



Article

CSK-CNN: Network Intrusion Detection Model Based on Two-Layer Convolution Neural Network for Handling Imbalanced Dataset

Jiaming Song ¹, Xiaojuan Wang ^{2,*}, Mingshu He ²  and Lei Jin ³ 

¹ Institute of Cloud Computing and Big Data, China Academy of Information and Communications Technology, Beijing 100191, China

² School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

³ School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: wj2718@bupt.edu.cn

Abstract: In computer networks, Network Intrusion Detection System (NIDS) plays a very important role in identifying intrusion behaviors. NIDS can identify abnormal behaviors by analyzing network traffic. However, the performance of classifier is not very good in identifying abnormal traffic for minority classes. In order to improve the detection rate on class imbalanced dataset, we propose a network intrusion detection model based on two-layer CNN and Cluster-SMOTE + K-means algorithm (CSK-CNN) to process imbalanced dataset. CSK combines the cluster based Synthetic Minority Over Sampling Technique (Cluster-SMOTE) and K-means based under sampling algorithm. Through the two-layer network, abnormal traffic can not only be identified, but also be classified into specific attack types. This paper has been verified on UNSW-NB15 dataset and CICIDS2017 dataset, and the performance of the proposed model has been evaluated using such indicators as accuracy, recall, precision, F1-score, ROC curve, AUC value, training time and testing time. The experiment shows that the proposed CSK-CNN in this paper is obviously superior to other comparison algorithms in terms of network intrusion detection performance, and is suitable for deployment in the real network environment.

Keywords: network intrusion detection; class imbalance; convolutional neural network; Cluster-SMOTE



Citation: Song, J.; Wang, X.; He, M.; Jin, L. CSK-CNN: Network Intrusion Detection Model Based on Two-Layer Convolution Neural Network for Handling Imbalanced Dataset. *Information* **2023**, *14*, 130. <https://doi.org/10.3390/info14020130>

Academic Editors: Amjad Gawanmeh and Vishal Kumar

Received: 27 December 2022

Revised: 2 February 2023

Accepted: 13 February 2023

Published: 16 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of computer network, human beings increasingly rely on the network to process information such as work, study and life. Network security has become particularly important [1,2]. As a proactive security protection technology, intrusion detection has attracted more and more scholars' attention and research. Honey pot blocks the attacker from the network with knowing the information analysis to block the intruder [3]. According to the different sources of analysis data, intrusion detection system (IDS) can be divided into host-based IDS (HIDS) and network-based IDS (NIDS).

HIDS mainly protects the host by monitoring logs and system calls, while NIDS protects network devices by analyzing the communications that occur on network devices. At present, NIDS is the most widely used, mainly including rule-based misuse detection (MIDS) and statistics-based anomaly detection (AIDS). The former identifies abnormal behavior by matching existing attack rules. This method can accurately identify known attacks, but cannot detect new network attacks. The latter can identify by detecting the characteristics of network flow or the distribution deviating from normal behavior, which is helpful to identify unknown intrusions. The proposed CSK-CNN in this paper using two-layer CNN and Cluster-SMOTE + K-means to process imbalanced data (CSK-CNN) to realize network intrusion detection is an anomaly detection algorithm based on statistics.

Machine learning algorithms are widely used in statistical network anomaly detection. Machine learning algorithms, such as support vector machine (SVM) [4], K-means [5], XGBoost [6], random forest (RF) [7], distinguish between normal and abnormal network behaviors through feature engineering. However, with the increase of network anomaly intrusion types and data volume, traditional machine learning algorithms, as shallow learning methods, are difficult to capture important information, have weak generalization ability, and are not suitable for network intrusion detection with large amounts of data.

In recent years, deep learning algorithms that can fully mine and extract potential features between data have attracted attention. Deep learning models, including Convolutional neural networks (CNN) [8], Recurrent neural networks (RNN) [9], and Long short term memory (LSTM) [10], have been applied to network intrusion detection, and experiments show that they have good performance on large datasets.

However, network intrusion detection still has some problems. For example, it generally performs well in distinguishing between normal and abnormal network behaviors, but it does not perform well in detecting specific attack types. In addition, the performance in the classification of imbalanced datasets is not good, and the detection rate drops significantly on small type datasets. Therefore, this paper focuses on solving the multi classification problem of imbalanced datasets in large-scale network intrusion detection.

The datasets with obviously uneven distribution of different classes of samples are called imbalanced datasets. Among them, the class with a large number of samples is called majority class, on the contrary called minority class. In the real network world, due to the uneven distribution of normal samples and abnormal samples, the classification of network traffic is essentially an imbalanced classification problem. At present, there are four main methods to deal with imbalanced datasets, including data core method, algorithm core method, cost sensitive method and integration method. In this paper, we use data core method to solve the problem of imbalanced data in network intrusion detection. This method is realized by adding or reducing datasets of different categories in imbalanced data.

The main contributions of this paper are summarized as follows:

- (1) This paper proposes a network intrusion detection model CSK-CNN, which combines the imbalance processing algorithm Cluster-SMOTE + K-means and two-layer CNN algorithm, and has a high detection rate in identifying imbalanced datasets. CSK-CNN is an anomaly based network intrusion detection model, which uses two-layer CNN to identify and classify network intrusion behaviors: Layer 1 uses binary classification to identify normal traffic and abnormal traffic. Layer 2 uses multiple classification to classify abnormal traffic into specific attack categories.
- (2) In this paper, we propose a novel method, CSK algorithm, to deal with class imbalanced datasets on large datasets. This method first uses Cluster-SMOTE to oversample the training samples for minority classes, and then uses K-means to under sample the training samples for majority classes, finally making the training sample classes balanced. This method can not only avoid a large amount of time and space waste caused by over sampling, as well as over fitting, but also avoid the loss of important sample information caused by random under sampling. Experiments show that the anomaly detection rate is significantly improved in minority classes.
- (3) This paper uses accuracy, recall, precision, F1 score, ROC curve, AUC value, training time and testing time to evaluate the proposed CSK-CNN model, and compares the performance of four imbalanced class processing algorithms (SMOTE, ROS, ADASYN, RUS + SMOTE, K-means + SMOTE) and two machine learning classification algorithms (RF and MLP). The experimental results show that the CSK-CNN model proposed in this paper is effective in dealing with large-scale imbalanced network intrusion detection, and its performance is better than other algorithms. Therefore, CSK-CNN, the accurate and efficient network anomaly intrusion detection method proposed in this paper, can be deployed in the real world network environment.

The rest of the article is arranged as follows: The second part mainly introduces the related work of neural network algorithm and class imbalance dataset algorithm in the field of network intrusion detection. The third part introduces the CSK-CNN algorithm and preprocessing method proposed in this paper. The fourth part introduces the experimental process and discusses the experimental results. Finally, the fifth part summarizes the article.

2. Related Work

Since 2000, machine learning algorithms have been widely used in network intrusion detection. Koroniotis et al. [11] proposed the role of machine learning algorithm in network forensics mechanism based on network flow identifier, which can track suspicious activities of botnets. The experiment on UNSW-NB15 dataset shows that the flow identifier using machine learning algorithm can effectively detect and track botnet attacks. Jiang et al. [12] put forward the PSO-Xgboost model. First, build a classification model through Xgboost, and then use the PSO algorithm to adaptively search the optimal structure of Xgboost. Experiments show that the overall classification accuracy of PSO-Xgboost model is higher than Xgboost, Random Forest, Bagging, Adaboost and other models. With the proliferation of network traffic data, traditional machine learning algorithms also show deficiencies. For example, machine learning, as a shallow learning algorithm, relies too much on feature selection, and its performance on large datasets is average.

Since Hinton et al. put forward the concept of deep learning, deep learning has been widely used in various fields. Aljbal et al. [13] proposed an anomaly detection method based on bidirectional short-term memory algorithm (Bi LSTM). Experiments on UNSW-NB15 dataset show that Bi LSTM algorithm is superior to other machine learning and deep learning models in accuracy, precision, F1 score and recall. Andresini et al. [14] proposed a method to analyze abnormal behaviors in network traffic using convolutional neural networks (CNN). The network flow is represented as a 2D image by performing a combination of nearest neighbor search and clustering processes, and the image is used to train the 2D CNN architecture. Yin et al. [15] proposed a deep learning method for intrusion detection using recurrent neural networks (RNN-IDS), and studied the performance of the model in binary classification and multi class classification, as well as the impact of the number of neurons and different learning rates on the performance of the proposed model. Faker et al. [16] proposed an intrusion detection system based on K-means homogeneity metric feature selection, and used deep feedforward neural network (DNN), RF and gradient lifting tree (GBT) for binary and multi classification. Experimental results show that this method performs well in processing large datasets.

At present, many methods have been proposed to solve the class imbalance problem of network intrusion detection. Sun et al. [17] used the hybrid network model (DL-IDS) of convolutional neural network (CNN) and short-term memory network (LSTM) for intrusion detection, and used the category weight optimization method to solve the impact of class imbalance dataset on model performance. The verification on CICIDS2017 dataset shows that the overall accuracy of multi classification is 98.67%, and the accuracy of each attack type is above 99.50%. Zhang et al. [18] proposed a flow based IDS method, which uses Gaussian mixture model (GMM) and SMOTE to deal with class imbalance in network data. Gupta et al. [19] proposed a LIO-IDS model based on Long Short Term Memory (LSTM) and improved one-to-one algorithm to deal with frequent and infrequent network intrusions, that is, imbalanced datasets. Experiments on NSL-KDD, CIDS-001 and CICIDS2017 datasets show that LIO-IDS has the advantages of high attack detection rate and fast computing time. Abdulhammed et al. [20] used the unified distributed balanced Uniform Distributed Based Balancing (UDBB) method to build a machine learning based network intrusion detection system (NIDS). The experimental results on CICIDS2017 dataset show that UDBB can effectively alleviate the problem of imbalanced data distribution, and the multi classification accuracy can reach 99.6%. Table 1 lists some related literature that contain ML-based and DL-based methods.

Table 1. Summary of some related methods.

| Ref. | Method | Description | Type |
|-----------|--------------|--|------|
| [8] 2018 | ARM, ANN, NB | Detect botnet | ML |
| [9] 2020 | PSO-Xgboost | Improve accuracy and efficiency | ML |
| [10] 2020 | LSTM | Outstanding performance. | DL |
| [11] 2021 | 2D CNN | Nearest cluster-based intrusion detection | DL |
| [12] 2017 | RNN | Good classification results. | DL |
| [13] 2019 | DNN, K-means | Feature representations, accuracy | DL |
| [14] 2020 | CNN-LSTM | Feature representations | DL |
| [15] 2020 | CNN, GMM | Imbalanced classification | DL |
| [16] 2021 | LIO-IDS | Handling class imbalance | DL |
| [17] 2019 | UDBB | Features dimensionality reduction approaches | ML |

In order to solve the problem of imbalanced data distribution on large-scale network intrusion detection systems, this paper proposes a CSK-CNN model that combines two-layer CNN and imbalanced dataset processing algorithm Cluster-SMOTE + K-means. This paper verifies the anomaly detection rate of the model in Layer 1 and the multiple attack identification rate in Layer 2 on UNSW-NB15 and CICIDS2017 datasets respectively.

3. Proposed Methods

The CSK-CNN architecture proposed in this paper is used to detect abnormal network behavior. This architecture combines the two-layer CNN algorithm and the class imbalance processing algorithm CSK (Cluster-SMOTE + K-means). A new class imbalance data processing algorithm CSK is proposed, which combines the use of Cluster-SMOTE algorithm for over sampling on minority classes and K-means algorithm based under sampling on majority classes. CSK-CNN is an anomaly based NIDS with a two-layer classification structure: Layer 1 and Layer 2. Layer 1 uses CNN binary classification to identify normal network traffic and abnormal network traffic. Then send the identified abnormal network traffic to the Layer 2. Layer 2 uses CNN multiple classification to classify abnormal network traffic into their respective attack categories. Therefore, the CSK-CNN model proposed in this paper can not only identify exceptions, but also distinguish attack types. In particular, the Layer 2 is multiple classifiers that distinguish attack categories are as important as classifiers in the Layer 1 of identifying attacks, because in the real world, only when we know the exact categories of intrusion attacks can we choose appropriate defense technologies to defend against attacks. Figure 1 shows the working mode of the proposed CSK-CNN model. Details are described below.

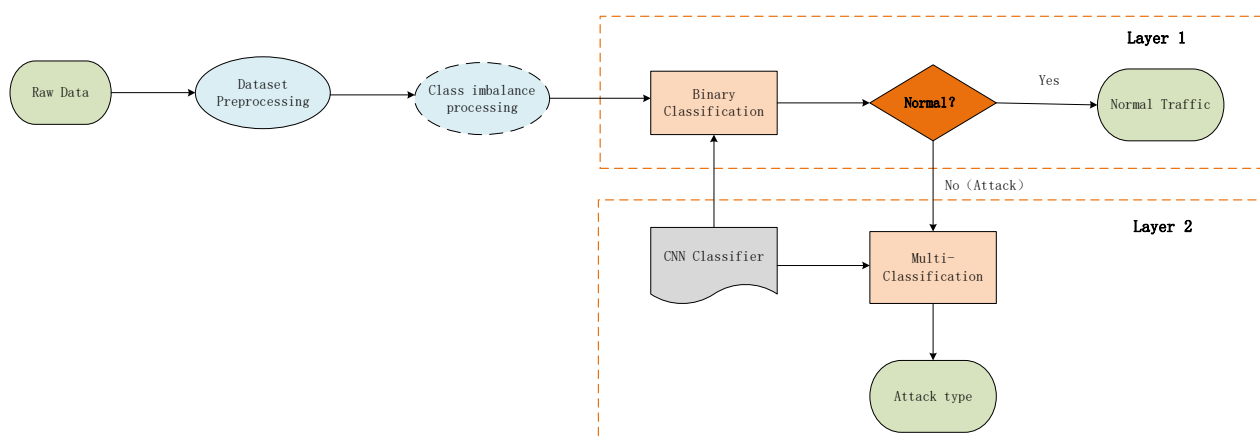


Figure 1. CSK-CNN intrusion detection model.

3.1. Dataset Preprocessing

Dataset preprocessing In this paper, the dataset preprocessing of network intrusion detection mainly includes three parts: feature reduction, quantification, and normalization. In the feature deletion part, first of all, we delete redundant and meaningless features. In the UNSW-NB15 dataset, we delete six features: “srcip”, “sport”, “dsport”, “dstip”, “ltime” and “stime”, reducing the UNSW-NB15 feature dimension from 47 to 41. In addition, we observed the original dataset and found that the characteristic values of the samples with “srcip” and “dstip” of 0 are identical, but the corresponding labels are different. Therefore, we deleted these invalid sample data with “srcip” and “dstip” of 0. In the CICIDS2017 dataset, we deleted six features: “Flow ID”, “Source IP”, “Source Port”, “Destination IP”, “Protocol”, and “Time stamp”, reducing the feature dimension from 84 to 77.

Quantization is achieved by converting the classified value of each nominal feature into a numerical value. There are three “proto”, “state” and “service” nominal features in the UNSW-NB15 dataset. The feature dimension of UNSW-NB15 changes from 41 to 73 through the one pot coding quantization method. The class labels of the two datasets can also be converted into quantifiable values using the One pot encoding method. CICIDS2017 dataset does not contain nominal features, so only class tags are required for one pot coding quantization. Quantization is very important because it can solve the nominal features that cannot be directly processed by machine learning algorithms. Because the final feature dimensions of these two datasets are not many, 77 and 73 respectively, and the model training time is within an acceptable range, this paper does not perform feature selection separately.

After quantization, we need to standardize all the numerical features, and use the standardized function `StandardScaler()` to change each feature into data with a mean value of 0 and a variance of 1. Standardization is very important for data pre-processing. On the one hand, the standardized data and the original data maintain the same linear relationship, and the training process will not be affected by different feature median ranges; On the other hand, it is helpful to improve the convergence speed and accuracy of the model.

3.2. Class Imbalance Preprocessing

In a real network, the number of samples for a specific abnormal traffic may be very small, which will greatly affect the performance of our model. In particular, it is difficult for minority class samples to find the correct class boundary, which makes it difficult to classify by defining the class region and boundary. Therefore, this paper proposes an algorithm CSK to solve the class imbalance dataset, that is, it combines the Cluster-SMOTE over sampling and K-means clustering based under sampling methods. This method not only solves the problem of information redundancy, time and space waste caused by only using random over sampling, but also solves the problem of information loss caused by only using random under sampling under the condition that the total amount of training data remains unchanged. Pseudo code visible Algorithm 1 of CSK algorithm proposed in this paper.

First, define the average sample quantity I of each category after resampling, as shown in Formula (1):

$$I = \text{Int}\left(\frac{N}{C}\right) \quad (1)$$

where, N is the total number of samples in the training set, and C is the number of categories.

3.2.1. Over Sampling Process Based on Cluster-SMOTE

When the number of samples is less than I_{resample} , we use the Cluster-SMOTE algorithm proposed by Cieslak et al. [21] to over sample the minority samples to I_{resample} . In order to estimate the region and boundary of minority samples, Cluster-SMOTE algorithm applies typical K-means clustering method to each minority sample. Then SMOTE method is used in each cluster to form a new sample set by reinserting composite samples in each

cluster. This method can improve the performance of SMOTE when the boundary of a few sets is unclear, as shown in Formulas (2) and (3):

$$K_j = K - \text{means}(I_i, C) \quad (2)$$

$$K_j' = \text{SMOTE}\left(K_j, \frac{I_{\text{resample}}}{C}\right) \quad (3)$$

where, C represents the number of clusters divided, K_j is the number of samples per cluster after K-means algorithm, and K_j' is the number of samples per cluster after sampling on SMOTE.

3.2.2. Under Sampling Process Based on K-Means Clustering

For the number of samples more than I_{resample} , we use K-means based clustering algorithm to under sample the majority samples to I_{resample} . The principle of this method is to initialize k cluster centers, count the samples under each cluster class based on the distance between the calculated samples and the center point, and iteratively realize that the distance between the samples and the center of the cluster class to which they belong is the minimum objective. The objective function is shown in Formula (4):

$$\text{argmin}_C J(C) = \sum_{k=1}^K \sum_{x^{(i)} \in C_k} \|x^{(i)} - \mu^{(i)}\|_2^2 \quad (4)$$

where, K is the number of samples in a cluster, and x is a sample point in the cluster, μ Represents the center of mass in the cluster, C_k represents the feature vector of each sample point, and i represents each feature of the component point x .

Algorithm 1 CSK

Input:

Training set $I = \{I_i, i = 1, 2, \dots, C\}$

C = the total number of classes;

$|I| = N$; # the total number of samples

Output:

a balanced training set needed I' ;

1: $I_{\text{resample}} = \text{Int}\left(\frac{N}{C}\right)$

2: **for** $i \leftarrow 1$ to C **do**

3: **if** $|I_i| < I_{\text{resample}}$ **then**

4: $K_j = K - \text{means}(I_i, C)$ # Use K-means to cluster I_i into C clusters,

5: $j = 1, 2, \dots, C$

6: **for** $j \leftarrow 1$ to C **do**

7: $K_j' = \text{SMOTE}\left(K_j, \frac{I_{\text{resample}}}{C}\right)$ # Use SMOTE to oversample K_j

8: **end for**

9: $I_i' = \text{Concatenate}(K_j')$

10: **end if**

11: $I' = \text{Concatenate}(I_i')$

12: **if** $|I_i| > I_{\text{resample}}$ **then**

13: $K_j = K - \text{means}(I_i, C)$ # Use K-means to cluster I_i into C clusters,

14: $j = 1, 2, \dots, C$

15: **for** $j \leftarrow 1$ to C **do**

16: $K_j' = \text{Resample}\left(K_j, \frac{I_{\text{resample}}}{C}\right)$ # Randomly select $\frac{I_{\text{resample}}}{C}$ samples from K_j

17: **end for**

18: $I_i' = \text{Concatenate}(K_j')$

19: **end if**

20: $I' = \text{Concatenate}(I_i')$

21: **end for**

22: **return** I'

3.3. Convolutional Neural Network

Convolution neural network is a kind of feedforward neural network, which has become one of the research focuses in many scientific fields. Convolutional neural network has the characteristics of local feature perception and parameter sharing, and can effectively classify network traffic from hierarchical structure.

Convolutional neural network mainly includes convolution layer, pooling layer and full connection layer. In general, convolution layer is used to extract local features, pooling layer prevents over fitting by reducing the number of parameters, and full connection layer integrates local features to form complete features. In this paper, 1D convolutional neural network is used for network intrusion detection. The network structure is shown in Figure 2, including eight layers of networks, namely, four layers of convolutional layer, two layers of pooling layer and two layers of full connection layer. First, the input information is automatically extracted through convolution operation. Assuming the Layer 1 is convolution layer, then l The calculation formula of x_j^l is shown in Formula (5):

$$x_j^l = f \left(\sum_{i \in M_j} x_j^{l-1} \otimes w_{ij} \right) + b_j^l \quad (5)$$

where, x_j^l is the j th output feature of layer l , $f(x)$ is a nonlinear activation function, reLU function is used in this paper, and \otimes is a convolution operation. The current feature is obtained by convolution operation on all associated features of layer $l - 1$ through convolution kernel w , and b is an offset parameter. The nonlinear operation of activation function can not only better map features and remove redundant information, but also enhance the expression ability of convolutional neural network.

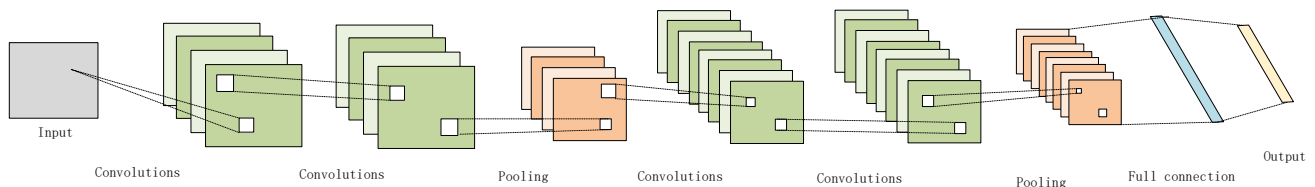


Figure 2. CNN Model Structure.

After the convolution layer, the dimension of the input data becomes higher and higher, and many parameters will be generated, which will not only greatly increase the difficulty of network training, but also cause the phenomenon of over fitting. Therefore, the dimension is reduced by pooling the layer data. The essence of pooling layer is under sampling. There are two common pooling operations, max pooling and mean pooling. This paper adopts the maximum pooling method.

The last is the full connection layer. The convolution layer and pooling layer in front are equivalent to feature engineering work. The full connection layer is equivalent to weighting features to form complete features. The last full connection layer will play a classifier role in the entire neural network through the softmax function.

4. Experimental Results and Analysis

The CSK-CNN intrusion detection model proposed in this paper is developed under Windows 10 operating system and Intel (R) Core (TM) i7-8700K processor environment. On UNSW-NB15 and CICIDS2017 datasets, experiments were conducted using python programming language to verify the effectiveness of CSK-CNN intrusion detection method.

4.1. Dataset Description

The UNSW-NB15 dataset [22] was created by the Cyber Range Lab of the Australian Cyber Security Center (ACCS) and is mainly used to generate normal traffic and attack

traffic in the real world. The dataset contains more than 2.54 million network traffic samples, involving one normal category and nine attack categories. Each sample has 49 features, and the last two columns are the binary label and attack type label. After data preprocessing, the dataset includes 2,204,107 normal samples and 281,896 abnormal samples. It can be seen that there is a serious imbalance in the dataset, which is suitable for evaluating the CSK algorithm proposed in this paper. On the UNSW-NB15 dataset, we split all the datasets into the training set, verification set and test set with a ratio of 7:1:2. The detailed data distribution of each category is shown in Table 2.

Table 2. UNSW-NB15 Dataset Description.

| Category | Trainset-Size | Testset-Size | Validset-Size | Total |
|----------------|---------------|--------------|---------------|-----------|
| Normal | 1,542,873 | 440,822 | 220,412 | 2,204,107 |
| Analysis | 436 | 124 | 62 | 622 |
| Backdoors | 250 | 71 | 36 | 357 |
| DoS | 2693 | 769 | 385 | 3847 |
| Exploits | 19,810 | 5660 | 2830 | 28,300 |
| Fuzzers | 15,060 | 4303 | 2151 | 21,514 |
| Generic | 149,602 | 42,744 | 21,372 | 213,718 |
| Reconnaissance | 8297 | 2371 | 1185 | 11,853 |
| Shellcode | 1057 | 303 | 151 | 1511 |
| Worms | 122 | 35 | 17 | 174 |
| Total | 1,740,200 | 497,202 | 248,601 | 2,486,003 |

The CICIDS2017 dataset was developed by the Canadian Institute of Network Security at the end of 2017 by generating and capturing network traffic that lasts for five days. Sharafaldin et al. [23] used the B-profile system to simulate normal and abnormal behaviors on the network. The dataset consists of eight CSV files, including 2,273,097 normal samples and 557,646 attack samples, including one normal category and 14 attack categories. It is suitable for use as a dataset to verify the class imbalance processing algorithm proposed in this paper. Among them, each sample has a total of 78 features, and the last column is a type label. The CICIDS2017 dataset is split in the same way as the UNSW-NB15 dataset. The detailed data distribution of each category is shown in Table 3.

Table 3. CICIDS2017 Dataset Description.

| Category | Trainset-Size | Testset-Size | Validset-Size | Total |
|--------------------------|---------------|--------------|---------------|-----------|
| BENIGN | 1,591,167 | 454,620 | 227,310 | 2,273,097 |
| DoS Hulk | 161,751 | 46,215 | 23,107 | 231,073 |
| PortScan | 111,251 | 31,786 | 15,893 | 158,930 |
| DDoS | 89,618 | 25,606 | 12,803 | 128,027 |
| DoS GoldenEye | 7205 | 2059 | 1029 | 10,293 |
| FTP-Patator | 5516 | 1588 | 794 | 7898 |
| SSH-Patator | 4128 | 1179 | 590 | 5897 |
| DoS slowloris | 4057 | 1159 | 580 | 5796 |
| DoS Slowhttptest | 3849 | 1100 | 550 | 5499 |
| Bot | 1376 | 393 | 197 | 1966 |
| Web Attack Brute Force | 1055 | 301 | 151 | 1507 |
| Web Attack XSS | 457 | 130 | 65 | 652 |
| Infiltration | 26 | 7 | 3 | 36 |
| Web Attack Sql Injection | 15 | 4 | 2 | 21 |
| Heartbleed | 8 | 2 | 1 | 11 |
| Total | 1,981,519 | 566,149 | 283,075 | 2,830,743 |

4.2. Evaluation Matrix

This paper uses six performance indicators to evaluate the proposed model: Accuracy (Acc), Recall, Precision, F1-score, false alarm rate (FAR), and receiver operating characteristic curve (ROC). For each type, we treat the samples as positive and the other samples as negative.

Acc refers to the percentage of correctly classified samples in the total number of samples, as shown in Formula (6).

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Recall refers to the percentage of correctly classified positive samples in the total number of positive samples, also known as true positive rate (TPR) or detection rate (DR), as shown in Formula (7).

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Precision refers to the percentage of positive samples among the samples classified as positive by the model, as shown in Formula (8).

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

F1-score is the harmonic average of Recall and Precision, as shown in Formula (9).

$$F1\text{-score} = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (9)$$

FAR is the false alarm rate, which refers to the percentage of negative samples wrongly classified as positive, as shown in Formula (10).

$$FAR = \frac{FP}{FP + TN} \quad (10)$$

where *TP* indicates the number of positive samples correctly identified, *FN* represents the number of negative samples incorrectly marked, *FP* represents the number of positive samples incorrectly marked, and *TN* represents the number of negative samples correctly identified. The confusion matrix of the proposed CSK-CNN in this paper is shown in Appendix A.

In multi classification, in order to more reasonably evaluate the classification performance of the model on the imbalanced dataset, the weighted averaging method, macro averaging macro averaging method and micro averaging micro averaging method are used to calculate and display each type.

4.3. Hyperparameters for Convolution Neural Network

Convolution neural network involves the selection of multiple hyperparameters, such as the number of convolution cores, learning rate, number of iterations, mini-batch-size, etc. Each hyperparameter directly affects the classification result of the model. After the hyperparameter is adjusted, the hyperparameters of the convolutional neural network model in this paper are shown in Table 4.

Table 4. Hyperparameters of convolutional neural network model.

| Hyperparameter | Value |
|-----------------------------|--------------------------|
| Convolutional kernel | 32-32-64-64 |
| Pool_size | 2×2 |
| Strides | 2 |
| Dropout | 0.2 |
| Full connection layer nodes | 128 |
| Learning rate | 0.008 |
| Iterations | 100 |
| mini-batch-size | 2048 |
| Activation function | ReLU |
| Optimization algorithm | Nadam |
| loss function | categorical_crossentropy |

The number of convolution cores of the four convolution layers is 32-32-64-64. The pooled layer uses the maximum pooling method with a sliding window of 2×2 and a step size of 2 to sample the parameters of the convolution layer twice, and uses a drop out layer with a parameter of 0.2 behind each pooled layer to prevent over fitting. Finally, the full connection layer uses 128 nodes for connection, and the number of nodes in the output layer is the number of categories. Except that the output layer uses Softmax as the activation function, other layers use ReLU. The optimization algorithm uses the best “Nadam” [24], the learning rate is set to 0.008, the number of iterations is set to 100, and mini-batch-size is set to 2048.

4.4. Layer 1: Binary Classification Results

The binary classification experiments in Layer 1, in order to prove the effectiveness of the CSK algorithm proposed in this paper, this paper compares five different class imbalance processing algorithms, namely SMOTE, ROS, ADASYN, RUS + SMOTE, K-means + SMOTE. The last two algorithms use RUS and K-means for under sampling respectively. In addition, in order to prove the validity of the proposed one-dimensional CNN model, this paper compares two machine learning classification algorithms, namely, random forest (RF) and Multi-Layer Perceptron (MLP). RF is the most representative method in the integration algorithm. For fair comparison, the default parameters are used in this paper, and the super parameters are not specially adjusted. As a typical neural network, MLP uses 128, 64 and 32 neural units to set three hidden layers.

Table 5 shows the binary classification results on UNSW-NB15 dataset. We can observe that the performance of the combined algorithm of under sampling and over sampling is significantly better than that of other class imbalance processing algorithms. The CSK-CNN model proposed in this paper obtains the best classification results, with Acc, Recall, FAR, Precision, F1 score reaching 99.14%, 98.70%, 0.80%, 94.03%, 96.31% respectively. Except Recall, other indicators are optimal. For MLP model, CSK algorithm is better than RUS + SMOTE algorithm in other indicators except that the Recall indicator is 0.56% lower than RUS + SMOTE algorithm. RF is the same. Except for the Recall index, other indexes of the CSK algorithm are optimal. Among them, the Recall index of the RUS + SMOTE algorithm is optimal, reaching 99.99%.

Table 5. Performance Results of Binary Classification on UNSW-NB15 Dataset.

| Model | Imbalanced Algorithms | Acc | Recall | FAR | Precision | F1-Score | Train-Times (s) | Test-Times (s) |
|-------|-----------------------|--------|--------|--------|-----------|----------|-----------------|----------------|
| RF | SMOTE | 0.9875 | 0.9999 | 0.0141 | 0.9006 | 0.9476 | 30.13 | 0.26 |
| | ROS | 0.9875 | 0.9999 | 0.0141 | 0.9006 | 0.9476 | 28.84 | 0.26 |
| | ADASYN | 0.9875 | 0.9999 | 0.0141 | 0.9006 | 0.9476 | 29.40 | 0.27 |
| | RUS + SMOTE | 0.9875 | 0.9999 | 0.0141 | 0.9005 | 0.9476 | 21.08 | 0.26 |
| | K-means + SMOTE | 0.9876 | 0.9996 | 0.0139 | 0.9020 | 0.9483 | 18.05 | 0.28 |
| | CSK | 0.9878 | 0.9980 | 0.0135 | 0.9046 | 0.9491 | 18.06 | 0.26 |
| MLP | SMOTE | 0.9884 | 0.9980 | 0.0128 | 0.9090 | 0.9514 | 2859.42 | 6.60 |
| | ROS | 0.9880 | 0.9990 | 0.0134 | 0.9054 | 0.9499 | 3059.28 | 6.61 |
| | ADASYN | 0.9878 | 0.9994 | 0.0137 | 0.9030 | 0.9488 | 2996.84 | 6.26 |
| | RUS + SMOTE | 0.9885 | 0.9995 | 0.0137 | 0.9031 | 0.9488 | 1739.53 | 6.54 |
| | K-means + SMOTE | 0.9885 | 0.9980 | 0.0128 | 0.9090 | 0.9515 | 1884.72 | 6.36 |
| | CSK | 0.9898 | 0.9939 | 0.0107 | 0.9222 | 0.9567 | 1887.67 | 6.71 |
| CNN | SMOTE | 0.9877 | 0.9996 | 0.0138 | 0.9023 | 0.9485 | 7836.51 | 19.21 |
| | ROS | 0.9883 | 0.9991 | 0.0131 | 0.9068 | 0.9507 | 6077.96 | 20.05 |
| | ADASYN | 0.9889 | 0.9967 | 0.0121 | 0.9131 | 0.9531 | 7789.34 | 22.46 |
| | RUS + SMOTE | 0.9892 | 0.9961 | 0.0117 | 0.9157 | 0.9543 | 4595.55 | 21.93 |
| | K-means + SMOTE | 0.9890 | 0.9972 | 0.0121 | 0.9137 | 0.9536 | 4508.74 | 19.40 |
| | CSK | 0.9914 | 0.9870 | 0.0080 | 0.9403 | 0.9631 | 4212.50 | 19.48 |

In terms of calculation time, it can be seen from Tables 5–8 that the training time of the classification model using the combined algorithm of under sampling and over sampling, such as RUS + SMOTE, K-means + SMOTE, and CSK, is significantly lower than that of the single over sampling algorithm, because the number of samples in the training set of the combined algorithm is lower than that of the single over sampling algorithm. Among

them, the classification of CNN models takes the longest time and the RF is the shortest, which is due to the complexity of the number of CNN model layers.

Table 6. Performance Results of Binary Classification on CICDS2017 Dataset.

| Model | Imbalanced Algorithms | Acc | Recall | FAR | Precision | F1-Score | Train-Times (s) | Test-Times (s) |
|-------|-----------------------|--------|--------|--------|-----------|----------|-----------------|----------------|
| RF | SMOTE | 0.9948 | 0.9854 | 0.0029 | 0.9883 | 0.9868 | 42.25 | 0.34 |
| | ROS | 0.9949 | 0.9856 | 0.0028 | 0.9884 | 0.9870 | 44.73 | 0.35 |
| | ADASYN | 0.9949 | 0.9856 | 0.0028 | 0.9884 | 0.9870 | 43.58 | 0.34 |
| | RUS + SMOTE | 0.9950 | 0.9856 | 0.0027 | 0.9889 | 0.9873 | 43.98 | 0.36 |
| | K-means + SMOTE | 0.9950 | 0.9855 | 0.0027 | 0.9889 | 0.9872 | 37.27 | 0.32 |
| | CSK | 0.9959 | 0.9989 | 0.0049 | 0.9804 | 0.9896 | 31.26 | 0.35 |
| MLP | SMOTE | 0.9981 | 0.9993 | 0.0022 | 0.9912 | 0.9953 | 3138.29 | 7.28 |
| | ROS | 0.9978 | 0.9994 | 0.0026 | 0.9896 | 0.9945 | 3276.21 | 7.56 |
| | ADASYN | 0.9977 | 0.9983 | 0.0025 | 0.9900 | 0.9942 | 3186.58 | 8.24 |
| | RUS + SMOTE | 0.9982 | 0.9994 | 0.0022 | 0.9912 | 0.9953 | 2586.21 | 7.01 |
| | K-means + SMOTE | 0.9984 | 0.9994 | 0.0017 | 0.9896 | 0.9945 | 2784.78 | 7.50 |
| | CSK | 0.9986 | 0.9984 | 0.0013 | 0.9946 | 0.9965 | 2462.86 | 7.55 |
| CNN | SMOTE | 0.9993 | 0.9998 | 0.0008 | 0.9969 | 0.9984 | 6242.24 | 30.58 |
| | ROS | 0.9994 | 0.9997 | 0.0007 | 0.9974 | 0.9985 | 6028.14 | 32.23 |
| | ADASYN | 0.9992 | 0.9996 | 0.0009 | 0.9965 | 0.9980 | 6342.18 | 28.46 |
| | RUS + SMOTE | 0.9992 | 0.9996 | 0.0009 | 0.9965 | 0.9980 | 4042.39 | 24.98 |
| | K-means + SMOTE | 0.9993 | 0.9997 | 0.0008 | 0.9969 | 0.9983 | 3956.54 | 22.51 |
| | CSK | 0.9994 | 0.9996 | 0.0006 | 0.9976 | 0.9986 | 3917.58 | 26.29 |

Table 7. Multi-classification Performance Results on UNSW-NB15 Dataset.

| Model | Imbalanced Algorithms | Acc | Recall | Precision | F1-Score | Train-Times (s) | Test-Times (s) |
|-------|-----------------------|--------|--------|-----------|----------|-----------------|----------------|
| RF | SMOTE | 0.9205 | 0.9205 | 0.9534 | 0.9304 | 80.47 | 0.3441 |
| | ROS | 0.9209 | 0.9209 | 0.9532 | 0.9309 | 70.12 | 0.3361 |
| | ADASYN | 0.9231 | 0.9231 | 0.9519 | 0.9318 | 69.77 | 0.3381 |
| | RUS + SMOTE | 0.9232 | 0.9232 | 0.9525 | 0.9322 | 10.08 | 0.3351 |
| | K-means + SMOTE | 0.9223 | 0.9223 | 0.9521 | 0.9314 | 8.64 | 0.3431 |
| | CSK | 0.9312 | 0.9312 | 0.9465 | 0.9348 | 9.21 | 0.34 |
| MLP | SMOTE | 0.9441 | 0.9441 | 0.9512 | 0.9472 | 1740.83 | 1.38 |
| | ROS | 0.9492 | 0.9492 | 0.9530 | 0.9509 | 1595.38 | 0.96 |
| | ADASYN | 0.9374 | 0.9374 | 0.9469 | 0.9421 | 1708.83 | 1.00 |
| | RUS + SMOTE | 0.9378 | 0.9378 | 0.9537 | 0.9439 | 1053.86 | 1.85 |
| | K-means + SMOTE | 0.9414 | 0.9414 | 0.9545 | 0.9465 | 995.43 | 1.09 |
| | CSK | 0.9523 | 0.9523 | 0.9538 | 0.9525 | 988.85 | 0.78 |
| CNN | SMOTE | 0.9472 | 0.9472 | 0.9588 | 0.9518 | 3223.57 | 4.20 |
| | ROS | 0.9477 | 0.9477 | 0.9579 | 0.9517 | 3836.51 | 4.27 |
| | ADASYN | 0.9435 | 0.9435 | 0.9561 | 0.9486 | 3273.56 | 5.10 |
| | RUS + SMOTE | 0.9462 | 0.9462 | 0.9580 | 0.9508 | 1538.40 | 5.23 |
| | K-means + SMOTE | 0.9469 | 0.9469 | 0.9588 | 0.9510 | 1863.28 | 5.09 |
| | CSK | 0.9548 | 0.9548 | 0.9597 | 0.9560 | 1448.41 | 4.26 |

Table 8. Multi-classification Performance Results on CICDS2017 Dataset.

| Model | Imbalanced Algorithms | Acc | Recall | Precision | F1-Score | Train-Times (s) | Test-Times (s) |
|-------|-----------------------|--------|--------|-----------|----------|-----------------|----------------|
| RF | SMOTE | 0.9976 | 0.9976 | 0.9983 | 0.9977 | 32.59 | 1.01 |
| | ROS | 0.9976 | 0.9976 | 0.9983 | 0.9977 | 33.96 | 1.00 |
| | ADASYN | 0.9976 | 0.9976 | 0.9983 | 0.9977 | 33.51 | 1.25 |
| | RUS + SMOTE | 0.9977 | 0.9977 | 0.9984 | 0.9978 | 20.81 | 0.90 |
| | K-means + SMOTE | 0.9977 | 0.9977 | 0.9983 | 0.9978 | 35.02 | 0.89 |
| | CSK | 0.9977 | 0.9977 | 0.9983 | 0.9978 | 29.07 | 0.90 |
| MLP | SMOTE | 0.9978 | 0.9978 | 0.9984 | 0.9979 | 859.77 | 2.08 |
| | ROS | 0.9978 | 0.9978 | 0.9981 | 0.9978 | 901.44 | 1.67 |
| | ADASYN | 0.9978 | 0.9978 | 0.9981 | 0.9978 | 893.52 | 1.46 |
| | RUS + SMOTE | 0.9978 | 0.9978 | 0.9984 | 0.9978 | 567.42 | 1.52 |
| | K-means + SMOTE | 0.9977 | 0.9977 | 0.9980 | 0.9977 | 594.05 | 2.06 |
| | CSK | 0.9979 | 0.9979 | 0.9982 | 0.9979 | 540.80 | 1.96 |
| CNN | SMOTE | 0.9979 | 0.9979 | 0.9985 | 0.9980 | 1638.52 | 4.72 |
| | ROS | 0.9979 | 0.9979 | 0.9985 | 0.9980 | 1683.26 | 5.39 |
| | ADASYN | 0.9979 | 0.9979 | 0.9985 | 0.9980 | 1678.92 | 4.36 |
| | RUS + SMOTE | 0.9979 | 0.9979 | 0.9985 | 0.9980 | 669.13 | 6.55 |
| | K-means + SMOTE | 0.9980 | 0.9980 | 0.9985 | 0.9980 | 648.53 | 6.48 |
| | CSK | 0.9980 | 0.9980 | 0.9986 | 0.9982 | 653.85 | 4.73 |

Figure 3 shows the binary ROC curve on UNSW-NB15 and CICIDS2017 datasets. As shown in Figure 3a,b, it can be seen intuitively and clearly that the performances of the three classification algorithms are not different, and the AUC value of the area under the ROC curve is close to 1.

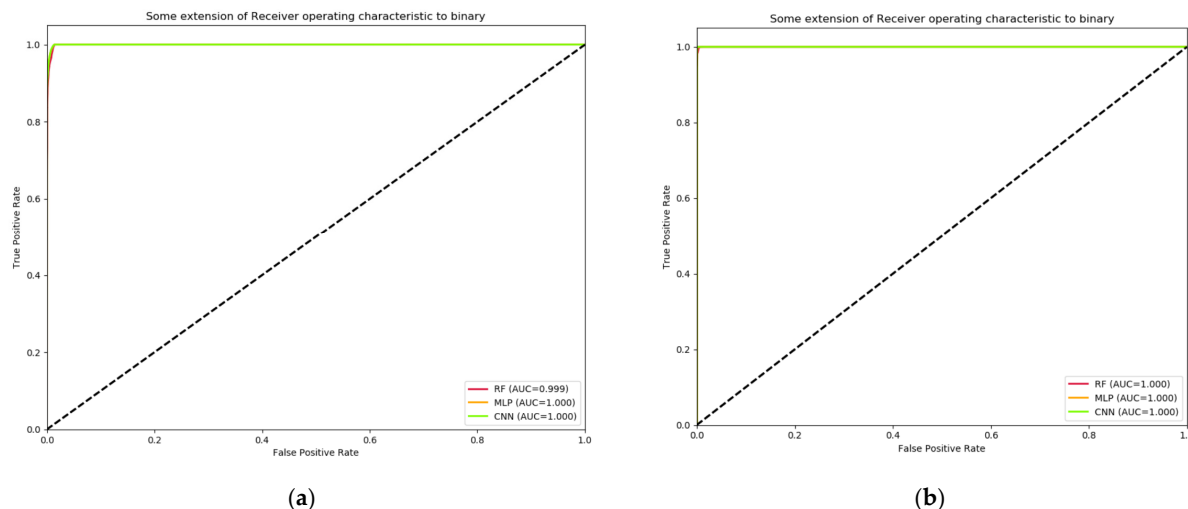


Figure 3. ROC curve of binary classification. (a) ROC curve for UNSW-NB15; (b) ROC curve for CICIDS2017.

Table 6 shows the binary classification results on CICIDS2017 dataset. Similarly, we can observe that the performance of the combined algorithm of under sampling and over sampling is significantly better than that of other class imbalance processing algorithms. The CSK-CNN model proposed in this paper obtains the best classification results, with Acc, Recall, FAR, Precision, and F1 score reaching 99.94%, 99.96%, 0.06%, 99.76%, and 99.86% respectively. In addition to Recall, other indicators are optimal, and the Recall of SMOTE algorithm is optimal, reaching 99.98%. For MLP model, CSK algorithm is the best except that Recall index is 0.1% lower than RUS + SMOTE. For RF model, the evaluation indexes of CSK algorithm are superior to other imbalance processing algorithms.

4.5. Layer 2: Multi Classification Results

In Layer 2, the multi classification experiment of abnormal samples uses the same classification model and class imbalance processing algorithm as the binary classification. On the UNSW-NB15 dataset and CICIDS2017 dataset, the number of neural units in the output layer of CNN and MLP models is 9 and 14 respectively, that is, the number of abnormal sample types. Other parameters are the same as those of the binary classification. The RF model uses default parameters for training.

Table 7 shows the multi classification results calculated using the weighted average method on the UNSW-NB15 dataset. It can be observed from Table 7 that the CSK algorithm is obviously superior to other imbalance processing algorithms. The CSK-CNN model has better overall optimal classification performance than other models. In terms of Acc, Recall, Precision and F1-score indicators, they respectively reach 95.48%, 95.48%, 95.97% and 95.60%. Among them, CNN model is slightly better than MLP model, and RF performance is the worst. For MLP and RF models, the CSK model proposed in this paper has a slightly lower precision, but Acc, Recall, and F1 score are higher than other imbalance processing algorithms. In particular, F1 score is the harmonic average of Precision and Recall. The high F1 score means that the algorithm proposed in this paper has higher overall classification performance than other algorithms.

Figure 4 shows the multi category ROC curve on UNSW-NB15 dataset. As shown in Figure 4b,c, it can be seen intuitively and clearly that the CSK-CNN model proposed in this paper has the best performance under macro average and micro average calculation, and

the AUC value of the area under the ROC curve is the largest. In addition, in Figure 4a, we can see the classification performance of each attack type under the CSK-CNN model intuitively. Backdoor, DoS, and Worms have poor detection performance, which is one of the main factors affecting the overall classification performance.

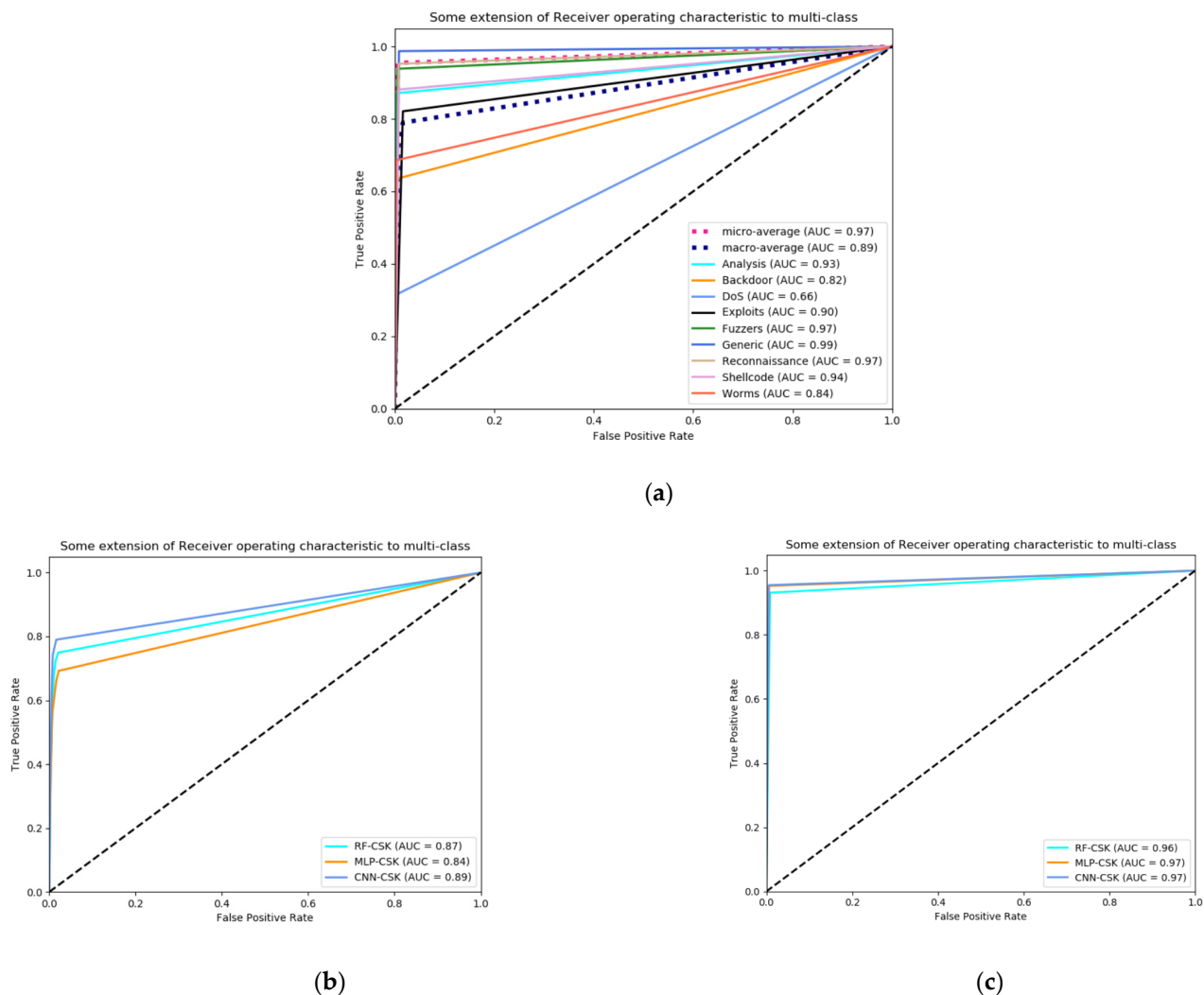
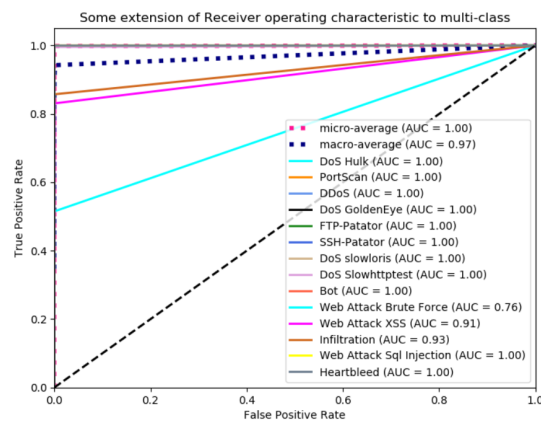


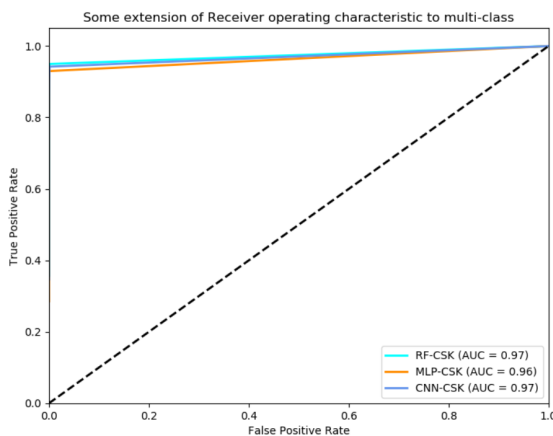
Figure 4. ROC curve of multi-classification for UNSW-NB15. (a) 9-class; (b) macro; (c) micro.

Table 8 shows the multi classification results calculated using the weighted average method on the CICIDS2017 dataset. It can be observed from Table 8 that the CSK-CNN algorithm has better overall optimal classification performance, reaching 99.80%, 99.80%, 99.86% and 99.82% respectively in Acc, Recall, Precision and F1 score indicators. Among them, the classification performance of MLP model is similar to that of CNN model, but CNN is slightly better than MLP model, and RF performance is the worst. For MLP and RF models, similar to UNSW-NB15 dataset, the CSK model proposed in this paper has a slightly lower precision, but Acc, Recall, and F1 score are higher than other class imbalance processing algorithms.

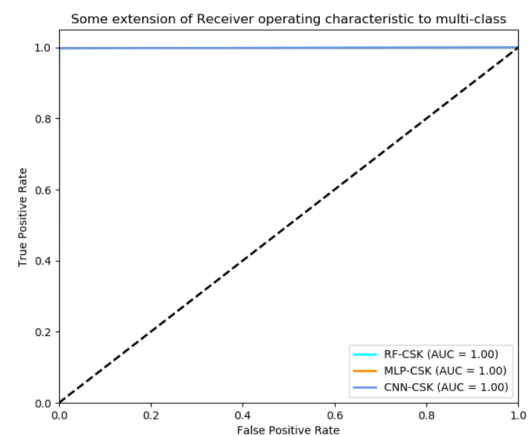
Figure 5 shows the multi category ROC curve on the CICIDS2017 dataset. As shown in Figure 5b,c, it can be seen intuitively and clearly that the CSK-CNN model proposed in this paper has the best performance under macro average and micro average calculation. Like the RF model, the AUC value of the area under the ROC curve is the largest. In addition, in Figure 5a, we can see the classification performance of each attack type under the CSK-CNN model intuitively. Web Attack Brute Force has poor detection performance, reaching 80%, which is one of the main factors affecting the overall classification performance.



(a)



(b)



(c)

Figure 5. ROC curve of multi-classification for CICIDS2017. (a) 14-class; (b) macro; (c) micro.

4.6. Overall Performance of Network Intrusion Detection

Through the two-layer classification algorithm, we can calculate the overall accuracy of network intrusion detection Acc of the CSK-CNN model proposed in this paper. In addition, we compare the CSK-CNN model proposed in this paper with the current four latest works on UNSW-NB15 dataset and CICIDS2017 dataset, as shown in Table 9. Table 9 shows that the CSK-CNN intrusion detection algorithm proposed in this paper has the highest anomaly detection accuracy compared with the four latest works. The overall detection accuracy of UNSW-NB15 dataset is 98.77%, and that of CICIDS2017 dataset is 99.91%.

Table 9. A comparison of proposed CSK-CNN with the current state-of-the-art models.

| Dataset | Model | Acc (%) |
|------------|------------------|---------|
| UNSW-NB15 | CSCADE-ANN [25] | 95.98 |
| | Chameleon [26] | 89.52 |
| | ICVAE-DNN [27] | 89.08 |
| | SMOTE + GMM [19] | 96.54 |
| | Proposed CSK-CNN | 98.77 |
| CICIDS2017 | CFS-BA [28] | 99.89 |
| | PCA + RF [20] | 99.60 |
| | DNN [18] | 99.57 |
| | PCCN [29] | 99.87 |
| | Proposed CSK-CNN | 99.91 |

5. Conclusions

In this paper, in order to solve the problem that class imbalance in intrusion detection datasets affects the performance of classifiers, we propose a two-layer network detection model CSK-CNN, which combines the class imbalance processing algorithm Cluster-SMOTE + K-means (CSK) and convolutional neural network. In Layer 1, binary classification is used to separate normal traffic and abnormal traffic, and in Layer 2, multi-classification is used to further classify abnormal traffic into specific attack categories. Compared with five kinds of imbalance processing algorithms and two kinds of classification algorithms, the CSK-CNN model proposed in this paper has the overall best classification performance, the overall detection accuracy on the UNSW-NB15 dataset reaches 98.77%, and that of CICIDS2017 dataset reaches 99.91%. So it is suitable for deployment in real networks. In the future, we plan to explore other methods to improve the classification performance of abnormal categories, such as Dos, Backdoor, Web Attack Brute Force, etc.

Author Contributions: Conceptualization, J.S. and X.W.; data curation, L.J.; methodology, J.S. and L.J.; supervision, M.H., X.W. and L.J.; writing—original draft, J.S.; writing—review and editing, X.W., M.H. and L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62071056.

Institutional Review Board Statement: The study did not involve humans or animals.

Informed Consent Statement: The study did not involve humans.

Data Availability Statement: The datasets used in this paper are available online [22,23], and they are also available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Confusion Matrix obtained at Layer 1 of the proposed CSK-CNN is shown in Tables A1 and A2.

Table A1. Layer 1 confusion matrix for UNSW-NB15 Dataset.

| UNSW-NB15 | Normal | Attack |
|-----------|---------|--------|
| Normal | 437,288 | 3534 |
| Attack | 735 | 55,645 |

Table A2. Layer 1 confusion matrix for CICIDS2017 Dataset.

| CICIDS2017 | Normal | Attack |
|------------|---------|---------|
| Normal | 454,351 | 269 |
| Attack | 48 | 111,481 |

Confusion Matrix obtained at Layer 2 of the proposed CSK-CNN is shown in Figures A1 and A2.

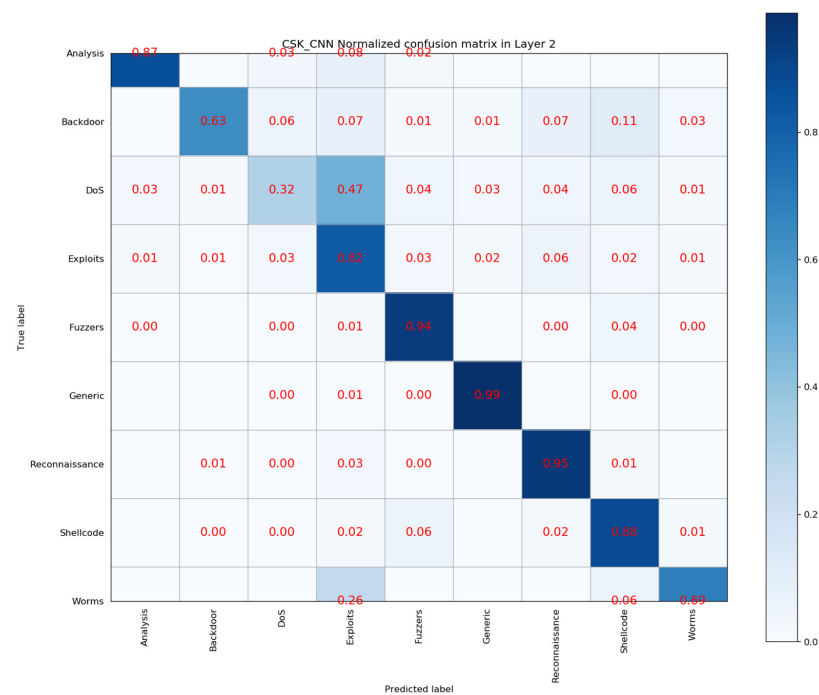


Figure A1. Layer 2 confusion matrix for UNSW-NB15 Dataset.

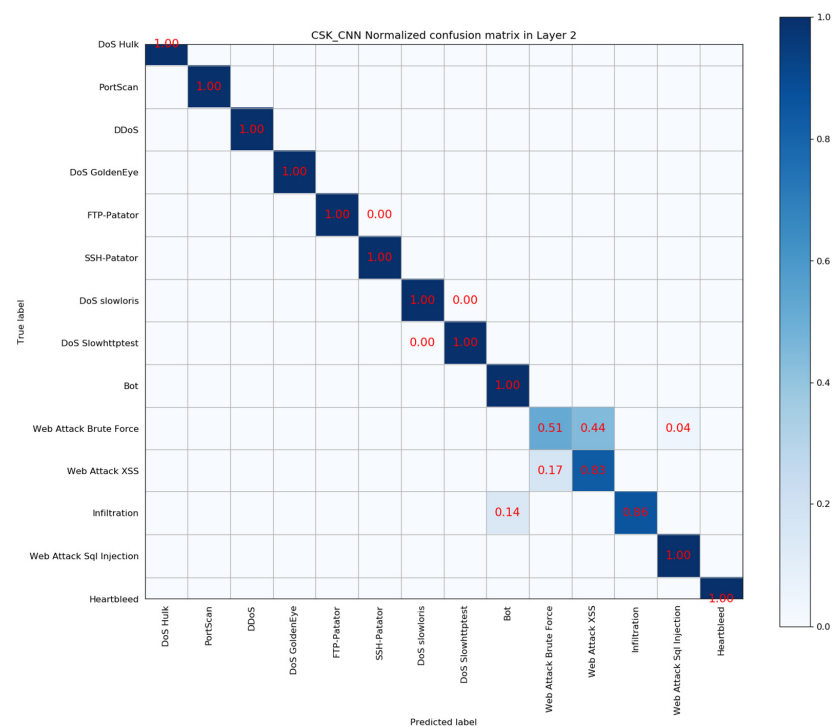


Figure A2. Layer 2 confusion matrix for CICIDS2017 Dataset.

References

1. Abboud, Z.A.; Khaleel, I.; Aggarwal, K. Challenges and Future Directions for Intrusion Detection Systems Based on AutoML. *Mesop. J. CyberSecurity* **2021**, *2021*, 16–21. [\[CrossRef\]](#)
2. Alajanbi, M.; Ismail, M.A.; Hasan, R.A.; Sulaiman, J. Intrusion Detection: A Review. *Mesop. J. CyberSecurity* **2021**, *2021*, 1–4. [\[CrossRef\]](#)
3. Umamaheswari, A.; Kalaavathi, B. Honeypot TB-IDS: Trace back model based intrusion detection system using knowledge based honeypot construction model. *Clust. Comput.* **2019**, *22*, 14027–14034. [\[CrossRef\]](#)

4. Zong, W.; Chow, Y.-W.; Susilo, W. Interactive three-dimensional visualization of network intrusion detection data for machine learning. *Future Gener. Comput. Syst.* **2020**, *102*, 292–306. [\[CrossRef\]](#)
5. Ravale, U.; Marathe, N.; Padiya, P. Feature selection based hybrid anomaly intrusion detection system using k-means and RBF kernel function. *Procedia Comput. Sci.* **2015**, *45*, 428–435. [\[CrossRef\]](#)
6. Chen, T.Q.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
7. Sangkatsanee, P.; Wattanapongsakorn, N.; Charnsripinyo, C. Practical real-time intrusion detection using machine learning approaches. *Comput. Commun.* **2011**, *34*, 2227–2235. [\[CrossRef\]](#)
8. Liu, Y.; Wang, C.; Zhang, Y.; Yuan, J. Multiscale convolutional CNN model for network intrusion detection. *Comput. Eng. Appl.* **2019**, *55*, 90. [\[CrossRef\]](#)
9. Sheikhan, M.; Jadidi, Z.; Farrokhi, A. Intrusion detection using reduced-size RNN based on feature grouping. *Neural Comput. Appl.* **2012**, *21*, 1185–1190. [\[CrossRef\]](#)
10. Althubiti, S.A.; Jones, E.M., Jr.; Roy, K. LSTM for anomaly-based network intrusion detection. In Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 21–23 November 2018.
11. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Slay, J. Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques. *Mob. Netw. Manag.* **2018**, *235*, 30–44. [\[CrossRef\]](#)
12. Jiang, H.; He, Z.; Ye, G.; Zhang, H. Network intrusion detection based on PSO-XGBoost model. *IEEE Access* **2020**, *8*, 58392–58401. [\[CrossRef\]](#)
13. Aljbali, S.; Roy, K. Anomaly detection using bidirectional LSTM. In *Intelligent Systems and Applications. IntelliSys 2020*; Advances in Intelligent Systems and Computing; Springer International Publishing: London, UK, 2020.
14. Andresini, G.; Appice, A.; Malerba, D. Nearest cluster-based intrusion detection through convolutional neural networks. *Knowl.-Based Syst.* **2021**, *216*, 106798. [\[CrossRef\]](#)
15. Yin, C.; Zhu, Y.; Fei, J.; He, X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **2017**, *5*, 21954–21961. [\[CrossRef\]](#)
16. Faker, O.; Dogdu, E. Intrusion detection using big data and deep learning techniques. In Proceedings of the ACMSE 2019, Kennesaw, GA, USA, 18–20 April 2019. [\[CrossRef\]](#)
17. Sun, P.; Liu, P.; Li, Q.; Liu, C.; Lu, X.; Hao, R.; Chen, J. DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. *Sec. Commun. Netw.* **2020**, *2020*, 8890306. [\[CrossRef\]](#)
18. Zhang, H.; Huang, L.; Wu, C.Q.; Li, Z. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **2020**, *177*, 107315. [\[CrossRef\]](#)
19. Gupta, N.; Jindal, V.; Bedi, P. LIO-IDS: Handling class imbalance using LSTM and Improved One-vs-One technique in Intrusion Detection System. *Comput. Netw.* **2021**, *192*, 108076. [\[CrossRef\]](#)
20. Abdulhammed, R.; Musaffer, H.; Alessa, A.; Faezipour, M.; Abuzneid, A. Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics* **2019**, *8*, 322. [\[CrossRef\]](#)
21. Cieslak, D.A.; Chawla, N.V.; Striegel, A. Combating imbalance in network intrusion datasets. In Proceedings of the 2006 IEEE International Conference on Granular Computing, Atlanta, GA, USA, 10–12 May 2006; pp. 732–737. [\[CrossRef\]](#)
22. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive dataset for network intrusion detection systems (UNSW-NB15 network dataset). In Proceedings of the IEEE: 2015 Military Communications and Information Systems Conference, IEEE, Canberra, ACT, Australia, 10–12 November 2015.
23. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Funchal, Madeira, Portugal, 22–24 January 2018.
24. Tahmassebi, A.; Gandomi, A.H.; Fong, S.; Meyer-Baese, A.; Foo, S.Y. Multistage optimization of a deep model: A case study on ground motion modeling. *PLoS ONE* **2018**, *13*, e0203829. [\[CrossRef\]](#)
25. Baig, M.M.; Awais, M.M.; El-Alfy, E.-S.M. A multiclass cascade of artificial neural network for network intrusion detection. *J. Intell. Fuzzy Syst.* **2017**, *32*, 2875–2883. [\[CrossRef\]](#)
26. Chohra, A.; Shirani, P.; Karbab, E.B.; Debbabi, M. Chameleon: Optimized feature selection using particle swarm optimization and ensemble methods for network anomaly detection. *Comput. Secur.* **2022**, *117*, 102684. [\[CrossRef\]](#)
27. Yang, Y.; Zheng, K.; Wu, C.; Yang, Y. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors* **2019**, *19*, 2528. [\[CrossRef\]](#)
28. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [\[CrossRef\]](#)
29. Zhang, Y.; Chen, X.; Guo, D.; Song, M.; Teng, Y.; Wang, X. PCCN: Parallel cross convolutional neural network for abnormal network traffic flows detection in multiclass imbalanced network traffic flows. *IEEE Access* **2019**, *7*, 119904–119916. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.