RNN architecture to detect the data set. The input of the model selects 41 features, and the output is 4 intrusion categories. However, the nodes of each layer of the model are partially connected. The simplified RNN cannot handle complex features, and the author has not studied the binary classification capabilities.Wang et al. [12] uses convolutional neural networks to achieve network intrusion detection. However, the above research ignores the time characteristics of network data packets. Yin et al. [8] used a recurrent neural network (RNN) to detect network intrusions. The algorithm can learn the long-term dependency, and can easily lead to problems gradient disappears.Staudemeyer et al. [13] used the long-term short-term memory (LSTM) to detect network intrusions. LSTM can record historical information and discover the correlation of this information in the time domain. This paper presents a classifier can effectively detect DOS attacks and eavesdropping attack, because they have different time sequence of events. Experiments show that the accuracy of LSTM reaches 93.82%. Kimetal et al. [14] used the LSTM to train on the KDD99 data set. Through training the model, the intrusion detection is effective, and a method to reduce the false alarm rate is also proposed. However, this type of LSTM-based model has many parameters, so it takes more time to adjust the model parameters to achieve the best state, which is not conducive to real-time detection of traffic.Tang et al. [15] used a deep neural network (DNN) todetect network intrusions. This document shows that deep learning methods can achieve very good results in network anomaly detection.

Therefore, this article chooses the simplest MLP neural network structure and adopts the more applicable Adam[16] algorithm as the optimizer.Adam optimizer is one of the most popular optimizers. It is suitable for many kinds of problems, including models with sparse or noisy gradients. It’s easy to fine-tune feature makes it possible to quickly obtain good results, in fact, the default parameter configuration can usually achieve good results. Adam optimizer combines the advantages of AdaGrad[17] and RMSProp[18]. Adam uses the same learning rate for each parameter and adapts independently as learning progresses. In addition, Adam is a momentum-based algorithm that uses the historical information of the gradient. Based on these characteristics, Adam may be one of the best choices when choosing an optimization algorithm.

The traditional network anomaly detection algorithm model is mainly based on manual selection in feature selection, and the accuracy and efficiency of classification problems are not high. This article proposes an MLP network for IDS, and the network optimizer chooses Adam. The test of the KDD99[19] data set has been completed. Experiments show that the accuracy of the algorithm can reach 99%, and the structure is simple and easy to implement. An algorithm model that can realize real-time online detection is provided for future network abnormal data detection work, which will have higher accuracy and better real-time performance.

II.Methods

2.1 IDS architecture

The proposed IDS architecture is composed of three preprocessing modules, a hidden layer and an output module.The pre-processing module is divided into two sub-modules, vectorization and normalization modules. Data pre-processing the data into a standard format for the input model. The data dimensionality reduction module performs dimensionality reduction processing on the input vector and extracts features from the original input for use by the fully connected layer. The fully connected layer module consists of one 50 and one 10-node hidden layer. Output module is softmax layer, it can be standardized and outputs the classification probability. Theperformance of the multilayer perceptron is crucial. These are two different types of network and the combination of the two can get a better deep network structure.

The detection method includes two parts: training phase and testing phase. The training phase include: data pre-processing and training; the testing phase include: data pre-processing and attack detection. The preprocessing of the data set can be done at one time. Data pre-processing uses one-hot encoding and Z-Score to process the feature matrix. Extracting feature data forming a feature vector.Based on its model training for generating classifier. During the testing phase, the feature vector is input to the classifier for classification. Detected by the classifier for determining whether the input test data malicious traffic. Different layers cooperate with each other to improve model performance.

2.1.1Data pre-processing

It is a pretreatment step can obtainvectors in matrix format, andvectored non-digital features.

2.1.2 Model training

Input the training data into the network, then use the classifier to detect malicious traffic. A feature vector obtained in the previous step as neural network training inputs.

2.1.2Attack detection

Input the test data set into the trained model, and the classifier determines whether the traffic is malicious. At the same time, data preprocessing should be completed before the attack detection stage.
During MLP training, the input layer has 10 nodes, and "relu" is the activation function; The 50 nodes in the first hidden layer, the activation function here also uses the "relu" function; The second hidden layer also has 10 nodes per node, "relu" is the activation function; "Softmax" is the activation function in the output layer.

2.2Data set pre-processing

2.2.1Data feature analysis

The KDD 99 dataset is the dataset used in the 1999 KDD competition. The data set is a local area network connection data collected 9 weeks from analog USAF. The data set contains normal network traffic and different types of attack traffic.

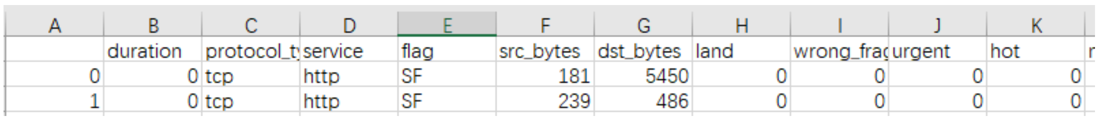
The KDD99 data set has marked each network connection as normal or attack data. Four types of abnormal network connections: dos, u2r, r2l and probe. These four types of abnormal network connections can be subdivided into 39 types of attacks.22 types of attacks occurred in the training set, and 17 other unknown types of attacks occurred in the test set.

2.2.2Standard score processing

After the data set is downloaded, use pandas to load the data set, and call the drop an method to delete the column containing the default value and return the modified object.

There is no column header (feature name) in the original KDD99 data set, you can add column headings to it and store the data in the CSV file, and add the column headings of the corresponding data to the csv file

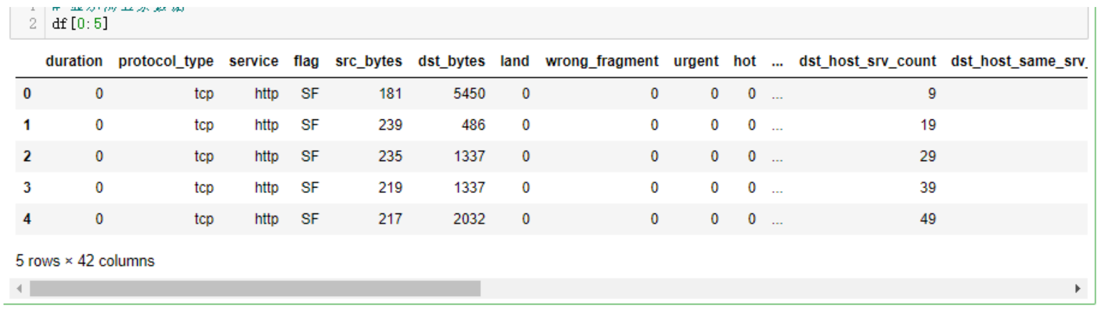
The content stored in the kdd99.csv file is as shown in Figure1.



A	B	C	D	E	F	G	H	I	J	K
	duration	protocol_t	service	flag	src_bytes	dst_bytes	land	wrong_frag	urgent	hot
0	0	tcp	http	SF	181	5450	0	0	0	0
1	0	tcp	http	SF	239	486	0	0	0	0

Fig 1: Contents of csv file

The df object after adding column headings is shown in Figure 2.



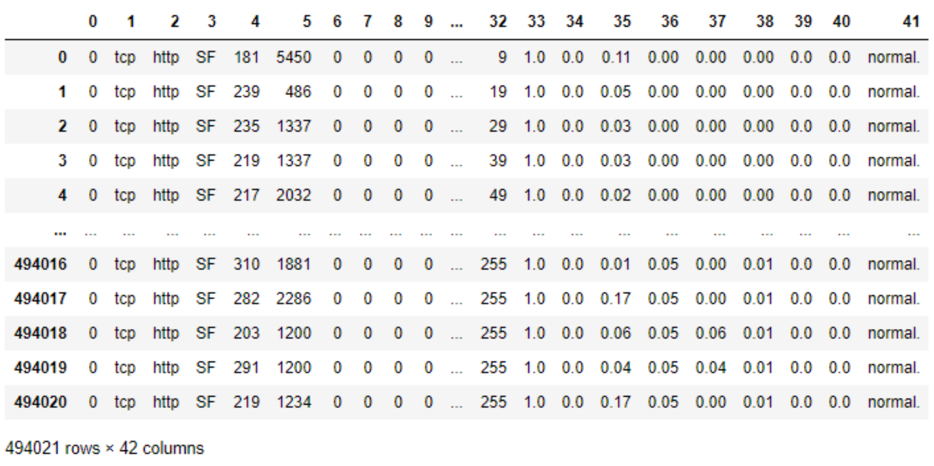
```
df[0:5]
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	

5 rows × 42 columns

Fig 2: Added column headings

The meanings of the corresponding features of the column headings are shown in Figure 3.



	0	1	2	3	4	5	6	7	8	9	...	32	33	34	35	36	37	38	39	40	41
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	0.0	0.11	0.00	0.00	0.00	0.0	0.0	normal.
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	0.0	0.05	0.00	0.00	0.00	0.0	0.0	normal.
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	0.0	0.03	0.00	0.00	0.00	0.0	0.0	normal.
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	0.0	0.03	0.00	0.00	0.00	0.0	0.0	normal.
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	0.0	0.02	0.00	0.00	0.00	0.0	0.0	normal.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
494016	0	tcp	http	SF	310	1881	0	0	0	0	...	255	1.0	0.0	0.01	0.05	0.00	0.01	0.0	0.0	normal.
494017	0	tcp	http	SF	282	2286	0	0	0	0	...	255	1.0	0.0	0.17	0.05	0.00	0.01	0.0	0.0	normal.
494018	0	tcp	http	SF	203	1200	0	0	0	0	...	255	1.0	0.0	0.06	0.05	0.06	0.01	0.0	0.0	normal.
494019	0	tcp	http	SF	291	1200	0	0	0	0	...	255	1.0	0.0	0.04	0.05	0.04	0.01	0.0	0.0	normal.
494020	0	tcp	http	SF	219	1234	0	0	0	0	...	255	1.0	0.0	0.17	0.05	0.00	0.01	0.0	0.0	normal.

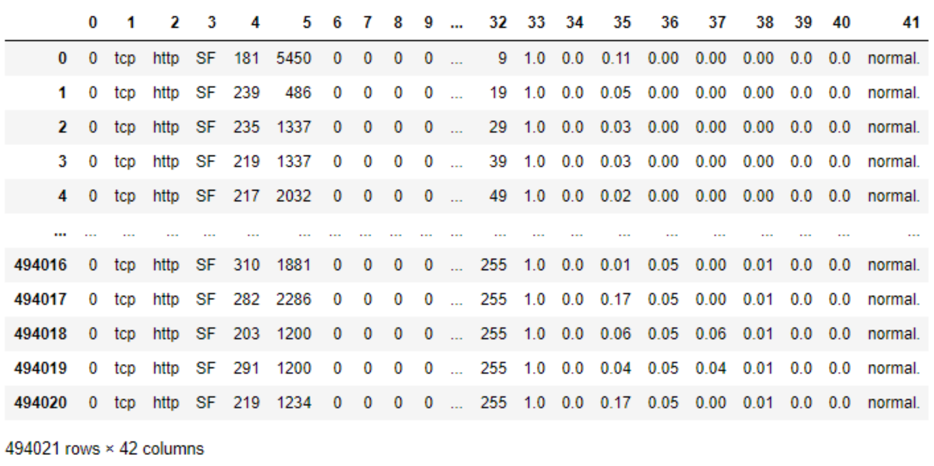
494021 rows × 42 columns

Fig 3: Data set preview

As can be seen from the figure above, each connected sample in the KDD99 dataset contains (columns 0 to 40) 41 fixed feature attributes and (column 41) 1 class identifier. The role of the class identifier is to indicate whether the connected data is "normal" or "attack".

Z-Score is also called stand score, which is generally translated as standard score in Chinese. The z-score is to convert a certain original score into a standard score that can make the original incomparable value comparable.

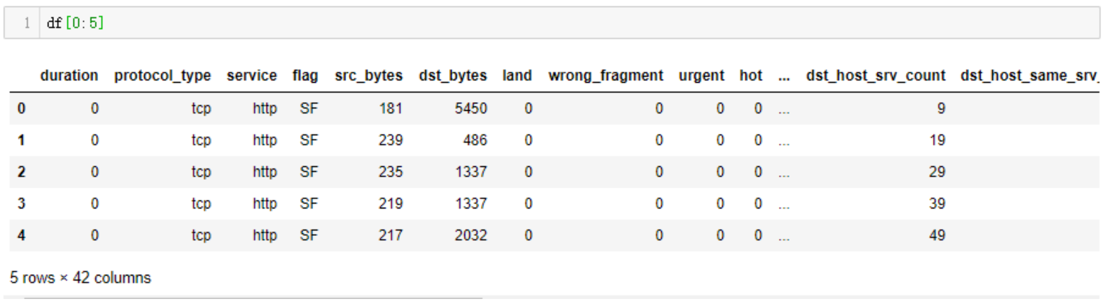
The z-score standard score can be used to compare the above-mentioned similar situations intuitively. The z-score calculation is defined as z = (x-μ)/σ. X is defined as the raw score; and z is the z-score converted; μ is the average fraction and σ is defined as the distribution of the sample space. Original data set is as shown in Figure 4. After using encode_numeric_zscore function to perform standard score processing on the "duration" column, the data set is displayed as shownin figure5.



	0	1	2	3	4	5	6	7	8	9	...	32	33	34	35	36	37	38	39	40	41
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	0.0	0.11	0.00	0.00	0.00	0.0	0.0	normal.
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	0.0	0.05	0.00	0.00	0.00	0.0	0.0	normal.
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	0.0	0.03	0.00	0.00	0.00	0.0	0.0	normal.
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	0.0	0.03	0.00	0.00	0.00	0.0	0.0	normal.
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	0.0	0.02	0.00	0.00	0.00	0.0	0.0	normal.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
494016	0	tcp	http	SF	310	1881	0	0	0	0	...	255	1.0	0.0	0.01	0.05	0.00	0.01	0.0	0.0	normal.
494017	0	tcp	http	SF	282	2286	0	0	0	0	...	255	1.0	0.0	0.17	0.05	0.00	0.01	0.0	0.0	normal.
494018	0	tcp	http	SF	203	1200	0	0	0	0	...	255	1.0	0.0	0.06	0.05	0.06	0.01	0.0	0.0	normal.
494019	0	tcp	http	SF	291	1200	0	0	0	0	...	255	1.0	0.0	0.04	0.05	0.04	0.01	0.0	0.0	normal.
494020	0	tcp	http	SF	219	1234	0	0	0	0	...	255	1.0	0.0	0.17	0.05	0.00	0.01	0.0	0.0	normal.

494021 rows × 42 columns

Fig 4: The original dataset



```
df[0:5]
```

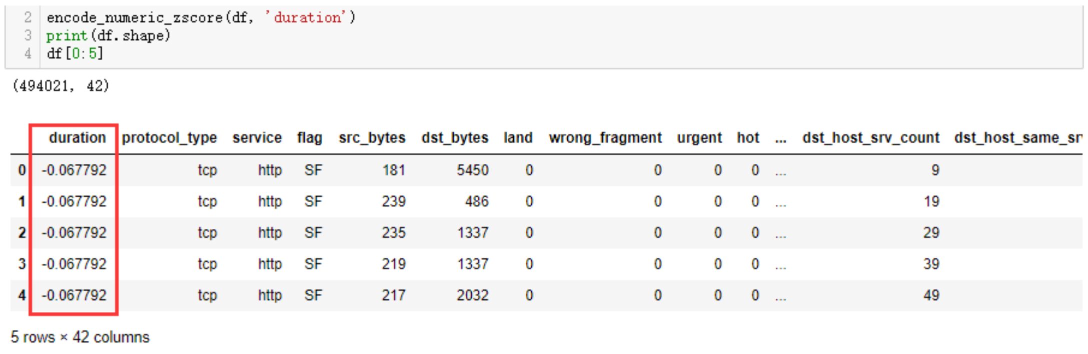
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	

5 rows × 42 columns

Fig 5: Standard score processing on the duration column

2.2.3 One-hot encoding processing
The encode_text_dummy function is used to perform one-hot processing on discrete column features, add the one-hot processed feature columns in the data set and delete the original feature columns.

Take the protocol_type feature column as an example. This column contains three discrete values "tcp", "udp" and "icmp". The original data is shown in Figure4. After the protocol_type feature column is processed using the ncode_text_dummy function, the original "protocol_type" feature column in the data set is deleted, and protocol_type-icmp, "protocol_type-tcp", and "protocol_type-udp" are added at the end of the column. The column expresses the original features in the form of one-hot encoding, shown in Figure 6.



```
encode_numeric_zscore(df, 'duration')
print(df.shape)
df[0:5]
```

```
(494021, 42)
```

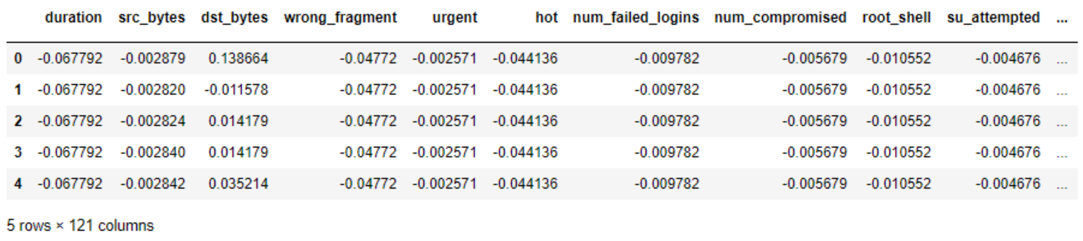
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_sr
0	-0.067792	tcp	http	SF	181	5450	0	0	0	0	...	9	
1	-0.067792	tcp	http	SF	239	486	0	0	0	0	...	19	
2	-0.067792	tcp	http	SF	235	1337	0	0	0	0	...	29	
3	-0.067792	tcp	http	SF	219	1337	0	0	0	0	...	39	
4	-0.067792	tcp	http	SF	217	2032	0	0	0	0	...	49	

5 rows × 42 columns

Fig 6: One-hot encoding of discrete column features

2.3 Create feature vector

After understanding the feature extraction method, add we can process all column features in the data set and obtain feature vectors for training.The dimension of the processed single sample data is 121, as shown in Figure 7.



	duration	src_bytes	dst_bytes	wrong_fragment	urgent	hot	num_failed_logins	num_compromised	root_shell	su_attempted	...
0	-0.067792	-0.002879	0.138664	-0.04772	-0.002571	-0.044136	-0.009782	-0.005679	-0.010552	-0.004676	...
1	-0.067792	-0.002820	-0.011578	-0.04772	-0.002571	-0.044136	-0.009782	-0.005679	-0.010552	-0.004676	...
2	-0.067792	-0.002824	0.014179	-0.04772	-0.002571	-0.044136	-0.009782	-0.005679	-0.010552	-0.004676	...
3	-0.067792	-0.002840	0.014179	-0.04772	-0.002571	-0.044136	-0.009782	-0.005679	-0.010552	-0.004676	...
4	-0.067792	-0.002842	0.035214	-0.04772	-0.002571	-0.044136	-0.009782	-0.005679	-0.010552	-0.004676	...

5 rows × 121 columns

Fig 7:Sample data after feature extraction

2.4Multilayer perceptron network

MLP (Multilayer Perceptron) neural network is a common ANN (artificial neural network) algorithm, which mainly includes an input layer, an output layer and one or more hidden layers.

MLP in all neurons are very similar. Each neuron has several inputs and one output neuron, and they will pass the same value. A typical MLP network as shown Figure 8.

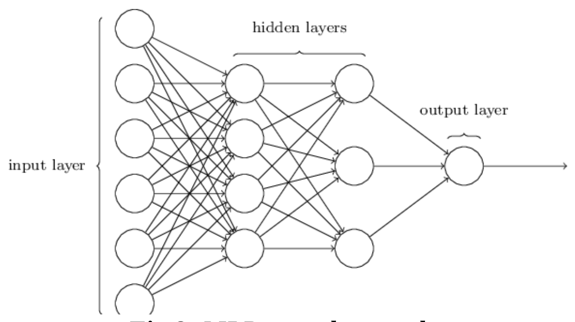


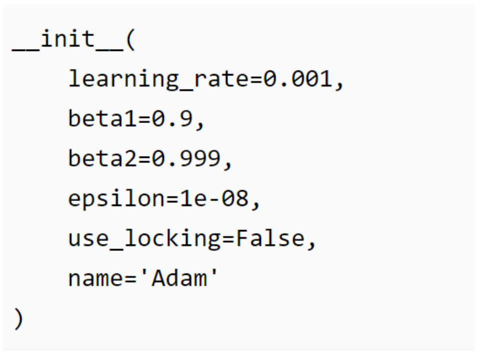
Fig 8: MLP neural network

The process of neural network is divided into forward and reverse. Reverse is usually used for training. In the forward process, each neuron contains: weight input, bias and activation function. The first layer passes the data to the second layer, and so on, and then the result is obtained. Multilayer perceptron network is calculated and predicted in this mechanism.

If the weights and the bias has been trained well. For a new input, after the above process, a predicted value can be obtained. The MLP network in this paper consists of an input layer, two hidden layers, and an output layer.

2.4 Adam optimizer

The Adam optimization algorithm is an extension of the stochastic gradient descent method, and has recently been widely used in deep learning applications in computer vision and natural language processing. Adam algorithm is easy to implement, has high computational efficiency and low memory requirements. Adam has the invariance of diagonal ratio, so it is very suitable for the network traffic problem studied in this paper. Adam estimation algorithm first, second gradient, calculation of different parameters of an adaptive learning rate.This article selects the initialization parameters as shown in Figure 9 below:



```
__init__(
    learning_rate=0.001,
    beta1=0.9,
    beta2=0.999,
    epsilon=1e-08,
    use_locking=False,
    name='Adam'
)
```

Fig 9: Adam optimizer parameters

III.Result

3.1Evaluation criteria

In the figure, ACC = (TP + TN)/(TP + TN + FP + FN), which represents the accuracy rate. All the data in the figure are accuracy rates, indicating the percentage of the correct predicted value to the total predicted value.
TPR representative of correctly predicted as the ratio of positive samples of all positive, and corresponds to the accuracy. The recall rate can also be expressed as the recall rate.

3.2. Experimental process and results

The intrusion detection is classified into two categories, the normal label is set to 0, and the attack label is set to 1. First, when training the model for the first time, the training set is divided into 15% of the data to serve as the verification set, and the training accuracy and verification accuracy are both high.

During MLP training, the input layer has 10 nodes, and "relu" is the activation function; The 50 nodes in the first hidden layer, the activation function here also uses the "relu" function; The second hidden layer also has 10 nodes per node, "relu" is the activation function; "Softmax" is the activation function in the output layer.After the model is built, call the compile method to configure it, the code is as follows:

```
model.compile(loss='categorical_crossentropy',
optimizer='adam',
metrics=['accuracy'])
```

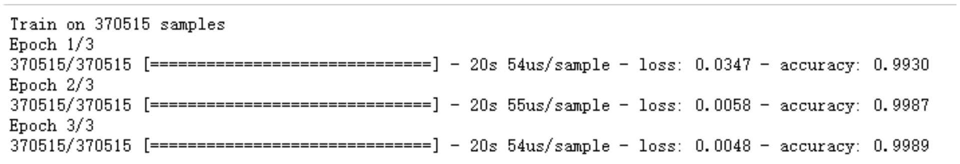
"Optimizer" specifies the optimizer instance. "Loss" refers to the classification cross entropy as a loss function. "Metrics" specifies the accuracy rate as the evaluation criterion.

Finally, call the fit method to start training.

```
model.fit(x_train,y_train,verbose=1,epochs=3)
```

The accuracy of each round of iterative training is also high, as high as 99.89%. When evaluating the model with data that has never appeared before (test set), it was found that the accuracy of the model test was about 99.9%.

After three rounds of training, the algorithm's detection accuracy on the training set has reached 0.9989, as shown in Figure 10.



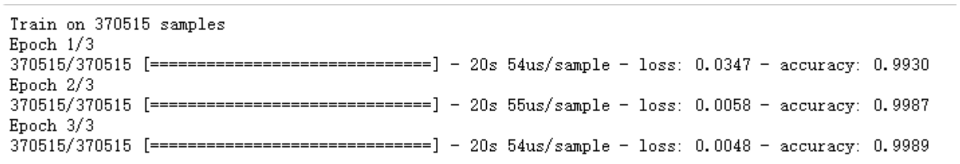
```
Train on 370515 samples
Epoch 1/3
370515/370515 [==============================] - 20s 54us/sample - loss: 0.0347 - accuracy: 0.9930
Epoch 2/3
370515/370515 [==============================] - 20s 55us/sample - loss: 0.0058 - accuracy: 0.9987
Epoch 3/3
370515/370515 [==============================] - 20s 54us/sample - loss: 0.0048 - accuracy: 0.9989
```

Fig 10: Training

After testing the test data set, the accuracy rate is obtained. The code is as follows:

```
model.evaluate(x_test, y_test)
```

The detection ACC of the test set reaches 0.9990, as shown in Figure 11.



```
Train on 370515 samples
Epoch 1/3
370515/370515 [==============================] - 20s 54us/sample - loss: 0.0347 - accuracy: 0.9930
Epoch 2/3
370515/370515 [==============================] - 20s 55us/sample - loss: 0.0058 - accuracy: 0.9987
Epoch 3/3
370515/370515 [==============================] - 20s 54us/sample - loss: 0.0048 - accuracy: 0.9989
```

Fig 11: Test result of test set

IV.Discussion

In this paper, several traditional machine learning algorithms are selected for comparative research, including SVM, KNN, Bayes, etc.;the paper also uses the following deep learning algorithms to test the data set: GRNN,PNN,RBNN[20]. The test results were compared are shown in Figure 12 and its accuracy rate asshown in figure13.

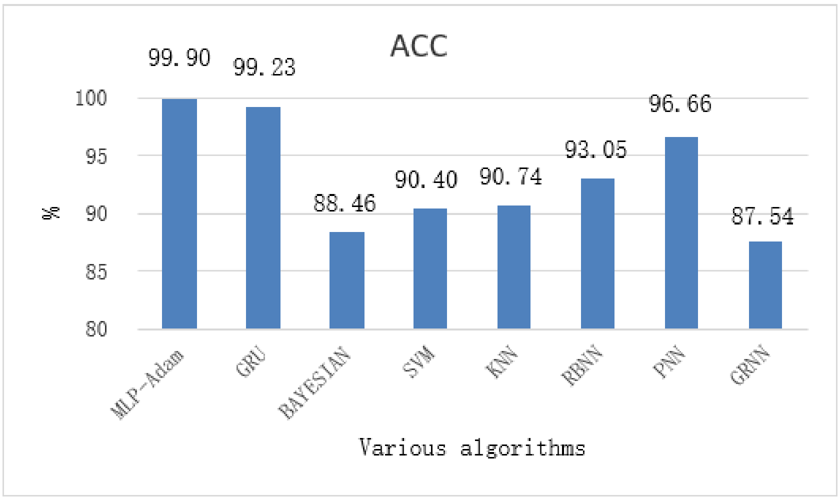


Fig 12: Experimental test results (ACC)

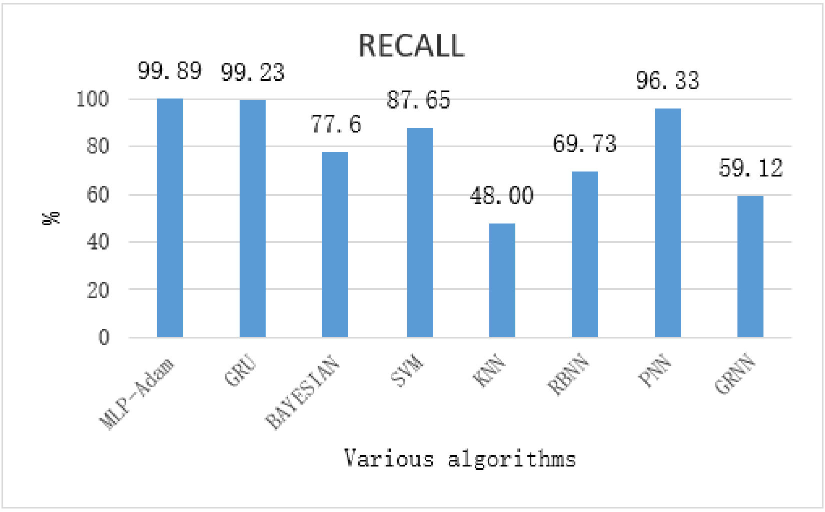


Fig 13: Experimental test results (RECALL)

The figure shows the evaluation index based on the deep learning model MLP+ADAM. From the figure data, it can be seen that the MLP+Adam model is better than other common algorithms in precision and recall rates. The traditional network anomaly detection algorithm model is mainly based on manual selection in feature selection, and the accuracy and efficiency of classification problems are not high. The main contribution of this article is to propose for the first time the role of multi-layer perceptron in the Adam optimizer. The test of the KDD99 data set has been completed. Experiments show that the accuracy of the algorithm can reach 99%, and the structure is simple and easy to implement.For future network abnormal data detection work, an algorithm model that can realize real-time online detection is provided, which will have higher accuracy and better real-time performance.

V.Conclusion

This article uses the MLP+ADAM method for the first time in network data anomaly detection, including data preprocessing, model training and model detection. We introduce standard score processing and one-key encoding to process the feature matrix.The training accuracy and test accuracy of this model are both over 99%. Although this model performs well in the KDD99 data set, as KDD99 is a relatively old data set, it may be relatively out of date for the existing network environment. The main contribution of this article is to propose for the first time the role of multi-layer perceptron in the Adam optimizer. Our experimental results have proved the feasibility of the algorithm. And its structure is simple and easy to implement. For future network abnormal data detection work, an algorithm model that can realize real-time online detection is provided, which will have higher accuracy and better real-time performance. In future research, we will focus on real-time network traffic anomaly detection technology.

References

[1] D.E. Denning, “An intrusion-detection model,” IEEE Transactions on Software Engineering, vol. 13, no. 2, pp. 222-232, 1987
[2] F. Kuang, W. Xu, S. Zhang, “A novel hybrid kpca and sym with ga model for intrusion detection,” Applied Soft Computing, vol. 18, no. C, pp. 178-184, 2014.
[3] R.R. Reddy, Y. Ramadevi, K.V.N. Sunitha,“Effective discriminant function for intrusion detection using SVM,” 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2016.
[4] W. Li, P. Yi, Y. Wu, et al., “A new intrusion detection system based on knnclassification algorithm in wireless sensor network,” Journal of Electrical & Computer Engineering, pp. 1-8, 2014.

[5] B. Ingre, A. Yadav, “Performance analysis of NSL-KDD dataset using ANN,” 2015 International Conference on Signal Processing and Communication Engineering Systems (SPACES), IEEE, 2015.

[6] N. Farnaaz, M.A. Jabbar,“Random forest modeling for network intrusion detection system,” Procedia Computer Science, vol. 89, pp. 213-217, 2016.

[7] J. Zhang, M. Zulkernine, A. Haque, “Random-forests-based network intrusion detection systems,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,vol. 38,pp. 649-659, 2008.

[8] Y. Lecun, Y. Bengio, G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436, 2015.

[9] L.O. Anyanwu, J. Keengwe, G.A. Arome, “Scalable intrusion detection with recurrent neural networks,” Seventh International Conference on Information Technology: New Generations, ITNG 2010, Las Vegas, Nevada, USA, IEEE, pp. 12-14, 2010..

[10] X. Zhang, Y. Lecun, “Text understanding from scratch,” 2015.

[11] A. Javaid, Q. Niyaz, W.Q. Sun, et al.,“A deep learning approach for network intrusion detection system. EAI Endorsed Transactions on Security and Safety. vol. 3, 2015.

[12] W. Wang, M. Zhu, X. Zeng, et al.,“Malware traffic classification using convolutional neural network for representation learning,” 2017 International Conference on Information Networking (ICOIN). IEEE, 2017.

[13] R.C. Staudemeyer, “Applying long short-term memory recurrent neural networks to intrusion detection,” South African Computer Journal, vol. 56, no. 1, pp. 136-154, 2015.

[14] J. Kim, H.L.T. Thu, H. Kim, “Long short term memory recurrent neural network classifier for intrusion detection,” International Conference on Platform Technology & Service, IEEE, 2016.

[15] T.A. Tang, L. Mhamdi, D. Mclernon, et al.,“Deep learning approach for network intrusion detection in software defined networking,” International Conference on Wireless Networks & Mobile Communications, IEEE,pp. 258-263, 2016.

[16] D.P. Kingma, J. Ba, “Adam: A method for stochastic optimization,”arXiv preprint arXiv:1412.6980, 2014.

[17] Duchi, John, et al.,“Adaptive subgradient methods for online learning and stochastic optimization,” Journal of Machine Learning Research, 2011

[18] T. Tieleman, G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” COURSERA: Neural networks for machine learning, vol. 4, no. 2, 26-31, 2012.

[19] KDD Cup 99 Dataset : https: /kdd. ics. uci. edu/databases/kddcup99/kddcup99. html.

[20] S. Devaraju, S. Ramakrishnan, “Performance comparison for intrusion detection system using neural network with kdd dataset,”ICTACT Journal on Soft Computing, vol. 4, no. 3, pp. 743-752, 2014.