An Intrusion Detection Model Based on SMOTE and Convolutional Neural Network Ensemble

Luyao Tian¹,² and Yueming Lu¹,²,*

¹ Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China
² The School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China

*Email: tianluyao@bupt.edu.cn, ymlu@bupt.edu.cn

Abstract. Massive, multi-dimensional and imbalanced network traffic data has brought new challenges to traditional intrusion detection systems (IDSs). The detection performance of traditional algorithms is closely related to feature extractions, which are not effective in the massive and imbalanced data environments. In this paper, we propose an intrusion detection model based on synthetic minority oversampling technology (SMOTE) and convolutional neural network (CNN) ensemble. It converts original traffic vectors into images, designs a CNN structure, and combines SMOTE and CNN ensemble to solve the problem of imbalanced datasets. Using the standard KDD CUP 99 dataset to evaluate the performance of the proposed model and analysing the contribution of features to model decision-making show that the model’s F1 score are better than traditional algorithms in the classes with few samples and the model improves the efficiency of network intrusion detection.

1. Introduction

The rapid development of the Internet brings people convenience but also security issues. It is a key issue needed to be solved urgently that how to detect various network intrusions, especially new ones. Intrusion detection system (IDS) provides protection for network information security by detecting potential intrusions, and plays an increasingly important role in network security [1]. The traditional intrusion detection method is to extract features from intrusion behaviours manually, establish a feature database, and complete the detection by matching the features in the database [2-3], or using machine learning classification algorithms. This method has a strong dependence on feature extraction. Extracting distinguishable features is vital for traditional intrusion detection. However, extracting features through expert knowledge and manual means is time-consuming, and cannot extract high-order features. Convolutional neural network (CNN), a classic neural network model, can automatically combine features and obtain complex high-order features, which is suitable for large-scale network intrusion detection. To ensure the generalization ability of the model, a large number of samples are needed to train the model. When the samples are not balanced, the detection performance for categories with few samples is not satisfactory. To solve this problem, an intrusion detection model based on synthetic minority oversampling technology (SMOTE) and CNN ensemble (SMOTE-ENSEMBEL-CNN-IDS) is proposed, which can provide better detection and classification performance.
2. Related Work

In 2010, Sommer et al. proposed to introduce machine learning into the field of intrusion detection, aiming to strengthen the IDSs [4]. Brao and Swathi proposed to search for K nearest neighbours using indexed partial distance to implement intrusion detection [5]. Experiments show that this method has high speed and accuracy, but the precision and recall are not considered. Lin et al. proposed an intrusion detection algorithm based on principal component analysis (PCA) and random forests, which uses PCA to reduce the dimension of data, and then uses random forests to classify the data [6]. The results show that this method can improve the detection performance. Sun et al. introduced a composite weighted Naive Bayes algorithm, which can effectively improve the detection rate [7].

Compared with classical machine learning algorithms, deep learning algorithms have obvious advantages: strong automatic learning ability, good at fitting high-dimensional nonlinear data, and suitable for incremental learning. Therefore, the application of deep learning in intrusion detection has been a research direction at home and abroad. Nadeem et al. proposed to use a ladder network combining neural networks and semi-supervised learning for intrusion detection [8]. It can achieve similar accuracy to the supervised learning method while only requiring a small number of labelled samples. But other evaluation indicators are not considered. Zhao et al. proposed an intrusion detection method based on deep belief networks (DBNs) and probabilistic neural networks [9]. Cai improved the MDBest algorithm and proposed the MDBest2 algorithm, which was combined with DBN for intrusion detection [10].

Compared with other deep learning algorithms, CNN greatly reduces parameters by weight sharing and sampling, which is not easy to overfit, and has strong generalization ability. There have been studies that applied CNN to intrusion detection. Kolosnjaji et al. combined convolutional structure and feedforward neural network to simulate the execution sequence of disassembling malicious binary files [11]. The results show that this method is better than feedforward neural network and support vector machine (SVM). Wang et al. proposed an end-to-end encryption service classification method based on one-dimensional CNN [12]. This method integrates feature extraction, feature selection and classification into a unified end-to-end framework, and automatically learns the nonlinear relationship between the original input and the expected output. Yu et al. proposed a deep learning method based on dilated convolutional autoencoders for network intrusion detection [13]. This method combines the advantages of stacked autoencoders and CNNs, which can automatically learn features from unlabelled network traffic data. Xiao et al. proposed an intrusion detection model based on feature reduction and CNN, with an overall accuracy of 94.0%, but the detection rates in the U2R and R2L are only 20.61% and 18.96% [14]. The reason is that the data distribution is imbalanced. This is a common problem of intrusion data. For example, the amount of data in normal traffic is much larger than that of abnormal traffic, and the amount of data in DoS is much larger than other attacks. The uneven number of samples in different classes will cause the model to learn inadequately from classes with small sample size, and tend to classify samples into classes with large sample size.

In order to perform intrusion detection with imbalanced datasets, researchers have proposed various solutions. Vinayakumar et al. added records to classes with small sample size, and randomly deleted records in classes with large sample size [15]. Wu et al. improved the calculation of the cost function of CNN to solve the problem caused by imbalanced data [16]. The weight of each class in the cost function was set according to the number of samples. Dong et al. compared the detection performance of SVM-RBM model with naive Bayes, decision tree and SVM [17]. In order to solve the problem, oversampling technology was used. The accuracy can be increased by 13.09%. A large reduction in the amount of data will reduce the feature learning ability of CNN. Analyzing the above solutions, this paper proposes an intrusion detection model based on SMOTE and CNN ensemble, which can improve the detection performance of classes with few samples.

3. Intrusion Detection Model Based on CNN

CNN has fewer parameters than other deep learning algorithms and has strong generalization capabilities. CNN greatly reduces parameters through weight sharing and pooling, reduces the
difficulty of training, and avoids overfitting caused by too many parameters. Therefore, using CNN as
the base model, we propose SMOTE-ENSEMBEL-CNN-IDS for intrusion detection with imbalanced
data. This section mainly describes the workflow and structure settings of the basic model—an
intrusion detection model based on CNN (CNN-IDS). The workflow of CNN-IDS is shown in Figure
1. First, the original data needs to be processed, including unifying data format, normalizing data and
converting data into images. Then, the processed dataset is split into a training set and a test set; finally,
the CNN-IDS model is trained, tested, and evaluated.

![Figure 1. The workflow of CNN-IDS.](image)

### 3.1. Introduction to the Dataset
The KDD CUP 99 [18] is derived from the DARPA intrusion detection evaluation project carried out
by the Lincoln Laboratory of Massachusetts Institute of Technology in 1998. It is a recognized dataset
in network intrusion detection. Many studies use it as the experimental basis. In this paper, 10%
subsets of the data are used as the training set and the test set, which contain 4 major types of attacks:
DoS, R2L, U2R and Probe. The distribution of 10% subsets is shown in Table 1.

| Label  | Train set (10%) | Test set(corrected) |
|--------|----------------|---------------------|
| Normal | 97278          | 60593               |
| DoS    | 391458         | 229853              |
| Probe  | 4107           | 4166                |
| R2L    | 1126           | 16189               |
| U2R    | 52             | 228                 |

### 3.2. Data Processing
Each record in the dataset is described by 41 features, including three symbolic features. To facilitate
training and calculation, one-hot encoding is used to convert them into numeric types. For example, 3
categories of the protocol can be encoded as [1,0,0], [0,1,0], [0,0,1]. After the conversion, each record
is described by 122 features.

Analysing the data can find that there are dimensional differences between different features. In
order to prevent the excessive dimensional differences from affecting training and testing, the data
must be normalized. The maximum-minimum normalization is used to map the data to the [0,1]
interval without destroying the linear relationship between the original data. The calculation formula of maximum-minimum normalization is as follows:

\[ x'_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \]  

(1)

where \( x_i \) represents the \( i \)-th value of the feature, \( x'_i \) represents the result of normalization, \( x_{\min} \) presents the smallest value of the feature, and \( x_{\max} \) presents the largest value of the feature.

Researches show that compared with one-dimensional input data CNN is better at learning from images [16]. Records in the KDD CUP 99 dataset are one-dimensional, which need to be converted into two-dimensional images. The following method is adopted: After one-hot encoding and normalization, each record is described by 122 features, filtered out a low-variance feature, and then 121 features are converted into an 11*11 image, which is used as input to the CNN-IDS.

3.3. CNN Structure

CNN is mainly composed of five basic layers: input layer, convolution layer, pooling layer, fully connected layer and output layer. The structure settings of the CNN-IDS are shown in Table 2. The first layer is an input layer, and its input is 11*11 images. The second and fourth layers are convolutional layers, which complete the convolution operation on images. ReLU function is used as activation function, and batch normalization (BN) is used to speed up the training process. The third and fifth layers are pooling layers, which use max-pooling to sample the images. The sixth layer is a fully connected layer. To prevent overfitting and improve the generalization ability, dropout is used, and the keep_prob is 0.7. The seventh layer is an output layer, using Softmax function as the classifier.

| Layer       | Filter size | Stride | Activation function | Output size |
|-------------|-------------|--------|---------------------|-------------|
| input       | -           | -      | -                   | 11*11*1     |
| convolution | 2*2         | 1      | ReLU                | 11*11*8     |
| pooling     | 2*2         | 1      | -                   | 10*10*8     |
| convolution | 2*2         | 1      | ReLU                | 10*10*16    |
| pooling     | 2*2         | 1      | -                   | 9*9*16      |
| fully connected | -          | -      | ReLU                | 64          |
| output      | -           | -      | Softmax             | 5           |

4. Intrusion Detection Model Based on SMOTE and CNN Ensemble

The experimental results in Section 5.2 show that the trained CNN-IDS would bias towards the class with a large number of samples, and the detection performance on the class with few samples is poor. It is caused by the imbalanced distribution among the classes, where DoS accounts for 79.24% of the training set, while U2R only accounts for 0.01%.

Methods to solve the problem can be roughly divided into two categories: at the data level, the distribution of data can be balanced by sampling, and at the algorithm level, methods such as ensemble learning and cost-sensitive learning could be used. Combining SMOTE and ensemble learning, this paper proposes SMOTE-ENSEMBEL-CNN-IDS to solve the problem. At the data level, SMOTE is used to balance the data distribution; at the algorithm level, based on ensemble learning, multiple CNN-IDSs are trained and integrated. The workflow of the model is shown in Figure 2.

4.1. Data Level
At the data level, sampling algorithms can be divided into three types: under-sampling, over-sampling methods, and hybrid sampling that use the former two at the same time. The under-sampling algorithm balances the data distribution by reducing the amount of data in classes with big numbers, but this method may cause the loss of some important information. The oversampling algorithm adjusts the distribution by increasing the amount of data in classes with small numbers. Common oversampling methods include random oversampling and SMOTE [19]. The random oversampling increases the amount of data by randomly copying the data of classes with small numbers to balance the data distribution, but the generated new data has a strong similarity to the original one, the problem of overfitting is prone to occur. SMOTE can solve this problem. It finds K nearest neighbours for a certain sample in the class with small numbers, randomly selects M neighbours and performs linear interpolation between the original sample and the selected neighbours to obtain M new samples that do not exist in the original dataset. This paper uses SMOTE to balance the data distribution, and the follow-up work will be performed on the balanced dataset.

![Figure 2. The workflow of SMOTE-ENSEMBEL-CNN-IDS.](image)

### 4.2 Algorithmic Level

Ensemble learning is to train multiple base models, and integrate the results of each model through a certain strategy to improve the performance. At the algorithm level, this paper uses the idea of bagging, one of the ensemble algorithms, to improve the detection performance.

The full name of bagging is bootstrap aggregating. It uses Bootstrap sampling to sample the training set with replacement multiple times to obtain multiple subsets, and train the base model on each subset in parallel. Each model will give a prediction result for the samples in the test set. For classification problems, voting is often used to integrate the results, and for regression problems, averaging is generally used.

But sampling with replacement will cause some samples to be sampled repeatedly, and some never to be sampled as training data. This may lose important information and is not conducive to the model's learning from classes with few samples. In this paper, we use sampling without replacement to form multiple datasets, ensuring that every sample would be added into the training set. The specific steps are as follows. The original data is sampled without replacement to obtain a data subset, and its complement is used as a training set. Repeat this step N times to ensure that all the samples in the original data is sampled, and N training sets are obtained. Train N CNN-IDS models in parallel, and integrate the results by voting to get the final result.
5. Experiments and Results

5.1. Evaluation Indicators
In this paper, AC (accuracy) and F1 (F1 score) are selected as indicators. F1 reflects the model's detection performance on the imbalanced dataset, and the variation of AC is used to measure the importance of features. The formulas for AC and F1 are as follows.

\[ AC = \frac{TP}{TP + FP + TN + FN} \]  
\[ F1 = \frac{P \cdot R}{P + R} \]

where TP (True Positive) indicates that an attack is classified as an attack, TN (True Negative) indicates that the normal traffic is classified as normal, FN (False Negative) indicates that the attack is classified as normal, and the FP (False Positive) indicates that the normal traffic is classified as an attack.

5.2. Experimental Results and Analysis
The performance of the CNN-IDS is evaluated first, and its confusion matrix is shown in Table 3. It can be calculated that the overall AC is 92.45%, but the F1 are only 11.10 % and 20.90% on R2L and U2R, which is due to the imbalanced distribution of the dataset. The data amount of R2L and U2R is too small, and the model learns insufficiently from those two classes.

Table 3. Confusion matrix of the CNN-IDS.

| Predicted class | Normal | DoS | Probe | R2L | U2R |
|-----------------|-------|-----|-------|-----|-----|
| Actual class    |       |     |       |     |
| Normal          | 59481 | 853 | 247   | 6   | 6   |
| DoS             | 5952  | 223814 | 87   | 0   | 0   |
| Probe           | 197   | 660 | 3286  | 23  | 0   |
| R2L             | 15180 | 11  | 39    | 953 | 6   |
| U2R             | 71    | 4   | 118   | 7   | 28  |

The confusion matrix of SMOTE-ENSEMBEL-CNN-IDS is shown in Table 4. Experimental results show that the overall AC is 92.72%, which has a certain improvement compared to CNN-IDS. The F1 comparison of the two models is shown in Figure 3. Compared with CNN-IDS the F1 of SMOTE-ENSEMBEL-CNN-IDS in five classes have improved, especially in classes with few samples such as R2L and U2R.

Table 4. Confusion matrix of SMOTE-ENSEMBEL-CNN-IDS.

| Predicted class | Normal | DoS | Probe | R2L | U2R |
|-----------------|-------|-----|-------|-----|-----|
| Actual class    |       |     |       |     |
| Normal          | 59485 | 778 | 247   | 63  | 20  |
| DoS             | 5695  | 223871 | 248  | 39  | 0   |
| Probe           | 433   | 193 | 3530  | 10  | 0   |
| R2L             | 14647 | 1   | 12    | 1483| 46  |
| U2R             | 37    | 0   | 139   | 13  | 39  |
Figure 3. The F1 comparison of models proposed in this paper.

In addition, use machine learning algorithms, such as K nearest neighbor (KNN), naive Bayes, random forests, and deep learning algorithm based on LeNet-5 [20] to implement network intrusion detection model. Compare each model’s detection performance in classes with few samples, the result is shown in Figure 4. It can be seen that the F1 of SMOTE-ENSEMBEL-CNN-IDS is higher than other models in R2L and U2R.

CNNs can learn features automatically, but due to the black box characteristics of neural networks, the process of feature extraction cannot be explicitly observed. We are concerned about which features can play an important role in the classification of network intrusions, so the following experiment is carried out to analyze the contribution of input features to the classification. The analysis of feature contribution is based on the prediction of out-of-bag data, and the specific steps are: 1) Preparing out-of-bag data, that is, data that is not involved in the training process, using the trained model to predict its class and calculating AC; 2) Adding noise to each feature of the out-of-bag data, calculating AC again, and using the variation of AC to describe the contribution of the feature. The greater the variation, the greater the contribution. Table 5 lists the top five features in terms of AC variation, that is, the five most contributing features.

Table 5. The five most contributing features.

| Feature                        | AC variation |
|--------------------------------|--------------|
| dst_host_rerror_rate.          | 10.66%       |
| service. = eco_i               | 4.05%        |
| count.                         | 2.87%        |
| dst_host_same_src_port_rate.   | 2.33%        |
| diff_srv_rate.                 | 0.67%        |

where dst_host_rerror_rate. represents the percentage of connections with REJ errors in the first 100 connections that have the same target host as the current connection; service. = eco_i indicates whether the network service type of the target host is eco_i; count. represents the number of connections with the same target host as the current connection in the past two seconds; dst_host_same_src_port_rate. represents the percentage of connections that have the same target host and the same source port as the current connection among the first 100 connections; diff_srv_rate. represents the percentage of connections that have the same target host as the current connection in the past two seconds.
6. Conclusions
Aiming at the issues caused by uneven data distribution, this paper proposes an intrusion detection model based on SMOTE and CNN ensemble. At the data level, SMOTE is used for oversampling to balance the data distribution; at the algorithm level, based on the idea of Bagging, multiple CNN-IDSs are trained and integrated by voting. The experimental results show that SMOTE-ENSEMBEL-CNN-IDS can significantly improve the F1 of classes with few samples (R2L, U2R) while ensuring the overall AC. In addition, according to the analysis of the contribution of each feature to the model decision based on the out-of-bag data, it is found that dst_host_rerror_rate, service = eco_i and count have larger contribution, which play an important role in the classification of network intrusions.

References
[1] Anderson J P 1980 Computer security threat monitoring and surveillance James P. Anderson Co. Report.
[2] Lindqvist U and Porras P A 2002 Detecting computer and network misuse through the Production-Based Expert System Toolset (P-BEST) IEEE Symp. on Security and Privacy (California, USA, 12-15 May. 2002) pp 146–61
[3] Porras P and Kemmerer R 1992 State transition analysis: A rule-based intrusion detection approach IEEE Trans. on Software Engineering 21 220–9
[4] Sommer R and Paxson V 2010 Outside the closed world: On using machine learning for network intrusion detection IEEE Symp. on Security and Privacy (California, USA, 16-19 May. 2010) pp 305-16
[5] Brao B and Swathi K 2017 Fast kNN classifiers for network intrusion detection system Indian J. Sci. Technol. 10 1–10
[6] Lin W, Chen M, Zhan Y and Liu C 2017 Research on an intrusion detection algorithm based on PCA and random-forest classification Netinfo Security 17 50–4
[7] Sun C, Xing J, Yang Q and Han D 2017 Intrusion detection methods based on improved Naive Bayesian Microcomputer & Its Appl. 1 8-10,14
[8] Nadeem M, Marshall O, Singh S, Fang X and Yuan X 2016 Semi-supervised deep neural network for network intrusion detection Proc. KSU Conf. Cybersecur. Educ. Res. Pract. (Georgia, USA, Oct. 2016) pp 1–13
[9] Zhao G, Zhang C and Zheng L 2017 Intrusion detection using deep belief network and probabilistic neural network IEEE Int. Conf. on Computational Science and Engineering and IEEE Int. Conf. on Embedded and Ubiquitous Computing (Guangzhou, China, 21-24 Jul. 2017) pp 639-42
[10] Cai Z 2016 DBN and MDBBoost2 Used in Intrusion Detection (Guangdong, China: Guangdong University of Technology)
[11] Kolosnjaji B, Eraisha G, Webster G, Zarras A and Eckert C 2017 Empowering convolutional networks for malware classification and analysis Int. Joint Conf. on Neural Networks (Alaska, USA, 14-19 May. 2017) pp 3838-45
[12] Wang W, Zhu M, Wang J, Zeng X and Yang Z 2017 End-to-end encrypted traffic classification with one-dimensional convolution neural networks IEEE Int. Conf. on Intelligence and Security Informatics (Beijing, China, 22-24 Jul. 2017) pp 43-8
[13] Yu Y, Long J, and Cai Z 2017 Network intrusion detection through stacking dilated convolutional autoencoders Secur. Commun. Netw. 2 1–10
[14] Xiao Y, Xing C, Zhang T and Zhao Z 2019 An intrusion detection model based on feature reduction and convolutional neural networks IEEE Access 7 42210-9
[15] Vinayakumar R, Soman K P and Poornachandran P 2017 Applying convolutional neural network for network intrusion detection Int. Conf. on Advances in Computing, Communications and Informatics (Udupi, India, 13-16 Sept. 2017) pp 1222-8
[16] Wu K, Chen Z and Li W 2018 A novel intrusion detection model for a massive network using convolutional neural networks IEEE Access 6 50850-9
[17] Dong B and Wang X 2016 Comparison deep learning method to traditional methods using for network intrusion detection 8th IEEE Int. Conf. on Communication Software and Networks (Beijing, China, 4-6 Jun. 2016) pp 581-5

[18] KDD Cup 1999 http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[19] Chawla N V, Bowyer K W, Hall L O and Kegelmeyer W P 2011 SMOTE: synthetic minority over-sampling technique J. Artificial Intelligence Research 16 321-57.

[20] Wu T 2018 Research and Application of Convolution Neural Network in Intrusion Detection System (Guangdong, China: South China University of Technology)