IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 26, 2019, accepted March 8, 2019, date of publication March 12, 2019, date of current version April 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904620

An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks

YIHAN XIAO[1], (Member, IEEE), CHENG XING[1], TAINING ZHANG[2], AND ZHONGKAI ZHAO[1]
[1]College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[2]Department of Electrical and Computer Engineering, School of Engineering, Tufts University, Medford, MA 02155, USA

Corresponding author: Zhongkai Zhao (zhaozhongkai@hrbeu.edu.cn)

ABSTRACT With the popularity and development of network technology and the Internet, intrusion detection systems (IDSs), which can identify attacks, have been developed. Traditional intrusion detection algorithms typically employ mining association rules to identify intrusion behaviors. However, they fail to fully extract the characteristic information of user behaviors and encounter various problems, such as high false alarm rate (FAR), poor generalization capability, and poor timeliness. In this paper, we propose a network intrusion detection model based on a convolutional neural network–IDS (CNN–IDS). Redundant and irrelevant features in the network traffic data are first removed using different dimensionality reduction methods. Features of the dimensionality reduction data are automatically extracted using the CNN, and more effective information for identifying intrusion is extracted by supervised learning. To reduce the computational cost, we convert the original traffic vector format into an image format and use a standard KDD-CUP99 dataset to evaluate the performance of the proposed CNN model. The experimental results indicate that the AC, FAR, and timeliness of the CNN–IDS model are higher than those of traditional algorithms. Therefore, the model we propose has not only research significance but also practical value.

INDEX TERMS Communication technology, convolutional neural network, data dimensionality reduction, intrusion detection.

I. INTRODUCTION

The rapid development of physical-layer wireless communication technology also brings security problems. Among them, wireless eavesdropping, identity spoofing, information tampering and other security issues frequently occur, causing trouble for people using wireless communication networks [1]–[3] With the development of Internet technology, an increasing number of physical devices become connected to the network [4]. The connection between devices leads to a large number of data being generated and saved. The era of ''big data'' emerges over time [5]–[7]. Owing to the complexity of the network system and the richness of attack methods, network attacks occur constantly. Thus, network attack detection means should be more intelligent and efficient than before to prevent the growing hacker technology. Given the limitations of traditional network security protection technology, establishing a stable, reliable, and accurate intrusion detection model has broad application prospects for improving network security. Therefore, intrusion detection systems (IDS) [8] have gradually drawn interest in network security technology. Currently, one of the widely known intrusion detection methods is error rate reduction using different machine-learning techniques. The most widely studied algorithms include support-vector machines (SVM) [9], [10], neural networks [11]–[14], and clustering algorithms. Chung constructed an intrusion detection model by combining various machine-learning algorithms, such as SVM, Bayesian classification, and decision trees [15]. Pan *et al.* [16] and Tahir *et al.* [17] proposed a hybrid machine-learning technology combining K-means and SVM to detect attacks. Shin used k-means clustering to calculate the similarity between data and adjust the parameters, which can detect denial-of-service attacks and worm attacks simultaneously [18]. Beqiri [19] applied intrusion detection with neural networks. Zhao [20] proposed the Least Squares SVM (LSSVM) model for network intrusion detection. Jha *et al.* [21] used the hidden Markov model to study network intrusion detection. Bamakan [22] used K-class support vector classification (KSVC) to classify

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

network intrusion. Meanwhile, traditional machine learning is classified as shallow learning, which has a desirable effect when the number of labeled data samples is small. However, with the continuous expansion of network data, a large number of high-dimensional and nonlinear labeled network data result in challenges to intrusion detection, prompting studies on deep learning. Alom and Bontupalli [23] used deep belief networks and extreme learning machines to conduct intrusion detection, improving detection accuracy. Kim *et al.* [24] used long short-term memory structures combined with recurrent neural networks to effectively train the intrusion detection model. Chuanlong *et al.* [25] used the recurrent neural network (RNN)–IDS, enhancing the accuracy of intrusion detection and exceeding the performance of traditional machine learning dual-classification and multi-classification methods. This technique provides a research method for network intrusion detection.

Convolutional neural networks (CNNs) [26], as the focus of deep-learning pattern recognition research, have achieved excellent research results in computer vision, speech recognition, and natural language processing. It can learn better features automatically than traditional feature selection algorithms. In fact, the structure of packets and traffic is very similar to words, sentences, and articles compared to bytes in a network stream. Therefore, CNNs can not only select features but also classify the traffic data. The more traffic data, the more useful features the CNN can learn, the better classification the CNN performs. Hence, CNN is suitable for the massive network environment. Besides, compared with other DL algorithms, the greatest advantage of CNN is that it shares the same convolutional kernels, which would reduce the number of parameters and calculation amount of training once greatly, it can more quickly identify attack type of traffic data.

While the network traffic generated in network communication has a data format of one-dimensional byte stream, the input data type suitable for CNN is two-dimensional. Considering the lack of compatibility between the distinct network structure of the CNN and the network data of IDS, this study applies data preprocessing methods to remove the redundant and irrelevant features in the network traffic data. Next, traffic is transformed into a two-dimensional matrix form, which can be used in the proposed CNN network. The method not only solves the problem preventing traditional machine learning models from determining the relationship between data features; it can more clearly elucidate the features compared with the general neural network. Finally, the feasibility and efficiency of the proposed method are verified, facilitating advances in CNN and network intrusion detection.

The rest of this paper is organized as follows. Section II briefly introduces principal component analysis (PCA), autoencoder (AE), and the CNNs. Section III gives the overall algorithm flow. Then Section IV is the experiment and the result. Finally, we draw conclusions and future research directions in Section V.

II. BACKGROUND

A. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is the most commonly used method for linear dimension reduction in machine learning. It is widely used in data analysis and preprocessing. PCA aims to map high-dimensional data to a low-dimensional space representation by linear projection. To reduce the dimension of initial variables while retaining the variance in these samples as much as possible, many highly correlated variables can be transformed into independent or unrelated variables.

Suppose that a dataset has m objects x_1, x_2, x_m, and each object contains n variables. To obtain the n′(n′ < n) principal components, the process is based on the following steps:
Step 1: Standardization of raw data.

$$Z_{ij} = \frac{x_{ij} - \overline{x_j}}{s_j}, \quad i = 1, 2, \ldots, \ m; j = 1, 2, \cdots, n \tag{1}$$

where $\overline{x_j} = \frac{\sum_{i=1}^{m} x_{ij}}{m}$, $s_j^2 = \frac{\sum_{i=1}^{m}(x_{ij}-\overline{x_j})^2}{m-1}$
Step 2: Finding the correlation coefficient matrix.

$$\mathrm{R} = \frac{Z^T Z}{m-1} \tag{2}$$

Step 3: Finding the eigenvalues $a_1 \geq a_2 \geq \ldots a_\mathrm{n}$ of R and the corresponding unit eigenvector.

$$a_1 = \begin{Bmatrix} \alpha_{11} \\ \alpha_{21} \\ \vdots \\ \alpha_{n1} \end{Bmatrix}, \quad a_2 = \begin{Bmatrix} \alpha_{12} \\ \alpha_{22} \\ \vdots \\ \alpha_{n2} \end{Bmatrix}, \cdots, \ a_n = \begin{Bmatrix} \alpha_{1n} \\ \alpha_{2n} \\ \vdots \\ \alpha_{nn} \end{Bmatrix} \tag{3}$$

Step 4: Calculating the principal components.

$$t_i = \alpha_{1i} Z_1 + \alpha_{2i} Z_2 + \ldots + \alpha_{ni} Z_n, \quad i = 1, \ \cdots, \ n' \tag{4}$$

The principal component n′(n′ < n) is used as a new data vector to replace the original data. After feature extraction, the unimportant and redundant feature can be removed to the greatest extent.

B. AUTO-ENCODER

The auto-encoder (AE) network is an effective nonlinear dimension reduction method proposed by Chicco *et al.* [27]. This method is incorporated into intrusion detection in the current study [28]. The technique uses several hidden layers of neural networks to transform the input high-dimensional datasets nonlinearly and maps the original high-dimensional features in unsupervised learning. By projecting low-dimensional features and reconstructing these low-dimensional features into high-dimensional features, the feature dimension of datasets can be effectively reduced to the greatest extent to ensure the integrity of feature information.

Suppose that the encoder input is I, the hidden layer is Z, and the output is O. The AE aims to achieve $I \approx 0$. This is also the origin of its name. Although this operation seems meaningless, the purpose of this deep structure is not its final output but the coding in the hidden layer. By decoding these data through different feature representations, the original data can be obtained. The data after encoding by the AE must be the most important one to represent the input vector. The AE is expressed as follows:

Given an input data, set the input to I. The AE obtains

$$Z = f(I) = f_l(Wx + b_i) \tag{5}$$

f_I is the activation function of the decoder, and the output of the decoder is as follows:

$$O = g(Z) = g_Z(W^T y + b_Z) \tag{6}$$

where g_Z is the activation function of the decoder, W is the initial weight of the network, b_l is the forward bias, and b_Z is the reverse bias. The training process of the AE is that the training parameters $\{W, b_l, b_Z\}$ minimize the reconstruction error of the neural network. The reconstruction error is expressed as follows:

$$E = \sum_{x \in I} J(x, g(f(x))) \tag{7}$$

In the aforementioned expression, J is a reconstruction error function. Generally, it can be the mean square error loss function or the cross-entropy loss function.

C. CONVOLUTIONAL NEURAL NETWORK

The CNN consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. The CNNs of different structures have varying numbers of convolution layers and pooling layers. Assume that the input characteristic of the CNN is feature map of the layer i is $M_i(M_0 = X)$. Then, the convolution process can be expressed as

$$M_i = f(M_{i-1} \otimes W_i + b_i) \tag{8}$$

where W_l is the convolution kernel weight vector of the i layer; the operation symbol " $\otimes$ " represents the convolution operation; b_i is the offset vector of the i layer; and $f(x)$ is the activation function. The convolutional layer extracts different feature information of the data matrix M_{i-1} by specifying different window values, and extracts different features M_i in the data through different convolution kernels. In the convolution operation, the same convolution kernel follows the principle of "parameter sharing"—that is, sharing the same weight and offset—which markedly reduces the number of parameters of the entire neural network. The pooling layer usually samples the feature map in accordance with different sampling rules after the convolutional layer. Suppose M_i is the input to the pooling layer and M_{i+1} is the output of the pooling layer; then, the pooling layer can be expressed as

$$M_{i+1} = subsampling(M_i) \tag{9}$$

The sampling criterion generally selects the maximum or mean value of the window region. The pooling layer mainly reduces the dimension of the feature, thereby decreasing the influence of redundant features on the model. One typical CNN model, Lenet-5, is presented as follows:

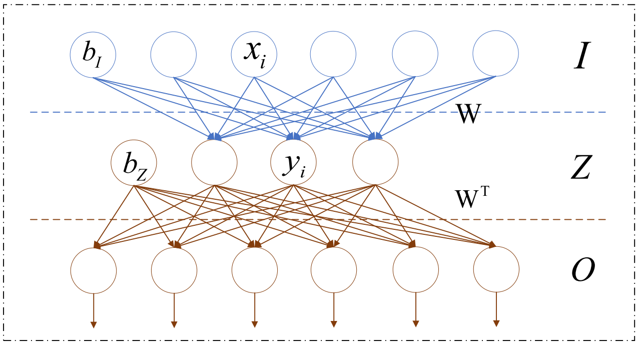


FIGURE 1. Principle diagram of the automatic encoder.

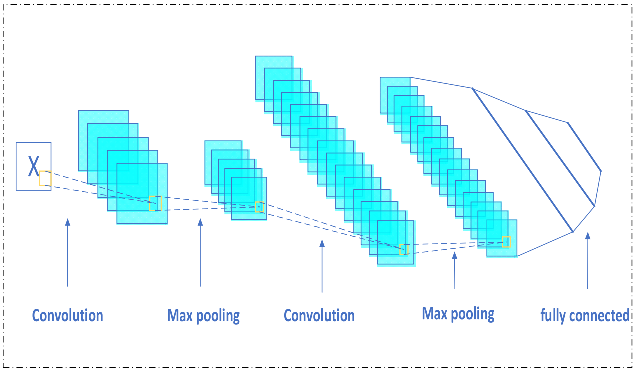


FIGURE 2. Lenet-5 network architecture.

III. PROPOSED INTRUSION DETECTION SYSTEM

The overall framework of the model presented in figure 3 consists of three steps.

STEP 1: Data preprocessing and data type conversion. The symbolic characteristic attributes in KDD datasets are digitized and normalized to obtain standardized datasets. After the standardized datasets undergo dimensionality reduction, each network dataset is converted into a two-dimensional dataset, which conforms to the input data form of the CNN. The data are then read into the model by using a data reading tool referred to as pandas.

STEP 2: Concrete structure of the CNN intrusion detection model. After training the transformed dataset with the CNN, the optimal features are obtained. Five attack states in the dataset are identified using the Softmax classifier. These attack states include DOS, Probe (Supervisor and Other Detection Activities), R2L (Illegal Access to Local Super User Privileges by Ordinary Users), U2R (Illegal Access from Remote Machines), and Normal (Normal Records).

STEP 3: Model training and reverse fine-tuning improve the performance of the model. In the CNN model, the back propagation (BP) algorithm fine-tunes the parameters of the network model. After the optimal parameters of the network model are determined, the performance of the model is evaluated by the classification results of the test dataset.

A. DATASET

KDDcup99 [29], which contains 4,898,431 traffic data, is a recognized dataset in the field of intrusion detection.

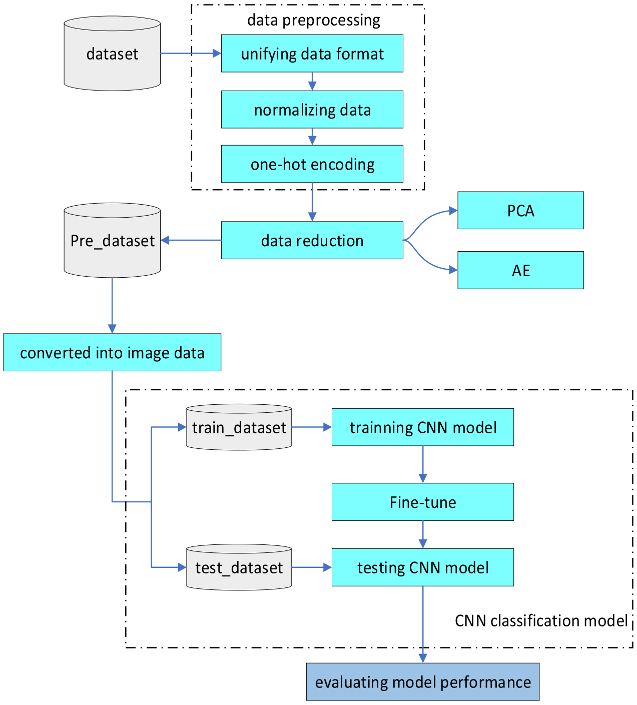


FIGURE 3. Proposed framework for intrusion detection.

Constructed by Stolfo *et al.* based on the data captured in the DARPA 1998 IDS evaluation program, it is divided into labeled training data and unlabeled test data, and unlabeled test data contains attack data types that do not appear in the training set. Attack data types that do not appear in the training set render the dataset verifiable and realistic. Each dataset contains 41 features, such as protocol type and service type. It is divided into four categories: basic characteristics of TCP connection, content characteristics of TCP connection, statistical characteristics of network traffic based on time and statistical characteristics of network traffic based on host. The 41 features are roughly divided into four aspects: one is the basic properties of the network connection, a total of nine features, such as the type of protocol, the number of bytes, etc.; the second is some content attributes of the network connection, a total of 13 features, The main purpose is to detect attacks that are not always present and are more concealed, such as U2R and R2L. These two types of attacks are hidden in the data payload of the packet. From the outside, this type of packet is no different from the normal packet. Therefore, the researchers extracted features such as the number of login failures that reflect the intrusion behavior to characterize such attacks; the third is the behavioral statistical properties of the network connection, describing the statistics of a behavior over a continuous period of time; Attributes that describe the relevance of some past behaviors. Each training data is assigned a label to indicate that the data belongs to. ''Normal'' or ''abnormal'' and the category of abnormal is indicated. The training dataset has 1 normal identification type and 22 training attack types. The entire dataset contains 39 attacks, and the rest of the attacks appear in the test dataset. These attacks can be divided into four categories: denial of service attacks (DOS), remote to local (R2L), user to remote (U2R) and probing.

B. DATA PREPROCESSING

Each connection record of the KDDcup99 dataset consists of 41 features, 38 of which are digital features and 3 are symbolic features. The dataset processing methods are listed below.

Step 1: Symbolic feature numerization

Attribute mapping transforms symbolic features into numerical data. For example, the second column has three values−tcp, udp, and icmp–that encode their characteristics as binary vectors [1, 0, 0], [0, 1, 0], [0, 0, 1], respectively. By analogy, the 41-feature dataset becomes a 122-feature dataset after transformation.

Step 2: Normalization

By analysis, we find a large between the data in the KDD99 dataset. For example, in the aforementioned five records, the 6-feature dataset becomes 10 times larger than the 5-feature dataset on the average, or greater. This result induces the model to think that the 6-feature dataset is far more important than the 5-feature dataset, which may not be the case in practice. To eliminate the large difference in feature values caused by variation in dimensions, the numerical data have to be normalized. We use min-max standardization to map data to [0, 1] without disrupting the linear relationship between the original data. The min-max standardization formula is as follows:

$$\mathrm{y} = \frac{y - MIN}{MAX - MIN} \tag{10}$$

where y is the attribute value, MIN is the minimum of the attribute, and MAX is the maximum of the attribute.

Step 3: Label numerization

Class identification of the label record is numerically processed, with 0 for Dos, 1 for Normal, 2 for Probe, 3 for R2L, and 4 for U2R, thereby facilitating the one-hot processing of the label in subsequent training and testing.

C. DATA DIMENSIONALITY REDUCTION

Data dimensionality reduction is part of data preprocessing in the entire intrusion detection system. Various studies have shown that not only serious redundancy among the characteristic dimensions of network data but also high correlation exists among the data of each dimension. Redundancy and correlation between feature dimensions not only reduce the response time of the intrusion detection system but also affect the learning efficiency of the training process. Therefore, dimensionality reduction of high-dimensional data is particularly necessary. Reducing the dimension of the dataset can not only improve the learning performance of the detection system; it can also reduce the redundancy of the dataset.

The initial data format of the KDD99 dataset is a 1*122 dimension vector after pretreatment; thus, to facilitate

TABLE 1. Number of types in KDDcup99 dataset.

KDD	All types	Normal	Dos	Probe	U2R	R2L
10%KDD(Train)	494021	97278	391458	4107	52	1126
Corrected(Test)	311029	60593	229853	4166	228	16189

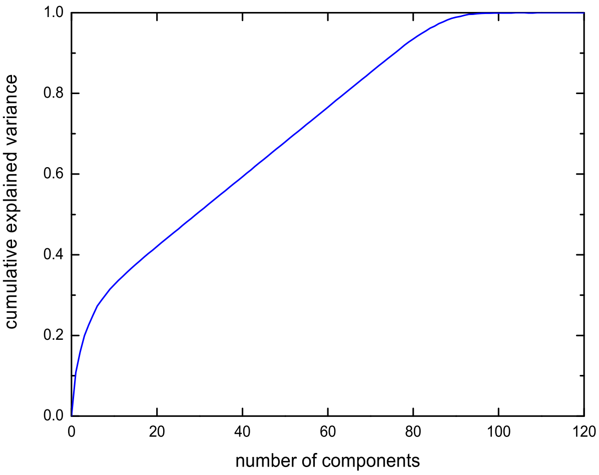


FIGURE 4. Proportion of cumulative dimension variance to total variance.

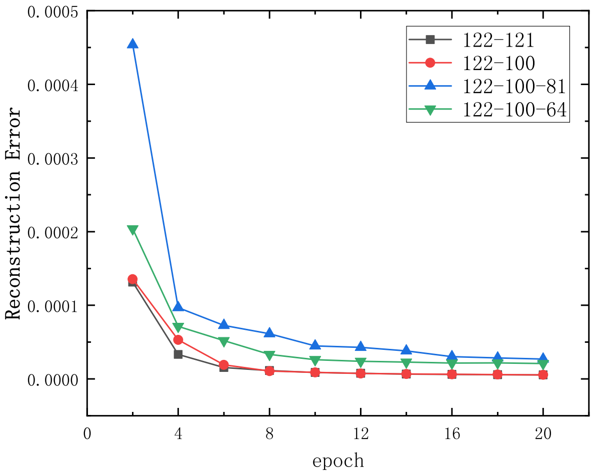


FIGURE 5. Reconstruction error using the optimal structure of the auto-encoder.

the subsequent convolution calculation in order to extract features, a one-dimensional network connection dataset needs to be mapped into two-dimensional feature vectors (1*122 vector converted to n*n image data). By so doing, 1*121 or 1*100 dimension reduction vectors can be transformed into 11*11 or 10*10 matrices. The transformed 2-dimensional network connection characteristic matrix can be used as an input sample of the CNN input layer. In this study, PCA and AE are used to reduce the dimensionality of features.

1) PCA for dimensionality reduction

PCA in Scikit–Learn library [30] is applied to analyze the variance ratio of each principal component after PCA transformation. Experiments in figure 4 show that the first 91 principal components can represent 99% of the pre-processed data. Therefore, to retain the effective information in the data, the dimension reduction parameter n_components should range from 91 and 122. Dimension reduction vectors need to meet the requirements of input vector dimension in the CNN network; thus, n-components = 100/121 is designated as the number of feature dimensions after dimensionality reduction with PCA. Finally, the reduced dimension datasets consisting of 100 effective features or 121 effective features are generated to be used in the subsequent network intrusion detection model.

2) Dimensionality reduction with an auto-encoder

To satisfy the requirements of an input vector dimension in the CNN network, the number *m* of hidden layer neurons of the AE network should satisfy m = n*n. To maximize the retention of information in the data, *m* may be assigned the values of 64, 81, 100, and 121. When m = 64, the structural model of the self-coding network is shown in figure 5.

To evaluate the effect of dimensionality reduction with AE, the concept of reconstruction error is introduced. A reconstruction error refers to the error obtained from the output and input values of feature reconstruction after dimensionality reduction. In dimensionality reduction with AE, whether the data after dimensionality reduction can be well restored to the original input data is determined. The reconstruction error is then expressed using the mean squared error (MSE), as follows:

$$MSE = \|x - x'\|^2 = \|x - \sigma'(W'(\sigma(Wx + \mathrm{b})) + b')\|^2 \quad (11)$$

In the experiment, the connection of the neural network is established using the fully connected layer. The number of iteration epochs is 20, and the maximum batch_size is 128 for each training. To optimize the structure of the AE network, parameters, such as the number of layers of the neural network, number of neurons in each layer, activation function, and optimization function are adjusted to observe the change in the reconstruction error during iteration of the AE network. The optimal structure of the AE network is finally determined. Comparison of the changes in reconstruction error indicates that the activation function of neurons uses the Relu function and that the optimization function uses the Adam optimizer. In different dimensions, the optimal number of network layers and neurons for AE networks is shown in figure 5.

After the structure of the self-encoder is determined, the reconfiguration error of the network is calculated by inputting the pre-processed network traffic data. By adjusting the network parameters with the BP algorithm, the network traffic data subjected to dimensionality reduction can finally be obtained through the middle layer.

D. CNN INTRUSION DETECTION MODEL

In this study, we propose CNN-IDS, an intrusion detection algorithm based on the typical CNN model Lenet-5 [31]. By improving its network structure and applying batch normalization (BN) optimization, this algorithm can be effectively used to detect network intrusion data.

The network structure and parameter setting of the CNN–IDS model we proposed are shown in Table 4. It mainly includes three steps to identify network intrusion data.

STEP 1. Modeling network connection data by the CNN.

First, the reduced dimension network traffic dataset is mapped into two-dimensional feature vectors. The transformed two-dimensional network connection feature matrix can be used as input samples of the CNN input layer. Second, the label is processed as one-hot encoding, which is convenient for training and testing.

STEP 2. Using the CNN to extract and analyze network traffic characteristics.

The CNN–IDS model consists of an input layer, an output layer, and five hidden layers. The input layer maps a one-dimensional network dataset into two-dimensional plane information, facilitating CNN feature learning. The implied layer includes a convolution layer and a pooling layer. The convolution layer maps the sample data to the high-dimensional space continuously and learns the feature information of the network connection data. The pooling layer reduces the computation and improves the detection efficiency of the model by reducing the dimension of the extracted features. In the Lenet-5 intrusion detection architecture, each convolution layer and each pooling layer are set alternately to accurately and efficiently extract the intrusion characteristics. The output layer maps the result of feature extraction to a one-dimensional array to predict classification.

STEP 3. Softmax classifier and CNN are combined to output the classification results—that is, the detection results of the intrusion behavior are obtained.

Aimed at achieving distinct network traffic data, this study proposes the following enhancements to the training and optimization of the CNN–IDS model:

1) In the convolution layer, the BN algorithm [32] is used to increase the network learning rate.

BN is a training optimization method proposed by Google. The batch represents the batch of data, and normalization is the standardization of data. Adding the BN algorithm to the convolution layer can accelerate the learning rate of the network structure. The flowchart of the BN algorithm is thus presented.

Input: x_1, x_2, x_m (these are the data ready to enter the activation function) a) find the average value of the data x_1, x_2, x_m to enter into the activation function:

$$u_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{12}$$

b) after averaging, the variance of each number $x_1, x_2, \cdots, x_m$ is calculated:

$$\sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - u_\beta)^2 \tag{13}$$

c) the variance is calculated and the data are standardized:

$$\hat{x}_i = \frac{x_i - u_\beta}{\sqrt{\sigma_\beta^2 + \in}} \tag{14}$$

d) training parameters $2'$ and β

e) output y is obtained by linear transformation of γ and β.

$$y_i = \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{15}$$

In forward propagation, the current output remains unchanged, and only l' and β are recorded. In reverse propagation, the learning rate is calculated to change the weight according to the chain derivative of γ and β.

2) A dropout layer is added in the middle of the fully connected layer.

Owing to over-fitting, the classification ability of the BP neural network is limited. Dropout can efficiently solve this problem. Dropout, a mechanism to improve the performance of the BP neural network by preventing the interaction of feature detectors, can efficiently solve this problem. This technique was proposed by Hinton *et al.* [33]. During model training, the dropout randomly sets the weight of the hidden layer nodes of the network temporarily to 0, and does not participate in the calculation of the network, but the weight remains. Therefore, a neural network with n nodes and dropout can be regarded as a set of 2^n models; however, the number of parameters to be trained at this time is unchanged, which can also solve the time-consuming problem. In dropout the meaning of the parameter P is, dropout with probability P abandon neurons and other neurons to make probability 1-P reserved. Each neuron is the same probability of being closed.

IV. SIMULATION

A. EVALUATION INDICATORS

The evaluation index of the CNN-IDS network intrusion detection model mainly includes three indicators: accuracy (AC), detection rate (DR), and false alarm rate (FAR). In the specific detection results, T (True) and F (False) represent the data classified correctly or incorrectly, respectively. P (Positive) and N (Negative) represent the prediction results of the detection system as abnormal or normal data, respectively. All data in the dataset must be categorized into any of the four categories: TP, TN, FP, and FN. Only TP indicates that the classification result of the system consists of anomalous attack data, and the classification result is correct; TN indicates that the classification result of the system is positive and correct; FP indicates that the system predicts the data as

TABLE 2. Classification of intrusion detection model prediction results.

	Classification result	
Label Attributes	Normal	Attack
Normal	TN	FP
Attack	FN	TP

anomalous attack data, but the classification result is wrong; FN indicates that the system predicts the data as constant data, but the classification result is wrong. Specifically, it can be clearly expressed in the table below.

AC is the probability that the total number of samples correctly classified by the system accounts for the total number of samples.

$$AC = \frac{TP + TN}{FN + TP + FP + TN} \tag{16}$$

DR indicates the probability that the system can correct alarms to account for the total amount of abnormal network connection data when an attack is present in the system environment.

$$DR = \frac{TP}{FN + TP} \tag{17}$$

FAR is the probability that the system misjudges the normal data as the attack data threatening the system and sends a false alarm. FAR is the probability that this part accounts for all the normal network connection data.

$$FAR = \frac{FP}{TN + FP} \tag{18}$$

B. EXPERIMENTAL PROCESS

The simulation system environment is shown in Table 3.

The experimental training set is 10% KDDCUP99 dataset, and the test dataset is the corrected dataset. First, according to the dataset processing flow of Section 3, datasets are pre-processed and transformed by dimensionality reduction. Datasets using dimension reduction with PCA are categorized into (PCA (100), PCA (121). By analogy, datasets using dimensionality reduction with AE can categorized into AE (121), AE (100), AE (81), and AE (64). Subsequently, we use the aforementioned datasets to evaluate the proposed CNN–IDS model.

The parameter settings of each network layer in the improved intrusion detection model based on Lenet-5 are listed in Table 5. In addition to the parameters of the convolution and pooling cores in each layer of the model, the convolution layer uses the method of same padding. In the last classification layer, the model is set up as five output neurons, which are classified by calling the Softmax function; that is, the corresponding digital label of the coding position of the most probable rate in the five outputs. The cross_entropy function is used to solve the error loss. The Adam optimization algorithm is used to reduce the error. The learning rate is set to 1e-4. The weights and biases of all layers are initialized by the 0-mean Gaussian function. The data distribution in the dataset is uneven; thus, the BN layer is used before convolution to increase the learning rate of the network structure. To prevent the network from falling into over-fitting and to improve the generalization capability of the network structure, dropout is used in the second pooling layer and the fully connected layer, while the optimal P value is 0.3 after the experiment of the connection probability P between layers. In addition, when training the model, the number of iteration epochs is set to 50, and batch_size is set to 128. After model training is completed, the model is evaluated according to the classification of network traffic in the test dataset.

TABLE 3. Experimental environment.

Project	Environment/Version
Operating system	Windows10
CPU	i7-8700k
Memory	32GB
GPU	GTX1080ti
Framework	Keras1.2

TABLE 4. convolutional neural network–intrusion detection system detection performance on different datasets.

Metrics / Datasets	AC	DR	FAR
PCA(100)	0.930	0.916	0.006
PCA(121)	0.926	0.910	0.015
AE(64)	0.927	0.928	0.027
AE(81)	0.933	0.924	0.008
AE(100)	0.940	0.930	0.005
AE(121)	0.922	0.917	0.024

C. EXPERIMENTAL RESULTS AND ANALYSIS

First, we use different dimensionality reduction datasets to evaluate the performance of the CNN–IDS model. The model uses training data to train and verifies the training results on the test set. AC, DR, and FAR are used to measure the performance of the model. The final experimental results are listed in Table 4.

The experimental results show that the CNN–IDS model proposed in this study efficiently detects network intrusion data by dimensionality reduction. AC, DR, and FAR can reach 94.0%, 93.0%, and 0.5%. Thus, the detection performance of dimensionality reduction with PCA and that with AE only slightly vary. In the listed datasets, AE (100) detection reaches the highest accuracy of 94.0%, whereas that of PCA (100) is higher. This result indicates that the low-dimension feature dataset after dimensionality reduction

TABLE 5. Convolutional neural network–intrusion detection system network structure and parameter settings.

Layer	Type	Kernel/Pool_size	Strides	Active Function	Output
L1	input	-	-	-	10*10*1
L2	Conv	2*2	1	BN+ Relu	10*10*8
L3	Maxpool	2*2	1	Relu	9*9*8
L4	Conv	2*2	1	Relu	9*9*16
L5	Maxpool	2*2	1	Relu + Dropout	8*8*16
L6	FC	-	-	Relu + Dropout	64
L7	FC	-	-	Softmax	5

TABLE 6. Confusion matrix of convolutional neural network–intrusion detection system.

Predicted class Actual class	Normal	Probe	Dos	U2R	R2L	total
Normal	60276	198	104	3	12	60593
Probe	126	3480	488	0	72	4166
Dos	4287	79	225487	0	0	229853
U2R	66	100	7	47	8	228
R2L	13053	4	58	5	3069	16189

realizes redundancy removal in network traffic and obtains better detection results for the CNN–IDS model.

We select the dataset with the optimal effect, and the classification result draws the confusion matrix. The confusion matrix in Table 6 shows that the detection accuracy of the model is as high as 94.0%, but the detection rates of U2R and R2L are considerably low at 20.61% and 18.96%, respectively.

The detection accuracy of these two attacks is one of the main factors restricting the overall detection accuracy. One reason is that the number distribution of various types of attacks in datasets is considerably unbalanced. U2R and R2L attacks in the training and test sets have very small amount of data. Thus, deep structure has a limited ability to learn these two types of attacks.

In the second group of experiments, we use 6 traditional machine learning algorithms (Naive Bayes, Logistic Regression, Decision Tree, Random Forest, SVM, Adaboost) to implement network intrusion detection on PCA (100) datasets. The evaluation index used is the same as the previous one, and the results of the test are compared with the CNN–IDS model proposed in this study, as shown in figure 6. Compared with AC, DR, and FAR, the classification results of the CNN–IDS are better than those of the machine learning algorithm. Thus, the learning ability of the shallow neural network is limited relative to the deep structure of the CNN. The CNN detection model proposed in this study has a better effect.

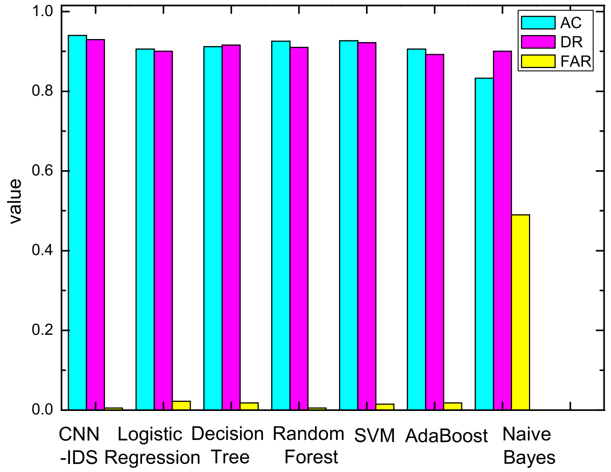


FIGURE 6. Performance comparison between the machine learning model and the convolutional neural network–intrusion detection system model.

Finally, we compare the detection performance of the CNN–IDS model with those of the DNN and RNN models [34]–[36], all of which use the same dataset AE (100). The DNN uses a three-layer fully connected neural network [37]. The RNN model first uses dimensional transformation to divide the 10*10 matrix into 10 time points, with 10 characteristics at each time point, and then input into the RNN unit for training. The iteration times, batch_size, and optimization function of the three models are all set in the same manner. Comparison of the training time and classification results of

TABLE 7. Comparison time of three models.

time model	training at every epoch	test
CNN–IDS	20s	11s
DNN	26s	15s
RNN	82s	124s

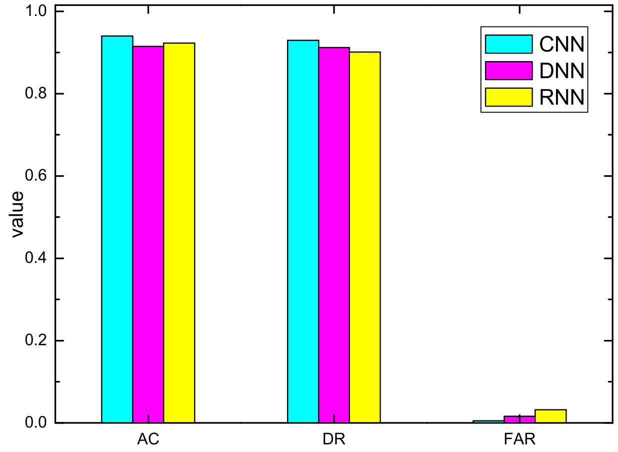


FIGURE 7. Performance comparison of the three deep learning models.

each iteration in Table 7 shows that the CNN–IDS model has a faster training speed than the other two deep learning algorithms. In addition, the detection performance of the model is slightly better, compared with the RNN and DNN models from figure 7.

V. CONCLUSION

As DL algorithms can select features from massive data environment automatically and CNN can share weights, a massive network intrusion detection based on CNN is proposed in this paper. For the dataset to satisfy the CNN requirement with regard to the input data form, dimensionality reduction is used to generate a dataset that can be transformed into a two-dimensional matrix form. The experimental results indicate that the proposed CNN–IDS model not only considerably improves the classification detection performance of the intrusion network traffic but also significantly reduces the classification time, which can satisfy the real-time requirements of the intrusion detection system. In future research, aiming to address the problem of low detection rate and difficulty of feature learning in a small number of attack categories (U2R, R2L), we attempt to generate new sample data by using the generative adversarial network method to identify more features of attack categories.

REFERENCES

[1] Z. Xue, J. Wang, G. Ding, Q. Wu, Y. Lin, and T. A. Tsiftsis, "Device-to-device communications underlying UAV-supported social networking," *IEEE Access*, vol. 6, pp. 34488–34502, 2018.

[2] Z. Zhang, X. Guo, and Y. Lin, "Trust management method of D2D communication based on RF fingerprint identification," *IEEE Access*, vol. 6, pp. 66082–66087, 2018.

[3] Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, "The individual identification method of wireless device based on dimensionality reduction and machine learning," *J. Supercomput.*, vol. 5, pp. 1–18, Dec. 2017.

[4] M. Liu, J. Yang, T. Song, J. Hu, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 641–653, Jan. 2019.

[5] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.

[6] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[7] Y. Li, X. Cheng, and G. Gui, "Co-robust-ADMM-net: Joint ADMM framework and DNN for robust sparse composite regularization," *IEEE Access*, vol. 6, pp. 47943–47952, 2018.

[8] Y. Lin, C. Wang, C. Ma, Z. Dou, and X. Ma, "A new combination method for multisensor conflict information," *J. Supercomput.*, vol. 72, no. 7, pp. 2874–2890, 2016.

[9] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput.*, vol. 18, pp. 178–184, May 2014.

[10] K. V. Narayana, V. V. R. Manoj, and K. Swathi, "Enhanced face recognition based on PCA and SVM," *Int. J. Comput. Appl.*, vol. 117, no. 2, pp. 40–42, 2015.

[11] R. Samrin and D. Vasumathi, "Hybrid weighted K-means clustering and artificial neural network for an anomaly-based network intrusion detection system," *J. Intell. Syst.*, vol. 27, no. 2, pp. 135–147, 2016.

[12] Y. Tu, Y. Lin, J. Wang, and J.-U. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Comput., Mater. Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[13] Y. Lin, C. Wang, J. Wang, and Z. Dou, "A novel dynamic spectrum access framework based on reinforcement learning for cognitive radio sensor networks," *Sensors*, vol. 16, no. 10, pp. 1–22, 2016.

[14] T. Liu, Y. Guan, and Y. Lin, "Research on modulation recognition with ensemble learning," *EURASIP J. Wireless Commun. Netw.*, vol. 1, pp. 179–187, 2017.

[15] S. Chung and K. Kim, "A heuristic approach to enhance the performance of intrusion system using machine learning algoritms," in *Proc. Korea Inst. Inf. Secur. Cryptol. Conf.* Seoul, South Korea: CISC, 2015.

[16] X. Pan, Y. Luo, and Y. Xu, *K-Nearest Neighbor Based Structural Twin Support Vector Machine*. Amsterdam, The Netherlands: Elsevier, 2015.

[17] M. Tahir *et al.*, "Hybrid machine learning technique for intrusion detection system," *Proc. 5th Int. Conf Comout. Inform. (ICOCI)*, Istanbul, Turkey, 2015, pp. 11–13.

[18] D. Shin *et al.*, "Malicious traffic detection using K-means," *J. Korean Inst. Commun. Inf. Sci.*, vol. 41, no. 2, pp. 277–284, 2016.

[19] E. Beqiri, "Neural networks for intrusion detection systems," in *Proc. Int. Conf. Global Secur., Safety, Sustainability*, 2009, vol. 45, no. 3, pp. 156–165.

[20] Z. Fuqun, "Detection method of LSSVM network intrusion based on hybrid kernel function," *Modern Electron. Techn.*, vol. 38, no. 21, pp. 96–99, 2015.

[21] S. Jha, K. Tan, and R. A. Maxion, "Markov chains, classifiers, and intrusion detection," in *Proc. 14th IEEE Comput. Secur. Found. Workshop*, Cape Breton, NS, Canada, Jun. 2001, pp. 206–216.

[22] S. M. H. Bamakan, W. Huadong, and S.Yong, "Ramp loss K-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem," *Knowl. Based Syst.*, vol. 126, no. 10, pp. 113–126, 2017.

[23] A. Zahangir, V. R. Bontupalli, and T. M. Taha "Intrusion detection using deep belief network and extreme learning machine," in *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*. 2016, pp. 357–378.

[24] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2016, pp. 1–5.

[25] Y. Chuanlong, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[27] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," in *Proc. 5th ACM Conf. Bioinf., Comput. Biology, Health Inform.*, 2014, pp. 533–540.

[28] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink MIMO," *IEEE Access*, vol. 7, pp. 7599–7605, 2019. doi: 10.1109/ACCESS.2018.2887308.

[29] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "Analysis and results of the 1999 DARPA Off-Line intrusion detection evaluation" in *Proc. Recent Adv. Intrusion Detection*. Springer, 2000, pp. 162–182.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[31] [Online]. Available: http://www.leiphone.com/news/201406/deep-learning-yann-lecun-facebook.html

[32] S. Ioffe and C. Szegedy "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Mach. Learn. Res.*, 2015, pp. 448–456.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *Comput. Sci.*, vol. 3, no. 4, pp. 212–223, 2012.

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

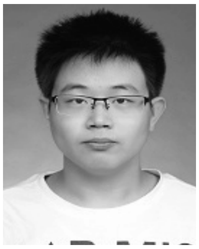
[35] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating effectiveness of shallow and deep networks to intrusion detection system," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2017, 1282-1289.

[36] G. Wang, X. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6225–6232, 2010.

[37] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, Y.-D. Yao, "An amateur drone surveillance system based on the cognitive Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 29–35, Jan. 2018.

YIHAN XIAO received the B.S. degree in electrical and information engineering, the M.S. degree in communication and information systems, and the Ph.D. degree in signal and information processing from Harbin Engineering University, Harbin, China, in 2003, 2008, and 2012, respectively.

She is currently a Teacher with the College of Information and Communication Engineering, Harbin Engineering University. Her research interests include intrusion signal detection, signal feature extraction and analysis, machine learning, and emitter identification.



CHENG XING received the B.S. degree from the Harbin Institute of Technology, in 2016. He is currently pursuing the master's degree with Harbin Engineering University. His current research interests include intrusion detection, machine learning, and data analysis.

TAINING ZHANG received the B.S. degree from the Harbin Institute of Technology, in 2016. She is currently pursuing the M.S. degree with the Department of Electrical and Computer Engineering, Tufts University. Her current research interests include wireless information compression and machine learning.

ZHONGKAI ZHAO received the B.S. degree in electronic engineering, the M.S. degree in signal and information processing, and the Ph.D. degree in communication and information systems from Harbin Engineering University, Harbin, China, in 2002, 2005, and 2010, respectively.

He is currently an Associate Professor with the College of Information and Communication Engineering, Harbin Engineering University. His research interests include machine learning, radar signal processing, radar jamming, and digital communications, with applications to radar and communications.