Received April 29, 2021, accepted May 16, 2021, date of publication May 20, 2021, date of current version May 27, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3082147*

# A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning

**CHAO LIU[1,2], ZHAOJUN GU[2,3], AND JIALIANG WANG[3]**

[1]College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China
[2]Information Security Evaluation Center, Civil Aviation University of China, Tianjin 300300, China
[3]College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Chao Liu (liuc@cauc.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities of Civil Aviation University of China under Grant 3122020064.

**ABSTRACT** Digital assets have come under various network security threats in the digital age. As a kind of security equipment to protect digital assets, intrusion detection system (IDS) is less efficient if the alert is not timely and IDS is useless if the accuracy cannot meet the requirements. Therefore, an intrusion detection model that combines machine learning with deep learning is proposed in this paper. The model uses the k-means and the random forest (RF) algorithms for the binary classification, and distributed computing of these algorithms is implemented on the Spark platform to quickly classify normal events and attack events. Then, by using the convolutional neural network (CNN), long short-term memory (LSTM), and other deep learning algorithms, the events judged as abnormal are further classified into different attack types finally. At this stage, adaptive synthetic sampling (ADASYN) is adopted to solve the unbalanced dataset. The NSL-KDD and CIS-IDS2017 datasets are used to evaluate the performance of the proposed model. The experimental results show that the proposed model has better TPR for most of attack events, faster data preprocessing speed, and potentially less training time. In particular, the accuracy of multi-target classification can reach as high as 85.24% in the NSL-KDD dataset and 99.91% in the CIC-IDS2017 dataset.

**INDEX TERMS** Intrusion detection system, machine learning algorithm, k-means, random forest, deep learning algorithm.

## I. INTRODUCTION

The world is moving towards digitization, networking, and intelligence. The vigorous development of the internet has accelerated the flow of data assets. In particular, the Internet of things (IoT) with various devices interconnected has promoted the explosion and exchange of data. This change brings not only convenience to people but also hidden dangers to digital data. Therefore, protecting data assets from infringement or theft is a critical challenge and has become a desirable research direction for security scholars. There are countless attacks on personal interests and commercial games on the Internet and the IoT. In recent years, both the attack method and the number of attacks have increased dramatically. This brings unpredictable risks to the safe and stable operation of the Internet and the IoT. Therefore, a powerful tool to effectively identify various network security threats and resist these attacks is urgently needed. IDS is a kind of security device that can block illegal connections and resist illegal external attacks. IDS can provide confidentiality, integrity, and usability for the Internet and the IoT. There is no argument that IDS is essential to protect the Internet and the IoT.

Many scholars use various machine learning algorithms for the routine detection of abnormal data [1]. In addition, deep learning algorithms that have been successfully applied to image recognition [2], [3], motion analysis [4], and natural language processing [5] are also used for intrusion detection. Shallow machine learning algorithms have been proven to have good detection performance, such as Random forest (RF) [6]. But the detection accuracy needs to be improved. Deep learning algorithms, implemented by imitating the human brain's thinking and discriminating behavior through

The associate editor coordinating the review of this manuscript and approving it for publication was Yanjiao Chen.

neurons, can tap hidden intrusion features. Nevertheless, algorithms such as DNN [7], Convolutional Neural Network (CNN) [8], and Long Short-Term Memory (LSTM) [9], take too much time to train the model. Therefore, a single machine learning algorithm or deep learning algorithm cannot meet the requirements of the IDS in the new era.

To the best of our knowledge, there are many hybrid intrusion detection methods, for example, by combining two or more machine learning algorithms such as SVM and k-means [10], naive Bayes and support vector machine [11], SVM, modified naive Bayes (MNB), and LPBoost [12], one-class support vector machine with RBF kernel and recursive k-means clustering [13]. In addition, there are combinations of two deep learning algorithms. For example, Vinayakumar passed feature sets of connection records to a CNN and a recurrent neural network (RNN) and to its variants such as LSTM and gated recurrent unit (GRU) [14] in 2017. Wei Wang proposed [15] using CNN and LSTM networks to learn the low-level spatial features and the high-level temporal features in 2017. Lei Bai proposed an LSTM-CNN cascade model to automatically identify the semantic type of a device in 2018 [16]. Also, there are combinations of deep learning algorithms and machine learning algorithms. For example, Salama proposed a deep learning approach with deep belief network (DBN) as a feature selector and an SVM as a classifier in 2011 [17]. Mighan and Kahani [18] used a stacked auto-encoding network for feature extraction then used SVM, RF, and other classification methods for classification in 2020. In the same year, Souza *et al.* [19] proposed a DNN-KNN hybrid binary classification method.

In addition to these studies, there are also combinations of distributed machine learning algorithms and deep learning algorithms. Studies by many scholars have shown that distributed machine learning algorithms [20], [21] can process high-dimensional data quickly and efficiently, so they can be used to process massive amounts of data in the initial phase of the two-stage classification of the IDS. At the multi-target anomaly classification stage of IDS, deep learning algorithms can tap into hidden features and discover unknown attack types while ensuring accuracy. To the best of our knowledge, however, there is little researchs on intrusion detection in this area. Khan *et al.* [22] proposed a two-level cascade method in 2019, which implemented the first stage in Scala using Spark MLlib as the ML platform and the Conv-LSTM network in Python using Keras. The time cost is increased when the dataset is transformed on different platforms many times. Many of these methods classify only between normal and attack events without further subdividing attack events. These hybrid methods perform well on known datasets, but considering there will be more novel attack methods and more information from other sources in the future, an improvement is necessary. Therefore, ensuring the speed of intrusion detection without affecting the accuracy is an important problem that needs to be considered for future IDS models. Based on these studies, this paper considers improving the model so that different attack events can be classified more quickly thus making the model more useful. Distributed k-means, RF, and deep learning model algorithms are proposed to solve the current problems faced by IDS. k-means and RF implemented in Spark, which is a unified analytics engine for large-scale data processing. Spark can make practical machine learning scalable and easy.

Therefore, the contribution of this paper is summarized as follows:

1. To propose a cascade intrusion detection method based on distributed machine learning and deep learning, which can be used for high dimensional massive data;

2. To use deep learning on Spark's driver side to learn only the hidden features of attack samples and Adaptive Synthetic Sampling (ADASYN) to avoid the polarization of the classifier, which not only reduces the coupling between normal and attack events but also reduces the time of data processing and transformation;

3. The comparison of overall accuracy and TPR on the NSL-KDD and CIC-IDS2017 datasets, which shows good TPR for most attack events and higher accuracy for multi-target classification.

The rest of this paper is described as follows. Section 2 presents the intrusion detection framework and the related algorithms used in our paper. The detailed process of the intrusion detection model proposed in this paper is introduced in Section 3. The experimental design and the results are given in Section 4. Finally, Section 5 concludes our paper, and future work is also conceived.

## II. INTRUSION DETECTION FRAMEWORK AND KEY TECHNOLOGIES

The intrusion detection model proposed in this paper is shown in Figure 1. It is divided into three stages. The first is the processing stage of the original intrusion detection dataset, including the digital processing of the categorical features, the normalization processing of the digital features, and the deletion of useless data. The second stage is to classify normal and abnormal events based on the combination of k-means and RF on Spark. In the third stage, the hidden features are learned through deep learning algorithms. As we can see from the Figure 1, Spark is the foundation of the proposed model. Before forecasting, the corresponding packages of Spark need to be imported in advance. Spark runs in a standalone mode in the experiment of this paper. The training set in the third stage comes from the results of the sampling by ADASYN after removing normal events in the first stage, and test sets of the third stage come from the remaining results after separating test sets in the second stage.

### A. DISTRIBUTED K-MEANS ALGORITHM IN SECOND STEP

The key steps of the k-means algorithm is to select k points as cluster centers, calculate the distance between other sample points and k cluster centers respectively, then classify each point to the nearest cluster center. These steps are repeated until the set suspension condition is met. The k-means

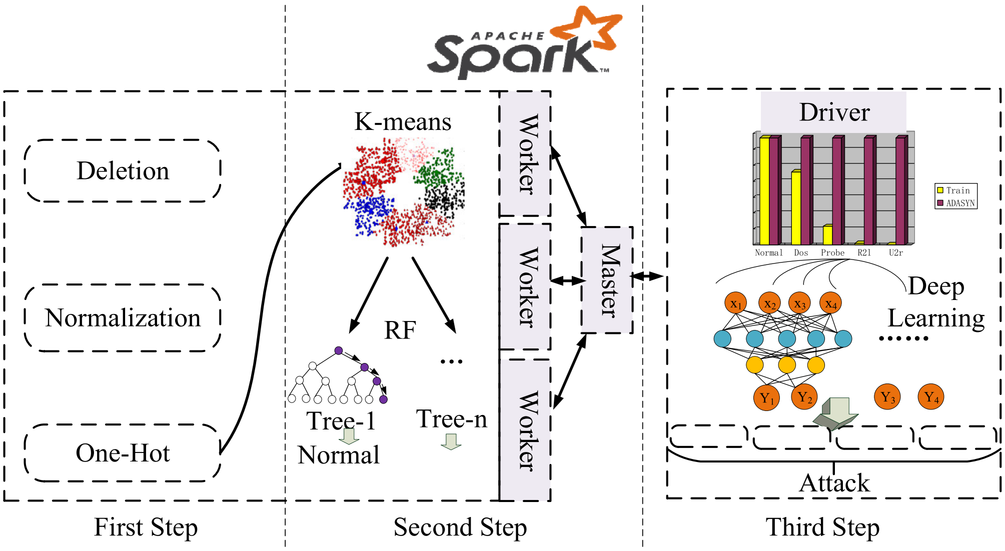


**FIGURE 1.** Intrusion detection framework proposed in this paper.

algorithm calculates the distance between the cluster center and the sample point by using Euclidean distance. The corresponding formula is calculated as Formula 1, where $x_1$ and $x_2$ represent two events with n fields and d$(x_1, x_2)$ represents the Euclidean distance between $x_1 = (x_{11}, x_{12}, \ldots\ldots, x_{1\mathrm{n}})$ and $x_2 = (x_{21}, x_{22}, \ldots\ldots, x_{2\mathrm{n}})$. All the experiments are implemented in Spark. The preprocessed data will be divided into multiple subsets, which correspond to different resilient distributed datasets (RDDs). Each RDD stores multiple partitions. The k-means clusters the partitions in each RDD and brings the final results together. Spark starts multiple threads in parallel to compute the partitions in different workers. The initial cluster center is 8, so the final cluster result is 8. The next step is to classify these clusters by the RF algorithm.

$$\mathrm{d}(x_1, x_2) = \sqrt{(x_{11}-x_{21})^2+(x_{12}-x_{22})^2+\cdots\cdots+(x_{1\mathrm{n}}-x_{2n})^2} \tag{1}$$

### B. DISTRIBUTED RANDOM FOREST ALGORITHM IN SECOND STEP

The RF algorithm is an integrated machine learning algorithm that builds decision trees through random node splitting and random node resampling. The final classification result is voted by the multiple trees [23]. After the k-means clustering in the previous step, normal events and abnormal events of the KDD test set dataset will be classified by RF, and these abnormal events will be further subdivided. Random forest train a set of decision trees separately, so the training can be done in parallel on Spark. These datasets are also loaded and parsed as RDDs. It is necessary to set the number and the depth of trees reasonably. The number of trees is 450 and the maximum depth is 20. The classification decision method of the RF algorithm is described as Formula 2, where x and Y represent a single classification sample and classification target, respectively, $h_i(x)$ refers to a classification result, $P(h_{\mathrm{i}}(x) = Y)$ is the representation function, and $H(X)$ presents the classification result of the RF algorithm.

$$H(X) = \arg\max_Y \sum_{i=1}^{k} P(h_i(x) = Y) \tag{2}$$

### C. DEEP LEARNING ALGORITHMS FOR MULTI-TARGET ANOMALY CLASSIFICATION IN THIRD STEP

In the third stage, CNN, LSTM, and CNN combined with LSTM are used to complete multi-target anomaly classification. CNN generally consist of an input layer, convolutional layers, pooling layers, fully connected layers, and others. They are feedforward neural networks that replace traditional matrix multiplication with a convolution. Each convolution actually extracts specific features, and the convolution kernels perform multiple convolutions with the original dataset to extract key hidden features. CNN is divided into Conv1D, Conv2D, and Conv3D according to the input data type. Different dimension convolution layers have different application scenarios. Conv1D is generally used in serialization models, such as natural language processing. In this paper, attack events can be divided into normal events and attack events, which are similar to the positive and negative sentiments in sentiment analysis of natural language processing. Thus, Conv1D can be used as the intrusion detection model. The forward propagation of the CNN convolution layer can be calculated as shown in Formula 3, where $\sigma$ is the activation function. $\sigma$ uses the rectified linear unit (ReLU). The calculation process is described in Formula 4, where $\mathrm{a}^{l-1}$ and $\mathrm{a}^{l}$ represent the input and the output of the L-th neuron, respectively, $W^l$ refers to the L-th layer weight matrix, and $b^l$ presents the offset value of the L-th layer.

$$\mathrm{a}^l = \sigma(\mathrm{a}^{l-1} * W^l + b^l) \tag{3}$$

$$\text{Relu(a)} = \begin{cases} a, & (a > 0) \\ 0, & (a <= 1) \end{cases} \tag{4}$$

LSTM is designed to solve the problem of apparent gradient disappearance and explosion of RNNs. Compared with RNN, LSTM introduces a forget gate and a cell state. By filtering the past state, LSTM can select which state has more influence on the current cell state instead of directly selecting the nearest state like RNN. LSTM is generally suitable for dealing with problems that are positively related to time series, such as machine translation, conversation generation, encoding, decoding, and others. Because there is an absolute correlation before and after the intrusion detection event, this paper considers using the LSTM model. A standard LSTM unit is comprised of a cell, an input gate, an output gate, and a forget gate. In the three gates, $x_t$ represents the input vector, $h_{t-1}$ means the output value of the last state, $W$ and $b$ are used to describe the weight matrix and the bias vector respectively. The cell state is directly related to the forget gate and to the input gate, and the current cell state and output gate directly affect the final output. The three gates are defined as follows:

$$i_\text{t} = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

$$\text{f}_\text{t} = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6}$$

$$o_\text{t} = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

## III. PROPOSED INTRUSION DETECTION MODEL

Based on the ideas above, this paper considers using deep learning algorithms combined with machine learning algorithms to classify intrusion events. Simultaneously, machine learning algorithms are parallelized on the Spark platform to improve the speed of data preprocessing and training.

### A. DATASET DESCRIPTION AND PREPROCESSING

The NSL-KDD [24] and the CIC-IDS2017 datasets are used to verify the effectiveness of the model proposed in this paper. The NSL-KDD dataset is further processed and organized in the KDD CUP 99 [25] dataset. The CIC-IDS2017 dataset was generated in 2017 [26] by the Communications Security Establishment and the Canadian Institute for Cybersecurity. The NSL-KDD dataset has one label and 41 features that are either continuous or symbol. The input of the intrusion detection model must be numeric, and features with character values cannot participate in the calculations. The CIC-IDS2017 dataset used in this paper is in CSV format and contains only the continuous features. Some features in the NSL-KDD and CIC-IDS2017 datasets have large variances. The classification results tend to be influenced by the features with large variances. Therefore, it is necessary to normalize the input sample dataset.

#### 1) DIGITAL PROCESSING OF CATEGORICAL FEATURES

Protocol_type, service, and flag are categorial features in the NSL-KDD dataset. These categorial features need to be converted into numeric values. For example, values of the feature of protocol_type includes TCP, UDP, and ICMP, which can be converted into ordinal numbers 1, 2, and 3. In this example, the categorical feature service is converted to numerical values in the range of [1,70] with a step size of 1. The values of the categorical feature flag are converted to numeric values in the range of [1,11] with a step size of 1. These numbers will be further processed by the one-hot method in the next step.

#### 2) NORMALIZATION OF DIGITAL FEATURES

The purpose of normalization is to simplify the calculation method, that is, to transform the dimensional expression into nondimensional expression. Finally, preprocessing limits the data to specific ranges of values to eliminate the adverse effects caused by variance deviation. In this paper, all characteristic values are mapped to the range [0,1], and min-max normalization is used, where $\text{x}_{m,n}$ represents the nth feature of the m-th data, $Max_{:,n}$ means the maximum value of the nth feature, $Min_{:,n}$ is used to describe the minimum value of the nth feature, and min-max normalization is defined as follows:

$$\text{x}_{m,n} = \frac{\text{x}_{m,n} - Min_{:,n}}{Max_{:,n} - Min_{:,n}} \tag{8}$$

#### 3) LABEL ONE-HOT PROCESSING

The labels of the NSL-KDD and CIC-IDS2017 are discrete values, and one-hot encoding of the labels can be used to represent categorical variables in binary vectors. In the NSL-KDD dataset, DoS represents denial-of-service, R2L stands for unauthorized access from a remote machine, U2R means unauthorized access to local superuser privileges, Probing represents surveillance and other probing. In the CIC-IDS2017 dataset, the labels are of eight types, including DoS, DDoS, Infiltration, Brute Force, PortScan, Botnet and Web Attack. DDoS represents distributed denial-of-service. Infiltration is the use of vulnerable software into the internet from inside. Brute Force includes Brute Force FTP and Brute Force SSH. PortScan means using tools to scan ports and services. Botnet refers to the use of many devices connected to the Internet, each of which runs one or more bots to perform various tasks. Web Attack consist of cross-site scripting (XSS) and SQL injection. This paper performs one-hot encoding for the digitized categorical features and the labels of multi-target anomaly classification stage.

### B. BINARY CLASSIFICATION STAGE BASED ON DISTRIBUTED MACHINE LEARNING

The second stage is for quickly classifying attack events while ensuring the accuracy of attack events. If too many attack events are misjudged, the IDS false negative rate will increase, and the border protection of IDS cannot be effective. Therefore, the focus of this stage is to separate the attacks as quickly as reasonably possible. The improvement of IDS accuracy depends on the validity of the data. The data collected by IDS in the future will be diverse, not only in network traffic and host logs but also in firewall logs, user behavior information, intelligence information, and others. Only by

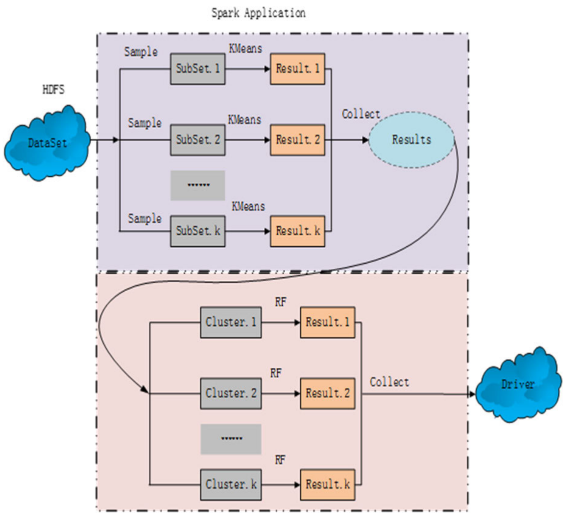


**FIGURE 2.** Distributed binary classification model.

considering this diversity can we conduct a multiscale and multidimensional analysis of security events. In the face of these massive data of different dimensions, IDS needs to filter traffic in advance before subdividing attack types. The attack traffic is filtered out by removing normal traffic, which reduces the difficulty and time cost of in-depth processing of attack traffic in the later stage. The classification method at this stage is shown in Figure 2.

The distributed intrusion detection binary classification model integrates the k-means algorithm and RF algorithm based on Spark. The k-means algorithm is used to cluster intrusion detection events, and RF is used to classify clustering results. The preprocessed dataset is divided into different subsets according to the configuration of Spark and the size of the input dataset. Each subset exists in the form of RDD in Spark. Each transform operation and action operation in the algorithm will eventually be applied to each RDD. Therefore, the dataset in Figure 2 is divided into multiple subsets and clusters. In the whole process, the k-means algorithm is first used to cluster the original dataset to obtain clusters. Then the RDD operator aggregation function is used to aggregate the clustering results into final results. For result.1, result.2, result.3 and other clustering results, different strategies are adopted according to the number of clustering points. Clusters with no more than 20 clustering points are directly judged as abnormal. Otherwise, the RF algorithm is used to classify the clustering results. Based on these two strategies, abnormal events and normal events based on binary classification can be obtained. The system flow diagram of the k-means and the RF algorithms can be seen in Figure 3. The threshold in Figure 3 is 20 as mentioned above.

### C. MULTI-TARGET ANOMALY CLASSIFICATION STAGE BASED ON DEEP LEARNING ALGORITHM

In the third stage, the events judged as normal by the RF algorithm in the second stage exit the model directly. Most of the attack events are successfully separated, and the events determined to be abnormal will enter the third stage.

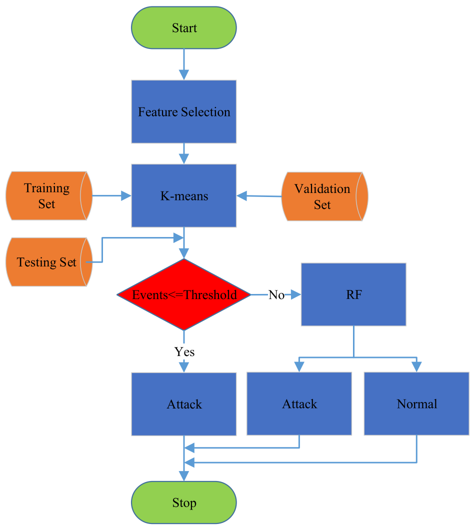
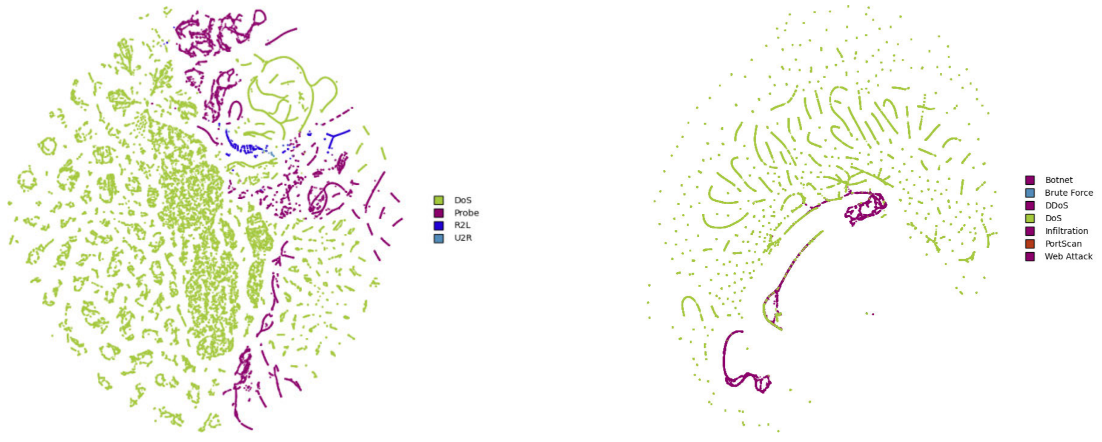


**FIGURE 3.** Overall flow of the second stage.

#### 1) TYPE IMBALANCE PROBLEM

The distribution of the training sets of the NSL-KDD and CIS-IDS2017 datasets is shown in figure 4. The visualization effect diagram is displayed in two dimensions using t-SNE, which was proposed by Policar *et al.* [27] for reducing high-dimensional data. The proportion of different attack sample types varies greatly, and training sets have an obvious data imbalance problem. In the NSL-KDD dataset, there are only a few of the types R2L and U2R. In the CIS-IDS-2017 dataset, DDoS and PortScan are more numerous. This phenomenon will cause the classification model to be more biased. Frequent attack events, that is, U2R and R2L events in the NSL-KDD dataset, are easily misclassified as DoS and Probe events with large sample sizes. Oversampling or undersampling facilitates reducing the polarization trend of the classification results. In this paper, the ADASYN oversampling algorithm is used to solve the problem of data imbalance. ADASYN can adaptively synthesize minority samples according to the distribution of minority samples. By calculating the probability distribution of minority classes, the number of minority classes to be synthesized can be determined. Finally, a new minority class sample is generated according to Formula 9, where $X_i$ is a minority class sample and $\hat{X}_i$ represents a random selection from the k nearest neighbors of $X_i$, $\lambda \in [0, 1]$. The distribution of attack samples synthesized by the ADASYN algorithm is shown in Figure 5 and Figure 6. It can be seen that the minority class sample has been made equal to the majority class sample, and the imbalance of the NSL-KDD and CIC-IDS2017 datasets has been alleviated.

$$X_{\text{new}} = X_i + (\hat{X}_i - X_i) \times \lambda \tag{9}$$



(a)

(b)

**FIGURE 4. The t-SNE visualization of attack events: (a) the t-SNE visualization of NSL-KDD; (b) the t-SNE visualization of CIC-IDS2017.**

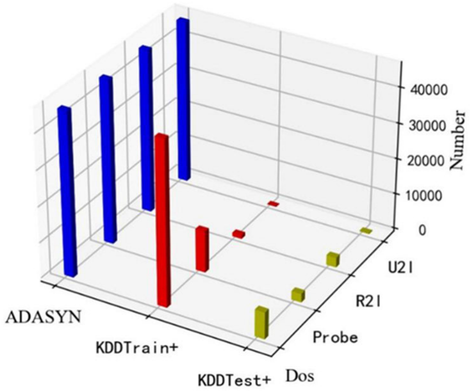


**FIGURE 5. The sample distribution of NSL-KDD dataset before and after ADASYN sampling.**

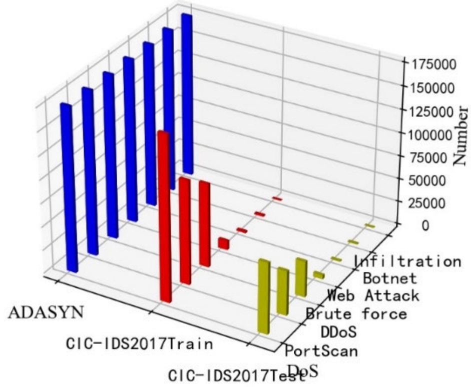


**FIGURE 6. The sample distribution of CIC-IDS2017 dataset before and after ADASYN sampling.**

### 2) INTRUSION DETECTION DEEP LEARNING MODEL FRAMEWORK

In this stage, CNN, LSTM, and CNN+LSTM are used to verify the model's performance. Each model network is set with a hidden layer, two hidden layers, and three hidden layers. The middle layer's activation function is ReLU, and the last layer of the model is fully connected with four neurons. The corresponding activation function is Softmax. These three frameworks are used to explore the most suitable multi-target anomaly classification method. Finally, all attack events are classified by the Softmax classifier. The overall structure of this stage is shown in Figure 7. After the final preprocessing, the NSL-KDD dataset has 115 features, and the CIC-IDS2017 dataset has 70 features. The number of dense blocks at the last layer is set to 4 in the NSL-KDD dataset and 7 in the CIC-IDS2017 dataset.

### 3) SELF-INCREASING ID

Attack events classified from test sets in the second stage need to be extracted in the third stage. The method adopted is to add a unique ID to test sets. By matching with the ID of the attack events in the second stage, all events classified as attack events in the KDD test set are extracted. The dataframe operator of the Spark application has a monodynamically_increasing_id method, which can add a different ID for each row of the dataframe. However, if the number of rows in the dataset is too large, multiple partitions will be generated. The sequencing of ID values is not based on the actual row number because of the partitions. To solve this problem, a custom method is used to an increment in the ID number. The detailed steps are described in Table 1.

## IV. INTRUSION DETECTION EXPERIMENTAL DESIGN

The master node and worker nodes of the cluster are located on the local Huawei private cloud. The flexibility of cloud resources to expand is convenient for the experiment. Nodes can be replicated quickly through cloning technology, so the hardware configurations of Spark computing nodes can be the same. Hardware configurations of the driver, the master node, and a single worker node are described in Table 2. The driver

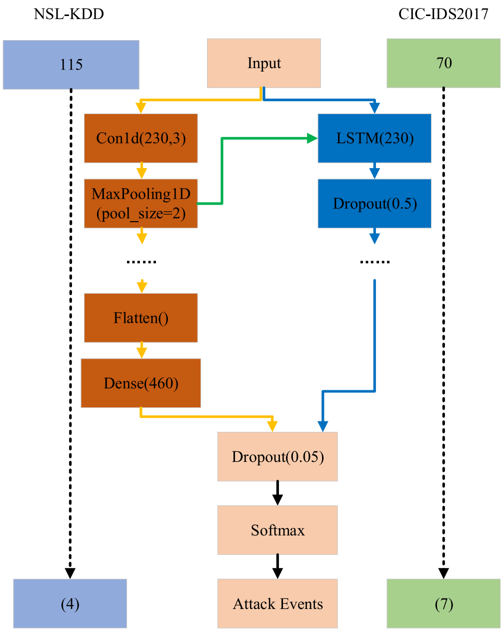


**FIGURE 7.** **Overall structure of the third stage.**

**TABLE 1.** **The self-increasing ID algorithm.**

```
Algorithm Self-increasing ID
Input: inputDF: Samples in Dataframe format with multiple fields.
Output: outPutDFWithId: Samples in Dataframe format with unique ID.
Step 1: create a StructField ← StructField( “id” , LongType(), True);
Step.2: create a StructType according to structFieldId and inputDF ’ s field for sche ← StructType([structFieldId]+ inputDF.schema.fields);
Step 3: convert inputDF in DataFrame format to RDD format;
Step 4: add indexes to inputDF using zipWithIndex function like this:
    for each row in inputDF do
      add one column to row with i;
      i++;
    end for
Step 5: extract all fields from inputDF into newRDD←inputDF. map(lambda args: ([args[1] + offset] + list(args[0])));
Step 6: get outPutDFWithId ← spark.createDataFrame(newRDD, schema);
Step 7: return outPutDFWithId;
```

node is a high configuration server. In addition to creating SparkContext and running Spark’s main program, it is also an important experimental environment for the third stage. The driver is configured with an RTX 4000 GPU, which supports TensorFlow as the backend framework of Keras. This allow us to speed up the training time of the neural network model. After the binary classifications by the machine learning algorithms on the Spark platform, the normal and abnormal events are directly collected by the distributed storage system, HDFS, to the driver side. In the third stage, the intrusion detection data stored on the driver side can be used for classification directly.

### A. EVALUATING INDICATOR

There are many evaluation metrics for detection results of abnormal events, such as accuracy, recall which is also called TPR. The proportion of each sample in the test sets may vary greatly, and the final prediction results may be biased to items with more sample labels. Therefore, the accuracy rate alone is not scientific and reasonable. Each prediction result generated by the prediction model can be divided into four types shown in Table 3.

A true positive (TP) is observed when the model correctly predicts the attack events, and a true negative (TN) is observed when the model correctly predicts normal events. A false positive (FP) is observed when the model incorrectly predicts the positive events and a false negative (FN) is observed when the model incorrectly predicts negative events.

Then AC is the ratio of samples that are correctly predicted in test sets to the total samples. AC in this paper is applicable to normal events and all abnormal events in the test set.

Then AC is the ratio of samples that are correctly predicted in test sets to the total samples. AC in this paper is applicable to normal events and all abnormal events in the test set.

$$AC = \frac{TP + TN}{TP + FN + FP + TN} \tag{10}$$

TPR is the ratio of samples labeled as attacks that are correctly predicted to be attacked in test sets to all incidents labeled as attacks.

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

TT is the training time of the intrusion detection model to train data sets.

$$TT = Training\_DataSet \tag{12}$$

PT is the prediction time of the intrusion detection model to predict test sets.

$$PT = \Pr edicting\_TestData \tag{13}$$

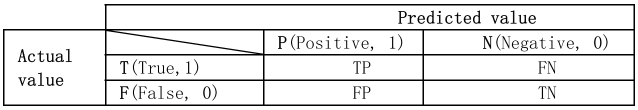
### B. RESULTS OF THE BINARY CLASSIFICATION STAGE

After the classification by the distributed machine learning binary classification model, the confusion matrices over the test sets are shown in Figure 8 and Figure 9. The detection rate in NSL-KDD dataset is 98.57%, and the detection rate in the CIC-IDS2017 dataset is 99.67%, which indicates that most of the attack events can be detected accurately. At the same time, the false alarm rate of attack events is very low. The AC of the NSL-KDD dataset reaches 92.90%, while the AC of

**TABLE 2.** Experimental environment configuration.

| Name | Number of CPU cores | Graphics card | RAM | Disk capacity | Operating System | Spark | Scala | Keras | Tensorflow |
|---|---|---|---|---|---|---|---|---|---|
| Driver Node | 12 | RTX 4000 | 64G | 1T | Ubuntu 16.04 | 2.3.1 | 2.11.8 | 2.4.3 | 2.3.1 |
| Master Node | 10 | None | 19G | 50G | Ubuntu 16.04 | 2.3.1 | 2.11.8 | 2.4.3 | 2.3.1 |
| Worker Node | 10 | None | 16G | 50G | Ubuntu 20.04 | 2.3.1 | 2.11.8 | 2.4.3 | 2.3.1 |

**TABLE 3.** Experimental environment configuration.



| | | Predicted value | |
|---|---|---|---|
| | | P(Positive, 1) | N(Negative, 0) |
| Actual value | T(True, 1) | TP | FN |
| | F(False, 0) | FP | TN |

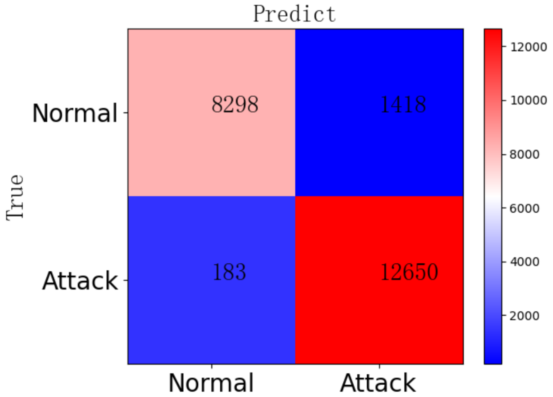


**FIGURE 8.** The confusion matrix of NSL-KDD.

the CIC-IDS2017 dataset is up to 99.87%. This suggests that the performance of the binary classification model is good. The higher detection rate and lower false alarm rate can meet the needs of further classification of attack events in the third stage.

### C. THE RESULTS OF MULTI-TARGET ANOMALY CLASSIFICATION

The multi-target anomaly classification is based on three deep learning frameworks: CNN, LSTM, and CNN+LSTM. The hyperparameters of the deep learning model directly affect the classification results. After many experiments, the filter of the CNN model is set to 230, the kernel_size to 3, and pool_size to 2. The unit hyperparameter of the LSTM model is set to 230. The hyperparameter of the CNN+LSTM model is set according to the previous two models. The initial learning rate is set to 0.01. The Adam algorithm is used for optimizers. The separation of normal events in the second stage can reduce the number of training iterations in the third stage. Therefore, the total number of training iterations is set to 100.

The accuracy and loss of three deep learning models with three hidden layers are shown in Figure 10, Figure 11, and Figure 12. The three models can achieve high training accuracy and small loss. By observing (a) and (b), it is found that the number of iterations required to classify only attack events requires about 50 epochs, which is significantly reduced compared with the 1000 iterations needed by Vinayakumar *et al.* [28] to classify normal events and four types of attack events. The early separation of normal samples saves unnecessary learning time for the multi-target abnormal classification stage based on deep learning algorithms. Thus the overall training time can be significantly reduced. The ROC curve of the KDD test set in three models with three hidden layers is shown in Figure 13. It can be seen that the ROC curve of the CNN model fits better. The AUC value of DoS, Probe, and R2L is more than 90%, which indicates that the model has a good effect on the classification of these three types.

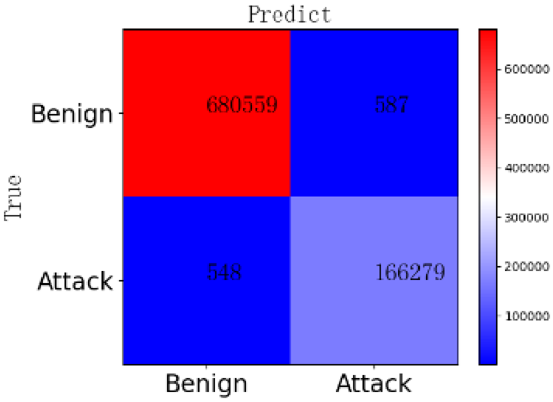
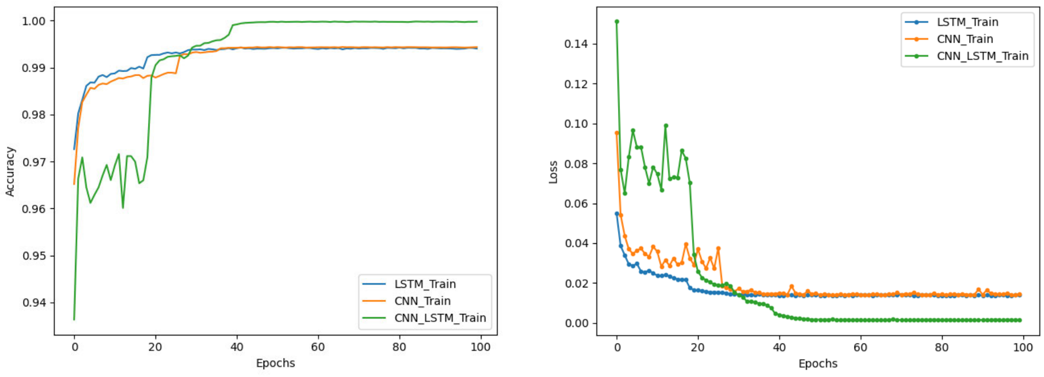


**FIGURE 9.** The confusion matrix of CIC-IDS2017.

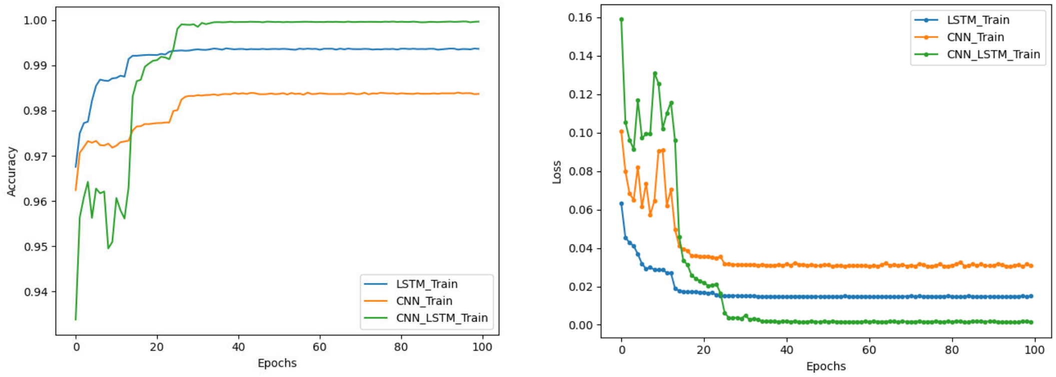
The TPR of anomaly events is shown in Table 4 and Table 5. In the NSL-KDD dataset, an intrusion detection method proposed by Pajouh *et al.* [29] is relatively balanced. The TPR of DoS and Probe in the CNN model with two hidden layers is above 80%, and the R2L attack events obtain the highest recall rate. Compared with the model proposed by Pajouh, the TPRs of DoS, Probe, and R2L are significantly improved in this paper, with the TPR of R2L as high as 73.84%. From the overall perspective, except for the U2R recall rate at only 25.79%, the other three types of attack events have a relatively balanced TPR as a whole. Table 5 shows that other common attack types have the best TPR in the proposed methods in the CIC-IDS2017 dataset, except for Web Attack. In particular, the TPR of Infiltration and Botnet reach to 100%. This means that Infiltration and Botnet can be predicted with complete accuracy.



(**a**) Accuracy

(**b**) Loss

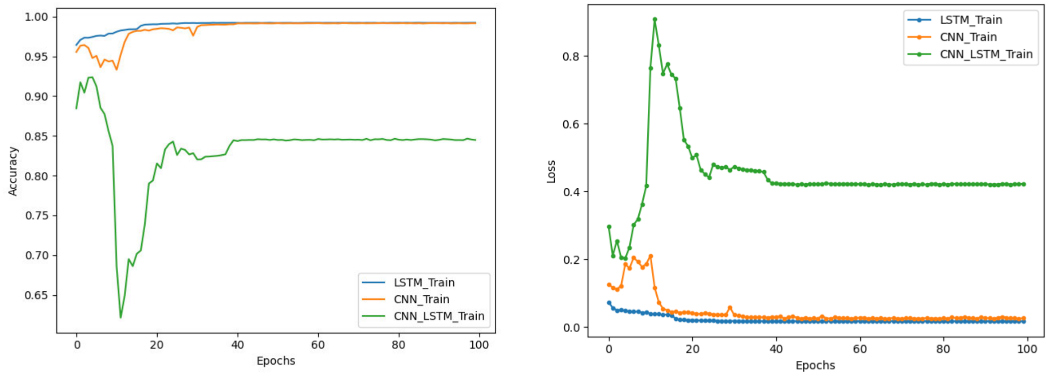
**FIGURE 10. Accuracy and loss of three deep learning models with one hidden layer in the NSL-KDD dataset.**



(**a**) Accuracy

(**b**) Loss

**FIGURE 11. Accuracy and loss of three deep learning models with two hidden layers in the NSL-KDD dataset.**

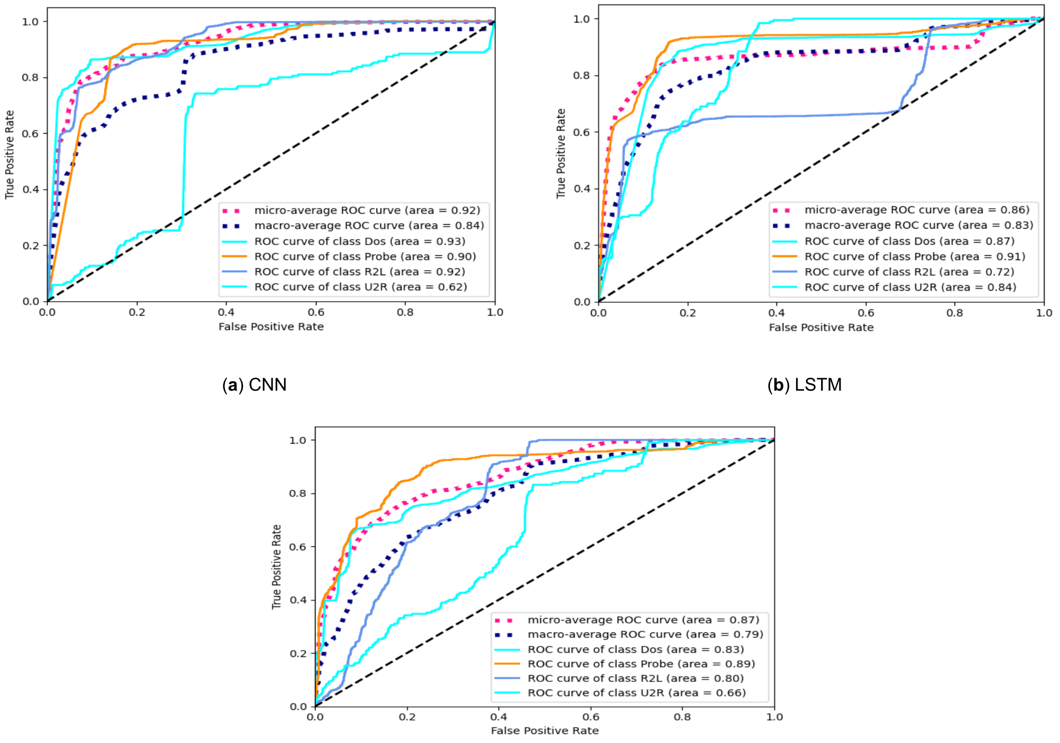


(**a**) Accuracy

(**b**) Loss

**FIGURE 12. Accuracy and loss of three deep learning models with three hidden layers in the NSL-KDD dataset.**

The accuracy of the proposed model and other deep learning algorithm models is shown in Figure 14 and Figure 15. The accuracy is based on binary classification in the second stage and multi-objective anomaly classification in the third stage. By combining the classification results of these two stages, the numbers of normal events and different attack events to the total number of samples in the test sets are finally obtained as measures of overall accuracy. So it is an overall accuracy. The algorithms used on the NSL-KDD dataset by other models include MLP [6], DNN [7], CNN [9], Deep-MLP [37], STL-IDS [38] and DIS-IDS [39]. Figure 14 shows that the accuracy of the 5-classification of most models is approximately 80% on the KDD test set. The main reason is that the sample type of the KDD test set is not completely



(**a**) CNN

(**b**) LSTM

(**c**) CNN+LSTM

**FIGURE 13.** ROC curve of three deep learning models with three hidden layer in the NSL-KDD dataset.

**TABLE 4.** Comparison of the TPR results of the 4 classification experiments on the NSL-KDD dataset (%).

| Method | DoS | Probe | R2L | U2R |
|---|---|---|---|---|
| TFTC[29] | 88.20 | 87.32 | 42.00 | 70.15 |
| MDPCA-DBNs[30] | 81.09 | 73.94 | 17.00 | 6.50 |
| Siam-IDS[31] | 85.37 | 48.66 | 33.25 | 56.72 |
| I-ELM[32] | 76.37 | 83.36 | 32.35 | 11.00 |
| HFR-MLR[33] | 83.94 | 85.17 | 38.53 | 7.00 |
| RF[35] | 75.80 | 64.80 | 52.20 | 4.90 |
| DNN[34] | 0.764 | 0.663 | 0.672 | 0.242 |
| Proposed | 90.42 | 91.53 | 73.84 | 25.79 |

**TABLE 5.** Comparison of the TPR results of the 7 classification experiments on the CIC-IDS2017 dataset (%).

| Method | DoS | DDoS | Brute Force | Port Scan | Botnet | Infiltration | Web Attack |
|---|---|---|---|---|---|---|---|
| AD-H1CD[35] | 99.63 | 99.86 | -- | 99.82 | 94.48 | -- | 87.38 |
| GA-BPNN[36] | -- | -- | 97.89 | 96.01 | 97.50 | 96.40 | 97.85 |
| Proposed | 99.96 | 99.95 | 99.69 | 99.94 | 100 | 100 | 94.49 |

consistent with the sample type of the KDD training set. The accuracy of the DIS-IDS model proposed by Haggag M in 2020 is 83.57%, while the model proposed in this paper achieves the highest accuracy at 85.24%. In the CIC-IDS2017 dataset, the test set is a sample from the training set. Therefore, the accuracy of the 8-classification is higher than that of the KDD test set. The accuracy over the CIC-IDS2017 dataset is more than 90%. The other algorithms include DeepDetect [40], a deep learning combination method [41] and AD-H1CD [35]. In Figure 15, the model proposed in this paper achieves the highest accuracy of 99.91%. This proves that the method proposed in this paper also has a good performance on the CIC-IDS2017 dataset.

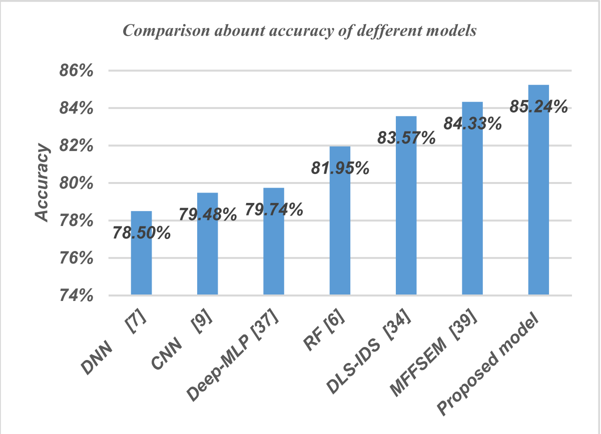


**FIGURE 14.** Comparison of accuracy of the proposed model with other deep learning models on the NSL-KDD dataset.

## D. THE RESULTS OF TIME COST

In this paper, TT and PT includes the time spent in the second and third phases. In the second classification stage, the Spark platform can speed up the processing and training

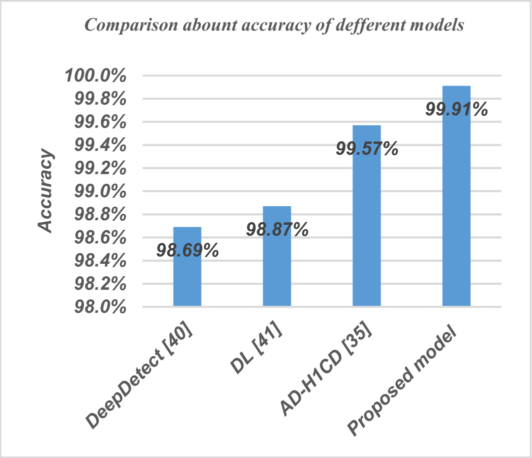


**FIGURE 15.** Comparison of accuracy of the proposed model with other deep learning models on the CIC-IDS2017 dataset.

**TABLE 6.** Time cost of different numbers of worker nodes in the second stage (s).

| The number of worker node | NSL-KDD | | CIC-IDS2017 | |
|---|---|---|---|---|
| | TT | PT | TT | PT |
| One | 181.29 | 15.60 | 2862.45 | 241.55 |
| Two | 170.00 | 10.58 | 2250.46 | 238.56 |
| Three | 164.53 | 10.39 | 2206.85 | 194.67 |

by expanding the number of worker nodes and adding more partitions. At the same time, increasing the total executor cores and executor memory size of Spark can speed up the calculation time for each partition. In addition, this paper uses a feature selection algorithm to select some features with a more significant effect [42], which significantly reduces the computing time cost caused by the rapid expansion of features from one-hot encoding. To compare the influence of the different numbers of worker nodes in the second stage, we set three kinds of worker nodes. The detailed result is shown in Table 6. As we can see, as the number of nodes increases, the values of TT and PT decrease. According to the characteristics of Spark, it is predicted that the reduction of TT and PT will be more obvious as the training set gets larger.

A direct comparison of time consumption can be problematic because these methods are implemented on different hardware platforms. Therefore, it might be better to explain the potential for shorter time through a variety of tips. The third stage of the experiment is based on the fact that test sets are divided into attack events and normal events in the second stage. The data classified as attack events in the second stage have been collected locally on the driver side. The deep learning classification model in the third stage is based on the driver side, so the dataset does not have to be transformed repeatedly on different platforms. Spark's distributed platform also completed the preprocessing data required by the training set in the third stage, saving preprocessing time during this stage. The classification model uses TensorFlow, which supports GPU acceleration for training, so the time cost of the third stage can be further controlled.

## V. CONCLUSION AND FUTURE WORK

This paper proposed a cascading intrusion detection algorithm based on distributed k-means, RF, and deep learning to solve IDS problems that were long time consuming and low efficiency. The distributed computing platform was used to achieve fast preprocessing of the intrusion detection dataset. The attack events and normal events were separated by integrating the distributed k-means and the RF algorithm. Then, based on isolated attack events, various deep learning models were designed to learn the hidden features of attack events. Finally, the classification of different attack events was realized quickly. The classification model proposed in this paper was evaluated through the NSL-KDD dataset and CIC-IDS2017 dataset. The experimental results show that the proposed method in this paper can effectively realize the classification of attack events. The overall accuracy of normal events and the other four types of attack events is 85.24% in the NSL-KDD dataset, while the overall accuracy of benign events and the other seven types of attack events reaches 99.91% in the CIC-IDS2017 dataset. Furthermore, compared with the traditional standalone deep learning intrusion detection models, the method proposed in this paper had better TPR for most attack events regardless of the sample number of these attack events, which meant that more attacks could be accurately identified. The proposed method had faster data preprocessing speed and potentially less training time. Whether the Spark parameters are properly set may affect the data uniformity of the partitions, which has a great impact on TT. In the future, we will consider how to deploy the model proposed in this paper in an actual application environment to verify the effectiveness of the proposed model in the paper.

## REFERENCES

[1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[2] S. J. Yang, M. Berndl, D. M. Ando, M. Barch, A. Narayanaswamy, E. Christiansen, S. Hoyer, C. Roat, J. Hung, C. T. Rueden, A. Shankar, S. Finkbeiner, and P. Nelson, "Assessing microscope image focus quality with deep learning," *BMC Bioinf.*, vol. 19, no. 1, pp. 77–85, Dec. 2018.

[3] X.-S. Wei, C.-W. Xie, J. Wu, and C. Shen, "Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization," *Pattern Recognit.*, vol. 76, pp. 704–714, Apr. 2018.

[4] C. Meng and X. Zhao, "Webcam-based eye movement analysis using CNN," *IEEE Access*, vol. 5, pp. 19581–19587, 2017.

[5] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "CNN for situations understanding based on sentiment analysis of Twitter data," *Procedia Comput. Sci.*, vol. 111, pp. 376–381, Jan. 2017.

[6] S. K. Dey, M. R. Uddin, and M. M. Rahman, "Performance analysis of SDN-based intrusion detection model with feature selection approach," in *Proc. Int. Joint Conf. Comput. Intell.* Singapore: Springer, 2020, pp. 483–494.

[7] P. Dahiya and D. K. Srivastava, "Network intrusion detection in big dataset using spark," *Procedia Comput. Sci.*, vol. 132, pp. 253–262, Jan. 2018.

[8] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.

[9] A. H. Mirza and S. Cosan, "Computer network intrusion detection using sequential LSTM neural networks autoencoders," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, Lzmir, Turkey, May 2018, pp. 1–4.

[10] A. Shrivastava and R. R. Ahirwal, "A SVM and K-means clustering based fast and efficient intrusion detection system," *Int. J. Comput. Appl.*, vol. 72, no. 6, pp. 25–29, Jun. 2013.

[11] A. D. Sagale and S. G. Kale, "Hybrid model for intrusion detection using naive Bayesian and support vector machine," *Int. J. Comput. Technol.*, vol. 1, no. 3, pp. 56–59, Jan. 2014.

[12] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3357–3368, Apr. 2019.

[13] L. A. Maglaras and J. Jiang, "A novel intrusion detection method based on OCSVM and K-means recursive clustering," *EAI Endorsed Trans. Secur. Saf.*, vol. 2, no. 3, p. e5, Jan. 2015.

[14] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Udupi, India, Sep. 2017, pp. 1222–1228.

[15] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, Dec. 2018.

[16] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Z. Yang, "Automatic device classification from network traffic streams of Internet of Things," in *Proc. IEEE 43rd Conf. Local Comput. Netw. (LCN)*, Chicago, IL, USA, Oct. 2018, pp. 1–9.

[17] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Soft Computing in Industrial Applications*. Berlin, Germany: Springer, 2011, pp. 293–303.

[18] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *Int. J. Inf. Secur.*, vol. 3, pp. 1–17, Jun. 2020.

[19] C. A. de Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral, and G. D. S. Vieira, "Hybrid approach to intrusion detection in fog-based IoT environments," *Comput. Netw.*, vol. 180, Oct. 2020, Art. no. 107417.

[20] M. Kulariya, P. Saraf, R. Ranjan, and G. P. Gupta, "Performance analysis of network intrusion detection schemes using apache spark," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Melmaruvathur, India, Apr. 2016, pp. 1973–1977.

[21] M. A. Khan, M. R. Karim, and Y. Kim, "A two-stage big data analytics framework with real world applications using spark machine learning and long short-term memory network," *Symmetry*, vol. 10, no. 10, p. 485, Oct. 2018.

[22] M. Khan, M. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol. 11, no. 4, p. 583, Apr. 2019.

[23] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[24] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.

[25] K. Siddique, Z. Akhtar, F. A. Khan, and Y. Kim, "KDD cup 99 data sets: A perspective on the role of data sets in network intrusion detection research," *Computer*, vol. 52, no. 2, pp. 41–51, Feb. 2019.

[26] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Lisbon, Portugal, Jan. 2019, pp. 108–116.

[27] P. G. Policar, M. Strazar, and B. Zupan, "openTSNE: A modular Python library for t-SNE dimensionality reduction and embedding," *BioRxiv*, Aug. 2019. [Online]. Available: https://www.biorxiv.org/content/biorxiv/early/2019/08/13/731877.full.pdf, doi: 10.1101/731877.

[28] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "A comparative analysis of deep learning approaches for network intrusion detection systems (N-IDSs): Deep learning for N-IDSs," *Int. J. Digit. Crime Forensics*, vol. 11, no. 3, pp. 65–89, Jul. 2019.

[29] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K.-R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 2, pp. 314–323, Apr. 2019.

[30] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," *Appl. Sci.*, vol. 9, no. 2, pp. 238–262, Jan. 2019.

[31] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network," *Procedia Comput. Sci.*, vol. 171, pp. 780–789, Jan. 2020.

[32] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, Feb. 2017.

[33] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *J. Inf. Secur. Appl.*, vol. 58, pp. 102804–102818, Mar. 2021.

[34] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, Apr. 2019.

[35] C. Ma, X. Du, and L. Cao, "Analysis of multi-types of flow features based on hybrid neural network for improving network anomaly detection," *IEEE Access*, vol. 7, pp. 148363–148380, Oct. 2019.

[36] S. Manimurugan, P. Manimegalai, P. Valsalan, J. Krishnadas, and C. Y. Narmatha, "Intrusion detection in cloud environment using hybrid genetic algorithm and back propagation neural network," *Int. J. Commun. Syst.*, pp. 1–15, Nov. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4667

[37] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A deep learning approach for network intrusion detection based on NSL-KDD dataset," in *Proc. IEEE 13th Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Xiamen, China, Oct. 2019, pp. 41–45.

[38] H. Zhang, J.-L. Li, X.-M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection," *Future Gener. Comput. Syst.*, vol. 122, pp. 130–143, Sep. 2021.

[39] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a deep learning model for intrusion detection on apache spark platform," *IEEE Access*, vol. 8, pp. 163660–163672, Aug. 2020.

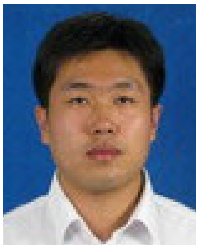
[40] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, "DeepDetect: Detection of distributed denial of service attacks using deep learning," *Comput. J.*, vol. 63, no. 7, pp. 983–994, Jul. 2020.

[41] A. Pektaş and T. Acarman, "A deep learning method to detect network intrusion through flow-based features," *Int. J. Netw. Manage.*, vol. 29, no. 3, p. e2050, May/Jun. 2019, doi: 10.1002/nem.2050.

[42] H. S. Chae and S. Choi, "Feature selection for efficient intrusion detection using attribute ratio," *Int. J. Comput. Commun.*, vol. 8, pp. 134–139, 2014. [Online]. Available: https://www.naun.org/cms.action?id=7623

**CHAO LIU** received the B.E. and M.E. degrees in electronic communication engineering from the Civil Aviation University of China (CAUC), China, in 2015 and 2018, respectively. He is currently a Teaching Assistant with CAUC. His research interests include computer network security and big data.

**ZHAOJUN GU** received the M.E. degree from the Harbin Institute of Technology, in 1996, and the Ph.D. degree from Nankai University, in 2004. He is currently a Professor with the College of Computer Science and Technology, Civil Aviation University of China, where he focuses on information systems and security. His research interests include networks, information security, and civil aviation information systems.



**JIALIANG WANG** received the Ph.D. degree from Northeastern University (HEU), in 2014. He is currently a Lecturer with the College of Computer Science and Technology, Civil Aviation University of China. His research interests include civil aviation information systems and UAV security.