Network Intrusion Detection System Based on One-Dimensional Convolutional Neural Networks

Jiwei Zhao[1,*†], Zeyu Zhang[2†], Peiwen Xing[3†], Jiahui Wu[4†]

[1]Northwestern Polytechnical University Xi'an, China

[2]Beihang University Beijing, China

[3]Hefei University of Technology Hefei, China

[4]South China University of Technology Guangzhou, China

*Corresponding author's e-mail: zhaojiwei@mail.nwpu.edu.cn

†These authors contributed equally.

Abstract. Network Intrusion leaks the personal information of network users on a large scale, causing serious security risks. It is of great significance to the Intrusion Detection Systems (IDS) to find abnormal traffic from a huge database in time. Traditional machine learning methods to detect abnormal network traffic usually need to manually extract features from the dataset, which is time-consuming and has low accuracy. This paper proposes a deep learning-based abnormal traffic detection method based on an Improved One-Dimensional Convolutional Neural Networks (ICNN-1D) to detect abnormal network traffic, which greatly improves the extraction accuracy of abnormal traffic features and improves the identification of attack traffic. CNN applies multiple filters (convolution kernels) to the raw pixel data of an image to extract and learn higher-level features. After multiple convolutions, the characteristic graph with the same number of categories as the number of samples is obtained. The experimental results on the dataset CIC-IDS2017 show that the accuracy of the hybrid algorithm is 99.8%. Compared with other learning algorithms, the accuracy of our method greatly improves, and the operation time has been reduced.

Keywords: network security; Intrusion Detection Systems; abnormal network traffic; deep learning; Convolutional Neural Networks (CNN).

1. Introduction

With the development of new infrastructure construction such as 5G, artificial intelligence, and industrial internet, high-speed networks and smart applications have enriched our daily life. However, hidden dangers such as viruses and vulnerabilities also come with them. Escalating cyber-attacks can steal data and destroy resources, which have greatly hindered social development. According to the report released by cyber security company Check Point Research, weekly attacks on corporate networks increased by 50 percent in 2021 [1]. Considering the massive, high-dimensional, and diverse network traffic, traditional anomaly detection techniques appear to be powerless in the face of new network vulnerabilities and escalating network attacks.

Many machine learning algorithms, including supervised and unsupervised learning have been widely used for traffic anomaly detection. In existing anomaly detection research work, Chen Shiwen et al. [2] transformed DDoS attack detection into a two-classification problem of machine learning, using three machine learning models: hidden Markov model, twin support vector machine, and conditional random field, to realize probabilistic point detection, classification hyperplane detection, and conditional random field detection. Rao Lan et al. [3] used the Support Vector Machine (SVM) algorithm to detect network attacks. Constructing a new RBF (Radial Basis Function) kernel function solved the difference measurement of heterogeneous data reasonably and scientifically, and the detection accuracy was improved.

However, for most traditional machine learning algorithms which rely on datasets that have been labeled as normal network behavior, manual extraction and selection of network traffic feature sets are primarily needed, which requires extensive feature engineering and complex task processing.

While among the unsupervised learning methods, K-means clustering methods [4, 5], which could detect anomalies by classifying data into normal traffic clusters and abnormal traffic clusters, are widely used for anomaly detection tasks due to their simplicity. However, the direct use of clustering algorithms can detect anomalies in high-traffic areas, they ignore anomalies in low-traffic areas. In addition, for the anomaly detection method based on traffic pattern K-means clustering [6], the problem of large-scale long-term sequence detection suffers from a limited number of processing regions, limited data processing time, and low accuracy.

To improve the performance of Intrusion Detection Systems (IDS), this paper applies ICNN-1D to detect anomalous traffic. It is to perform layer-by-layer convolution and pooling of input data to accurately extract input features, thereby ensuring the accuracy of data classification. The ICNN-1D in this paper does not perform a pooling operation after each convolution to preserve the complete traffic information as much as possible. According to the experimental results, the method has improved detection effect significantly compared with the existing traditional machine learning algorithms, and the accuracy rate reaches 99.8%.

2. Method

This section describes the proposed method. Primarily, we briefly describe the dataset and baselines we use for the experiment (Sec. A). Then the network traffic data in pcap format is transformed into two-dimensional grays-scale images that can be computed by the convolutional neural network through data preprocessing (Sec. B). Finally, we construct a convolutional neural network combined with the global pooling method detailed below to perform feature extraction and classification tasks (Sec. C). The accurate classification of network traffic can be performed after continuous learning of traffic features by the model to achieve intrusion detection.

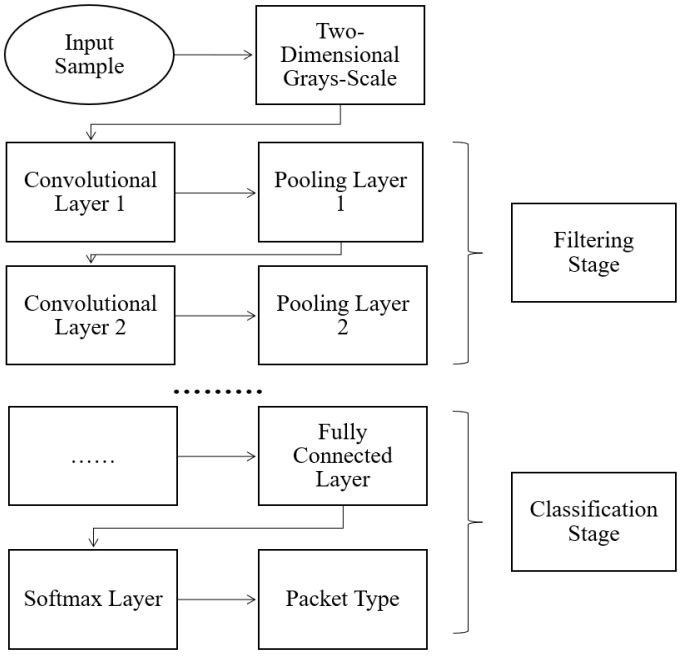

Figure 1. Overall algorithm flow.

2.1. Dataset Description

The dataset CIC-IDS2017 is from the CSE and CIC [7]. This dataset contains the latest common attacks including violent FTP, violent SSH, DOS, HeartBleed, web attacks, infiltration, botnets, and DDoS. They were executed in the morning and afternoon on Tuesday, Wednesday, Thursday, and Friday. We use this dataset to train and evaluate the performance of our network. The data type is shown in Table I, and there are around 80 characteristics, including the information of the data packet.

Table 1. Dataset distribution of cic-ids2017

Day	***Description***	***Size***
Monday	Normal	11GB
Tuesday	Force + SSH + SFTP + Normal	11GB
Wednesday	Dos+ Heartbleed + Normal	13G
Thursday	XSS+ Web Attack + Normal	7.8G
Friday	Botnet + DDos + PortScan + Normal	8.3G

2.2. Data Preprocessing

We convert the pcap-style network stream data into a two-dimensional grayscale image, which is computable for a convolutional nerve network. Specifically, we first normalize the data and converse symbolic. In this paper, we use Session as the basic research unit, which consists of a package of data for two-party communication.

The data normalization phase starts by slicing the data provided by the training dataset into packets according to the source IP and the destination IP and forms a time-series session set. The pcap file is divided into q packets, and the first x bytes of each packet are taken.

In the symbol data conversion phase, take the CIC-IDS2017 dataset as an example, its network traffic flow data has been extracted, and more than 80 features such as Source Port, and normal or attack type tags have been generated. By using One-Hot Encoding, each set of feature values is expressed as a y-dimensional vector, after which we transform the features into $q\ x \times y$ sized two-dimensional images.

2.3. Model Construction

1) Convolutional Neural Networks: In a traditional convolutional neural network [8], the results are usually flattened by pooling (typically max pooling), and fed into two and more completely connected layers. Then using the softmax function to compute the probability values of the samples concerning each class. Then we select the maximum probability value as the result of the classification. However, the problem of multiple parameters occurs with multiple fully connected layers. For example, if the feature vector is fed into the network model, after the last convolution, a $W \times H \times C$ feature map is obtained. Setting the number of fully connected layer neurons M, then the fully connected layer requires a total of $W \times H \times C \times M$ parameters. This makes the whole network with many parameters, slow computation, complicated parameter updating, inefficient, and even overfitting.

To address this issue, we use the strategy from the new deep network structure "Network in Network" (NIN) [9]. The classification results are obtained by global average pooling after convolution. After multiple convolutions, we finally obtain the feature map ($W \times H \times Label$) with the same number of categories as the number of samples obtained, where the label represents the number of sample categories. The mean value of each feature map is obtained as a matrix of $1 \times 1 \times Label$. The classification result of the sample data is obtained after transforming the matrix into a one-dimensional vector. The purpose of global pooling is to match the size between the pooling sliding window and one of the feature maps, which is to let a feature map only output one value.

The benefit of worldwide pooling over a completely associated layer is that there is no parameter setting, just the mean or most extreme value of the entire feature map is required, to significantly decrease the training time. In addition, the global pooling can total spatial data, so it has better heartiness to the spatial change of the information.

2) Model construction: We use convolutional layers with small convolution kernels to extract local features of the input vector, which can be used to pool layer 1 for accurate feature results. In convolutional layer 2, a large convolution kernel is used to compare the connections between two distant parts., such as the information about the valid load in the data package.

To be more specific, in the convolutional layer, different convolutional kernels are used to slide over the feature map of the previous layer. Then the nonlinear mapping is computed by the activation function to obtain the characteristic matrix of the previous layer, which is defined as:

$$Z^{l+1}(i,j) = \sum_{k=1}^{c}\sum_{x=1}^{f}\sum_{y=1}^{f}\left[Z_k^l(s*i+x, s*j+y)*\omega_k^{l+1}(x,y)\right]+b, \quad (1)$$

$$A_{i,j,k}^1 = f(Z_{i,j,k}^1). \quad (2)$$

In (1), $\omega_k^{l+1}(x,y)$ is the feature map of layer i and $\mathrm{Z}^{l+1}(\mathrm{i},\mathrm{j})$ is the feature map of layer j. * is convolution operation, f is convolutional kernel length, s is convolution stride and b is offset.

(2) is the activation function, where k is the channel number of the feature map. A is the output vector of Z vector through the activation function.

Then the convolution results are sent to the global average pooling and global maximum pooling respectively, and the two results are merged and reconstructed to obtain $1 \times k$ feature maps, with k means the number of feature maps. After global pooling, the results are merged and flattened, and the classification results are obtained by using the fully connected layer combined with the softmax function. The probability value of the flow in each category is output, and the category with the highest probability is the predicted category of the flow. Further, we compare the predicted value and the true value to calculate the loss value, and the Adam optimization algorithm is used to update the weights and biases iteratively until convergence. A training model is derived. Then, we can use the trained model for malicious traffic detection.

3. Experimental settings

In this experiment, we compare our model with four machine learning baseline models to verify the effectiveness of our model with accuracy (ACC), detection rate (DR), false alarm rate (FAR), and classifier accuracy score (F1-score). The four machine learning models which we choose are Random Forest, SVM, KNN, and DBN. We show the four baselines (Sec. A) and evaluation metrics (Sec. B) in the following.

3.1. Baselines

3.1.1. Random Forest

Random Forest is a calculation that coordinates numerous trees through the possibility of group learning, and its fundamental unit is a decision tree. Each tree is a classifier, so for an info test, there are as numerous grouping results as there are trees. At the point when we play out the order task, every choice tree in the backwoods passes judgment on the information tests and arranges them independently, every choice tree will get its grouping result, and the irregular timberland coordinates all the characterization casting ballot results, determining that the classification is the greater part vote as the last wanted outcome. [10]

3.1.2. SVM

The data is dimensionally reduced by the method of Principal Component Analysis (PCA), and then the features are classified into two categories by the method of Support Vector Machines (SVM). This algorithm is commonly used for face recognition functions. [11]

3.1.3. KNN

K-Nearest Neighbor (KNN) [12] can be understood as if there are N nearest neighbors (nearest neighbors in the feature space) in the feature space of a sample, and most of these neighbors belong to a certain category, then The algorithm will think that the sample also belongs to this category. This method only determines the category of the sample to be divided according to the category of one or several recent samples and is often used in character recognition, image recognition, and other fields.

3.1.4. DBN

Deep Belief Network (DBN) is a hybrid generative model consisting of Restricted Boltzmann Machine (RBM) and Sigmoid Belief Network (SBN). It belongs to an unsupervised learning method, which aims to fit the training data as much as possible. The "concatenation" of several RBMs forms a DBN.

3.2. Evaluation Metrics

The evaluation metrics used in this paper contain ACC, DR, FAR, and F1-score. The evaluation metrics are defined as:

$$ACC = \frac{TP+TN}{TP+FP+FN+TN}, \tag{3}$$

$$DR = \frac{TP}{TP+FN}, \tag{4}$$

$$FAR = \frac{FP}{FP+TN}, \tag{5}$$

$$F1-score = \frac{2\times\frac{TP}{TP+FP}\times\frac{TP}{TP+FN}}{\frac{TP}{TP+FP}+\frac{TP}{TP+FN}} \times 100\%. \tag{6}$$

Among them, TP addresses the quantity of accurately arranged target tests, TN is the quantity of other accurately ordered examples, FP is the number of misidentified target tests, and FN is the quantity of discarded target tests to be recognized.

4. Results and Discussion

The training process is shown in Fig. 2 and 3 and the final accuracy of this method on the training set is 99.8%.

Table 2. Horizontal comparison of machine learning algorithms

ALGORITHMS	*ACC*	*DR*	*FAR*	*F1-score*
Rep+Random Forest	0.967	0.94	0.01	/
PCA+SVM	0.929	0.96	0.05	/
KNN	0.985	0.96	0.02	0.96
SVM+DBN	0.926	0.97	/	0.97
CNN-1D	0.998	0.98	0.01	0.93

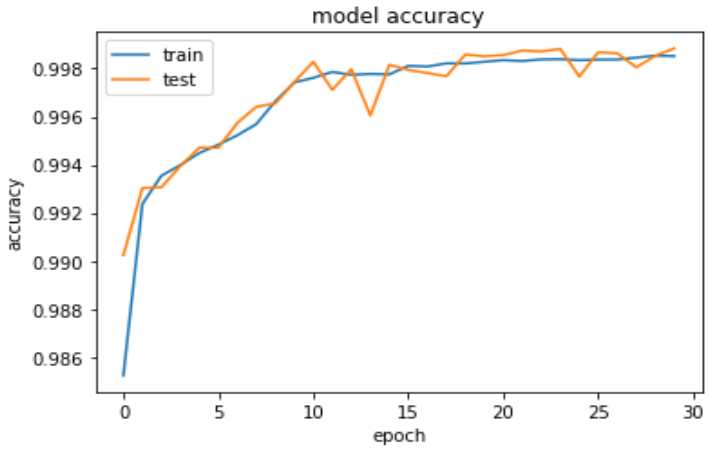

Figure 2. Model accuracy of the training set and testing set.

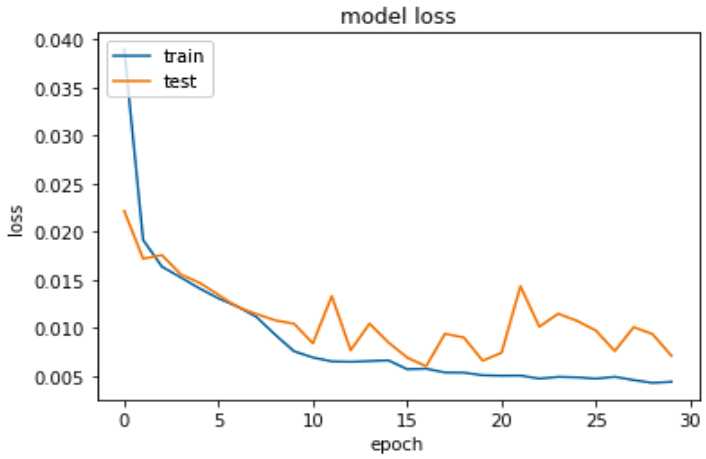

Figure 3. Model loss of training set and testing set.

On the same CIC-IDS2017 dataset, five kinds of machine learning algorithms such as Random Forest, SVM, KNN, and DBN are selected and compared. The experimental results in Table II show that the ICNN-1D algorithm proposed in this paper is superior to other algorithms in accuracy, detection rate, and false alarm rate.

5. Conclusion

Deep learning can automatically abstract high-level features from raw data, has high adaptability to massive high-dimensional data and is helpful to solve feature engineering problems in traditional machine learning. Therefore, this paper applies ICNN-1D to detect anomalous traffic in order to improve the performance of Intrusion Detection. Compared with other traditional machine learning models, ICNN-1D can realize network intrusion detection while maintaining high model accuracy and low data loss. It is of great significance to discover abnormal data in network packets in time and prevent the disclosure of personal information of network users.

In the future, we will focus on the following work. Firstly, we will improve the detection accuracy and the generalization ability of the method, which may require us to further modify the model to achieve the goal. Secondly, since detection time is also the key to anomalous traffic detection, it should be emphasized that the method needs to meet the detection time requirement while improving detection accuracy.

References

[1] Information from: https://blog.checkpoint.com/2022/01/10/check-point-research-cyber-attacks-increased-50-year-over-year/

[2] Chen, S.W., “Research of DDoS Attacks Detection Methods Based on Spectrum Analysis and Statistical Machine Learning,” Information Engineering University, (2013).

[3] Rao, L., “Research of Network Attacks Detection Based on Support Vector Machine,” Nanjing University of Science and Technology, (2007).

[4] Münz, Gerhard, Sa Li, and Georg Carle. “Traffic anomaly detection using k-means clustering.” GI/ITG Workshop MMBnet. Vol. 7, (2007).

[5] Lima, Moisés F., et al. “Anomaly detection using baseline and k-means clustering.” SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks. IEEE, (2010).

[6] Zhu, Qiqi, and Li Sun. “Big data-driven anomaly detection for cellular networks.” IEEE Access 8, 31398-31408(2020).

[7] Information from: http://205.174.165.80/CICDataset/CIC-IDS-2017/Dataset/

[8] Wu, Jianxin. “Introduction to convolutional neural networks.” National Key Lab for Novel Software Technology. Nanjing University. China 5.23 (2017): 495.

[9] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

[10] Belgiu M, Drăguţ L. "Random forest in remote sensing: A review of applications and future directions." ISPRS journal of photogrammetry and remote sensing, 114: 24-31 (2016).

[11] Yu H, Chen R, Zhang G. "A SVM stock selection model within PCA[J]. Procedia computer science", 31: 406-412 (2014).

[12] Guo, Gongde, et al. "KNN model-based approach in classification. "OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, Berlin, Heidelberg, (2003).