DOI : https://doi.org/10.56452/7-3-110

ISSN: 0974-5823 Vol. 7 No. 3 March, 2022

International Journal of Mechanical Engineering

Workstation Network Safety Exploration Prototyping Based on Deep Learning Technique

Saksham Mittal
Lecturer, Department of Computer Science, Graphic Era Hill University, Dehradun,
Uttarakhand India 248002

***Abstract*:**

Real time network security is a big challenge in this 6G era. Even though we have antivirus, firewalls software, encryption decryption methods, but all are not enough as attackers inventing new and new approaches and find loopholes to breach the security. This research papers proposes deep learning to use as a tool to achieve the real time or workstation network security. By deep learning specially by the multilevel convolutional neural network we can find the exact data sets from the huge number of data's and can apply the network security for that exact datasets not for all datasets, which will definitely enhance to get the information of attacks and attacker in lesser time. To enhance the security we uses normalized coding deep learning algorithm, which have three layers to get the results.

keywords: multilevel convolutional neural network, encryption decryption methods , Real time network security, firewalls software, Deep learning

I. INTRODUCTION

With the deepening integration of digital, network technology and all walks of life, there are more and more network security problems [1]. While emerging digital information technologies such as virtualization, big data, next generation communication network, artificial intelligence and block chain bring us convenience, new security risks also increase [2-3]. In the face of successive major data leakage and other security emergencies, we need to strengthen the theoretical innovation research of network security, and build an intelligent, interconnected and all-round security network.

In recent years, with the continuous improvement of relevant laws and regulations and management system in the field of network security, China's network security research ability and talent team construction level have been significantly improved, and remarkable results have been achieved in international cooperation [4]. At the same time, China has clearly put forward the strategy of network power, but the network security threats from the Internet black industry chain and the national network security confrontation make the cyberspace security situation facing China increasingly complex and hidden [5]. Therefore,

DOI : https://doi.org/10.56452/7-3-110

we must strengthen the research in the field of network security, in order to deal with the rapid change of network security situation in the Internet era.

II. DEEP LEARNING METHOD

A. Common Models of Deep Learning

Deep learning method has two algorithms one is supervised learning and another is unsupervised learning algorithm. In supervised algorithm datasets are labeled already and in unsupervised algorithms we have no idea about the data samples. Semi supervised method is also available which is combination of both the supervised and unsupervised methods. Convolutional neural network model uses supervised deep learning algorithm, and on the other hand sparse self-encoder method uses unsupervised algorithms. Apart from these methods deep learning have multiple models like Deep Belief Network (DBN), Autoencoder, Deep Feedforward Network (DFN), Recurrent Neural Network (RNN) and likewise.

Convolutional neural network model is one of the types of feedforward neural network which basically practices on multidimensional array datasets. Convolutional neural network have a feature extractor module with two layers one is convolution and another is pooling layer. In this neural network one neuron is connected with all its adjacent neurons as it supports the multidimensional array structure. If we take "x" as input and "w" as kernel function then the time "t" defined in the below discrete form of convolution :

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)\, w(t-a) \quad (1)$$

In practical applications, the input is usually a high-dimensional data group, which requires convolution operation on multiple dimensions. If I is the input two-dimensional image, and K is two-dimensional kernel then:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)\, K(i-m, j-n) \quad (2)$$

Convolution is commutative and equivalent:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)\, K(m,n) \quad (3)$$

The general formula (3) is easier to apply because it changes less in the effective range of m and n.

Each convolution layer generally includes several feature maps, and each feature map is composed of neurons, which form a rectangular structure. For these neurons in different positions, we can use the same statistical feature, that is, shared weights, which is called convolution kernel. Convolution kernel is usually initialized in the form of random decimal matrix, and the appropriate weights are obtained by learning the convolution kernel in the process of network training. The weight sharing in convolution operation ensures that we do not need to learn a single parameter set for each position. Although it does not change the forward propagation time, it can significantly reduce the storage requirement of the model to k parameters, and k is usually far less than the order of m. So convolution is much better than multiplication of dense matrix in storage requirement and statistical efficiency. Figure 3 shows how weight sharing is achieved.

DOI : https://doi.org/10.56452/7-3-110

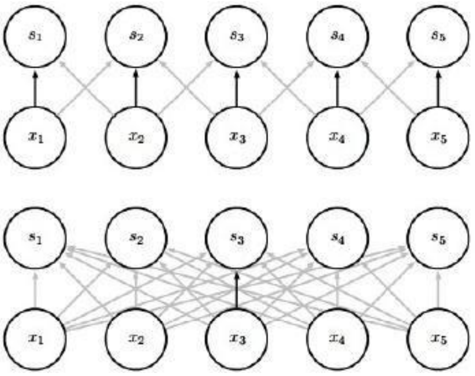


Fig.3 Weight Sharing

The black arrow indicates a connection in which special parameters are used in two different models. The black arrow above indicates the use of intermediate elements of the 3-element core (s is generated by convolution of width 3) in the convolution model. Because weights are shared, a single parameter is used for all input positions. The black tip below represents the use of the middle elements of the weight matrix in the fully connected model. The model does not use weight sharing, so it is only used once.

III. Learning Algorithm

Several classification algorithms are commonly used. Classification is a kind of data mining. Classification predicts the value of a specific attribute according to the value of other attributes. The value of a specific attribute determines that it belongs to one of several categories. In other words, classification belongs to the prediction task, which is to get an objective function f through the learning of existing data sets, and map each attribute set x to the objective attribute y, and y must be discrete.

It is difficult to give the rules of classification algorithm directly by programming, but it is easy to get them by learning algorithm. The classification process first needs to process the actual data into data that can be understood by computer (data preprocessing), generally in the form of table. If there are too many features in the learning data, we may need to select the most representative features from the feature set. Feature selection can reduce the training time, improve the performance of the learning algorithm, and avoid the dimension disaster problem. The common feature selection methods can be divided into three categories:

1) Filter method: give weight to each one-dimensional feature of the data set, and then select the feature with large weight according to the weight ranking. This feature selection method has nothing to do with the subsequent learning algorithm.

2) Wrapper: the selection of feature subset is regarded as a search optimization problem. Different subsets are generated, the selection of feature subsets is integrated into the training of learning algorithm, and the performance of learning algorithm is taken as the standard of feature subset selection.

3) Embedding: for a specific learning algorithm, learn the feature subset with the highest accuracy. That is to say, the selection process of feature subset is combined with the training process of learning algorithm. The two processes are achieved in the same optimization stage, and the selection of feature subset is automatically completed in the training process of learning algorithm.

IV intrusion detection based on Tensorflow using Softmax regression method

Softmax regression, namely multiple Logistic regression, is a commonly used multi-class classifier. Softmax function is a normalized exponential function and is defined as follows:

$$y_c = \varphi\left(z\right)_c = \frac{e^z c}{\sum_{d=1}^{c} e^z d} \quad (4)$$

In which φ stands for Softmax function, input z is a c-dimensional vector, and output y is also a c-dimensional vector. The denominator in the formula acts as a regular term, which can make:

$$\sum_{j=1}^{C} y_j = 1 \quad (5)$$

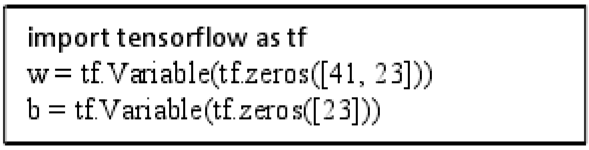
As the output layer of neural network, the value in Softmax function can be represented by c neurons. For a given input z, the probability t = c of each classification can be expressed as:

$$\begin{bmatrix} P = \left(t = 1 \mid z\right) \\ \vdots \\ P = \left(t = C \mid z\right) \end{bmatrix} = \begin{bmatrix} \varphi\left(z\right)_1 \\ \vdots \\ \varphi\left(z\right) \end{bmatrix} = \frac{1}{\sum_{d=1}^{C} e^{z_d}} \begin{bmatrix} e^{z_1} \\ \vdots \\ e^{z_c} \end{bmatrix} \quad (6)$$

⎣| |⎦ ⎣ c ⎦

In which P=(t=C | z) indicates the probability that the input data is classified as c when the input z is given.

Tensorflow provides an embedded Softmax implementation function. In order to build the Softmax classification model, it is necessary to establish the full connection between the input vector and the output category, and train the weight of each connection and the offset vector of the classification. Therefore, it is necessary to define the weight matrix and the offset term vector and give them initial values. Fig. 4 is a Python program for defining weight matrix and offset term vector.



```
import tensorflow as tf
w = tf.Variable(tf.zeros([41, 23]))
b = tf.Variable(tf.zeros([23]))
```

Fig.4 Definition of Weights and Offsets

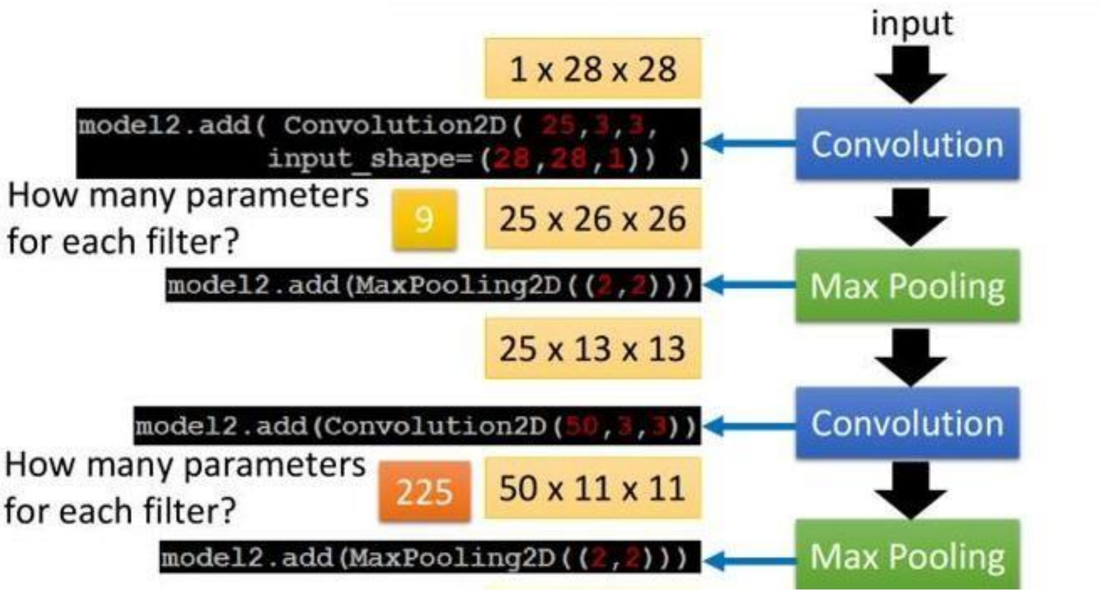
B. Feature Learning Algorithm Based on Depth Structure

1) Convolutional Neural Network (Cnn) Model

Convolutional neural network is a deep learning structure inspired by visual perception mechanism. It uses multi-layer network model to extract the features of things, and then classifies, recognizes, predicts or makes decisions according to the features. It has a wide range of applications, including image classification, target detection, target recognition, target tracking, text detection and recognition, position estimation and so on. The basic structure of CNN generally includes input layer, convolution layer, pooling layer and full connection layer. Each node of convolution layer is connected with a region of the upper layer by convolution kernel. In the same convolution layer, the weights of all neurons are the same. The pooling layer is sandwiched in the middle of the convolution layer, and its main function is to gradually compress, reduce the number of data and parameters, and also reduce the over fitting phenomenon to a certain extent, and compress a certain area of the input data of the upper layer into a value. The full join layer is mainly used for learning, mapping the learned feature representation to the sample label space.

Based on the analysis of CNN and test data set, we uses CNN training model, as shown in Figure 5.

DOI : https://doi.org/10.56452/7-3-110



```
model2.add( Convolution2D( 25,3,3,
                input_shape=(28,28,1)) )
model2.add(MaxPooling2D((2,2)))
model2.add(Convolution2D(50,3,3))
model2.add(MaxPooling2D((2,2)))
```

Fig.5 Cnn Training Model

2) *Sparse Self-Encoder (Sae) Model*

Among various network models of deep learning, sparse self-encoder is one of the algorithms that can effectively realize feature extraction. Sparse self-coding model thinks that input data can be transformed into a weighted representation of a certain set of bases. For example, integers can be expressed as the weighted representation of T=[number, ten, hundred, thousand, ten thousand, one hundred thousand ...], that is, any integer k can be expressed as: $k=\sum x_i \times T_i$ (7)

Where x is a weighting vector.

SAE, a neural network with multiple hidden layers, is an unsupervised learning method, which uses back propagation algorithm. The idea is to make the output equal to the input, and let the encoder find the hidden features (that is, the set of bases) in the input data. SAE is generally divided into coding process and decoding process. The decoding process is the reverse process of coding process, but it does not require that the decoding weight is the same as the coding weight, but is approaching through learning. The encoding process is to find T and express the input k as $\sum x_i \times T_i$, while the decoding process is to express $\sum x_i \times T_i$ as k again. Learn from the errors of output and input in every encoding and decoding process. SAE requires that the output is equal to the input, which is different from the requirements of intrusion detection (the input of intrusion detection is 41-dimensional network data, and the output is 23 categories of classification represented by 23-dimensional vector). Therefore, we changed the last decoding of SAE model into mapping to 23- dimensional vector, compared the mapping results with the actual classification results, and used its errors to learn. In this way, unsupervised learning methods have become supervised learning methods. See fig. 6 for the changed SAE model.

DOI : https://doi.org/10.56452/7-3-110

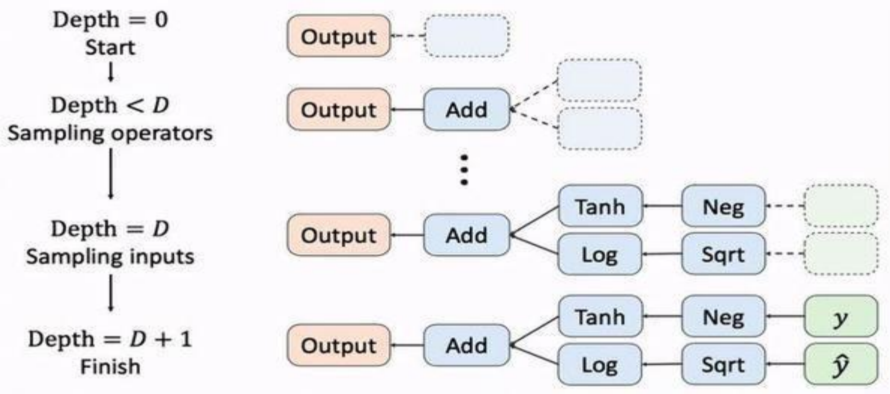


Fig.6 SAE Training Model

C. Experimental Results and Analysis

In order to verify the effectiveness of deep learning algorithm for network intrusion detection, we uses Softmax regression learning, CNN and SAE to conduct comparative experiments on NSL-KDD data set. The experiment adopts two ways to compare:

First, the Softmax regression learning method and deep learning algorithm are used to compare the intrusion detection effects, and the experimental results of the traditional Naive Bayes classification method in the literature are used to analyze them together.

The second is to compare the detection results of CNN algorithm under the depth structure and SAE algorithm, so as to choose a more suitable algorithm as the basis for further improvement. Figure 7.

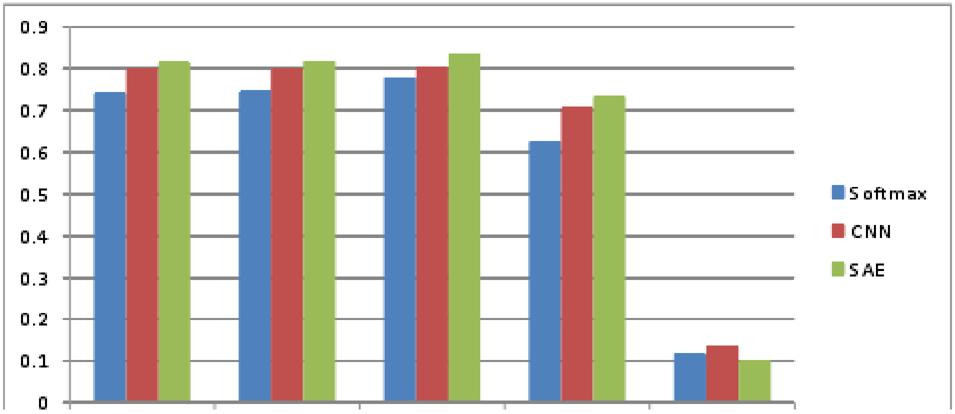


Fig.7 Comparison of Experimental Results of Deep Learning Algorithm

Based on the above results, it can be seen that these algorithms have higher detection accuracy when there are fewer types of attacks.

Ⅳ. Conclusion

In recent years, with the rapid development of Internet technology, network security has become an important issue that must be paid attention to in all fields, especially in the fields of industrial control, intelligent technology, mobile payment and cloud computing. At the same time, hackers and network terrorist organizations and other groups launched a variety of network attacks are more and more influential and destructive; India's network security

DOI : https://doi.org/10.56452/7-3-110

situation is more and more severe. Although the traditional intrusion detection system can collect and analyze the network behavior, audit data, security logs and other information to check whether the network or system is in violation of network security policies and whether there are signs of being attacked.

Intrusion attacks are regularly increasing with enhancement of technology and become a big threat for the real time applications and data security. And the available conventional intrusion detection techniques are not enough to meet the needs of reliable security aspects. But by the use of deep learning concepts its become possible to enhance efficiency with the exact tracking and also in quick.The main work of this paper is as follows: Based on the study of intrusion detection technology and challenges, the traditional learning algorithm and deep learning algorithm are compared and analyzed, and three learning algorithms, Softmax, CNN and SAE, are selected as the research objects. Among the three algorithms, Softmax is a traditional learning algorithm; CNN is good at extracting complex feature association information and has good anti-interference ability, while SAE performs better in coding.

REFERENCES

[1] Kuo, M. H. . “An intelligent agent-based collaborative information security framework.” Expert Systems with Applications, vol.32, no.2, pp.585-598, 2007.

[2] Guo, M.-H.,Liaw, H.-T.,Deng, D.-J.,Chao, H.-C. “Cluster-based secure communication mechanism in wireless ad hoc networks.” Iet Information Security, vol.4, no.4, pp.352-360, 2010.

[3] Nikooghadam, Morteza, A. Zakerolhosseini. “Secure Communication of Medical Information Using Mobile Agents.” Journal of Medical Systems, vol.36, no.6, pp.3839-3850, 2012.

[4] Alagheband, M. R , and M. R. Aref . “Dynamic and secure key management model for hierarchical heterogeneous sensor networks.” Information Security IET, vol.6, no.4, pp.271-280, 2012.

[5] Epistemological View: Data Ethics, Privacy Trust on Digital Platform, Harsh, R., Acharya, G., Chaudhary, S., 2018 IEEE International Conference on System, Computation, Automation and Networking, ICSCA 2018, 2018, 8541166

[6] Enhance the Data Security in Cloud Computing by Text Steganography, Sanghi, A., Chaudhary, S., Dave, M., Lecture Notes in Networks and Systemsthis link is disabled, 2018, 18, pp. 241–248

[7] Gupta, Ishu , and A. K. Singh . “Dynamic threshold based information leaked identification scheme.” Information Processing Letters, vol.147, no.7, pp.69-73, 2019.

[8] Carvalho, M. , et al. “Command and Control Requirements for Moving-Target Defense.” IEEE Intelligent Systems, vol.27, no.3, pp.79-85, 2012.

[9] Garcia-Magarino, Ivan , et al. “Security in networks of unmanned aerial vehicles for surveillance with an agent-based approach inspired by the principles of blockchain.” Ad Hoc Networks, vol.86, no.4, pp.72-82, 2019.

[10] Li, Jung Shian , C. J. Hsieh , and H. Y. Lin . “A hierarchical mobile-agent-based security operation center.” International Journal of Communication Systems, vol.26, no.12, pp.1503-1519, 2013.