Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6762175, 6 pages
https://doi.org/10.1155/2022/6762175

*Research Article*

# Optimization of Intrusion Detection System Based on Improved Convolutional Neural Network Algorithm

**Xiaochuan Pu, Yuanqiang Zhang, and Qingqiang Ruan**

*College of Information Engineering, Zunyi Normal University, Zunyi 563002, Guizhou, China*

Correspondence should be addressed to Xiaochuan Pu; jackypc@zync.edu.cn

Received 18 August 2022; Accepted 8 September 2022; Published 25 September 2022

Academic Editor: Zaoli Yang

The commonly used method in network intrusion detection is abnormal behavior detection, but because abnormal behavior detection is based on artificially setting abnormal values for judgment, the efficiency is low and the false alarm rate is high. For this problem, an intrusion detection system architecture combining K-means algorithm and convolutional neural network algorithm has been introduced. First, the data stream is clustered through the K-means algorithm, and the abnormal data is initially separated, and then the data is applied to the convolution. In the neural network algorithm, the intrusion data flow is judged. The experimental results prove that the framework improves the efficiency of the intrusion detection system to a certain extent and effectively improves the detection accuracy.

## 1. Introduction

In today's era of increasingly serious cyber threats, the safe use of computer networks has become an important issue of public research. However, as the network attack methods keep changing, network defense methods, especially intrusion detection systems (NIDS), have to be gradually improved. In this context, it is an urgent requirement for network security to study the detection mechanism of intrusion detection system and improve its performance.

In order to overcome the limitations of rule-based expert systems and statistical methods, many studies have introduced machine learning methods into IDM. The mechanism of intrusion detection system includes Bayesian network, AdaBoost algorithm, decision tree, support vector machine, and convolutional neural network. For example, Shone et al. [1] proposed a Nonsymmetric Deep AutoEncoder (NDAE) based on unsupervised feature learning. The literature [2] proposed the use of hybrid ANN and Kohenen's self-organizing Feature Mapping (SOM) for visual intrusion and the use of recoverable propagation neural network for classified intrusion. The literature [3] uses SOM and radial basis function (RBF) combination method, which has better results than RBF network. In addition, more advanced methods include multiclass SVM [4] and BP Neural Network BPNN [5]. Hu et al. [6] introduced cross-layer aggregation network model based on convolutional neural network and applied the improved convolutional neural algorithm to the intrusion detection framework. The accuracy of the intrusion detection system has improved. Zhifeng et al. [7] improved the initial value $K$ by using the interclass and intraclass dissimilarities and proposed an intrusion detection model based on the improved $K$-means algorithm, which reduced the false positive rate of the intrusion detection system. This combination method can meet the complex and high-dimensional characteristics of the current network. The methods mentioned above perform well in dealing with known network attacks, but they have problems in efficiency and cannot cope with the problem that the system characteristic database needs to be updated frequently.

Although the above research has improved the sample recognition ability and performance, in the case of huge data flow, complex data flow, and specific attacks, there will be problems such as poor accuracy and huge processing resource consumption. This paper proposes an intrusion detection method that combines $K$-means algorithm and convolutional neural network algorithm. Because the $K$-

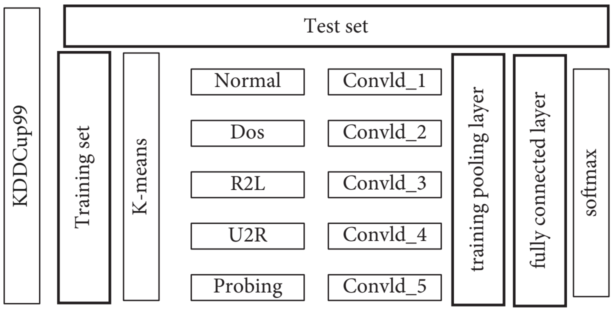

Figure 1: Intrusion detection process.

means algorithm is simple to calculate and has low data requirements, while the convolutional neural network algorithm has a high accuracy rate in multi-classified data, it is expected that the combination of the two algorithms will significantly improve the accuracy and efficiency of intrusion detection. This model is based on the improved convolutional neural network proposed in this paper, combined with the cross-layer design method, and uses the preprocessed original sample data set for model training. And then, through cyclic feature extraction and iterative optimization, the model achieves a good convergence effect. Through the classification test of the trained classifier, the experimental results verify that the method in this paper has a great detection effect.

## 2. Construction of Intrusion Detection System Based on Improved Algorithm

Since intrusion detection is a kind of multiclassification problem, we chose K-means unsupervised clustering algorithm as the basis for subdivision improvement combined with neural network. The design method is mainly based on the characteristics of convolutional neural network to build a multilayer aggregation model, using the sample data set after preliminary screening for model training [8], through cyclic feature extraction and optimization iteration, finally make the convergence effect of the algorithm verified. Through training and testing, we verify that the method in this paper has a good detection effect, and put forward the overall model framework as shown in Figure 1. First of all, the network data is preclustered by the K-means algorithm, and the data set is preclassified according to the type of data stream. Then, enter the preclassified data stream into the convolutional neural network training model by class, and build a multilayer neural network model. Finally, it contains the attribute values and data packets obtained by passing the data through the trained neural classifier.

*2.1. K-Means Clustering Algorithm.* This paper will use the $K$-means clustering algorithm to firstly classify the network data stream into different types of clusters [9], so as to separate the normal data classes and the intrusion data clusters to the greatest extent. The $k$-means clustering algorithm takes the clusters as parameters into the algorithm, and divides the data set into $K$ clusters according to the similarity [10]. Try to make the elements within the cluster similar and the data between the clusters different.

The basic process are as follows:

(1) Arbitrarily select a point from the collection of input data objects set as the cluster center

(2) For each point in the data set, bring into equation (1) to calculate its distance $d(x)$ from the nearest cluster center

$$d(v-\omega)=\sqrt{\sum_{i-1}^{n}(v_i-\omega_i)^2\, i=1,\ldots,n}, \tag{1}$$

where $v$ is the cluster center, and $\omega_i$ is the $i$-th point in the dataset;

(3) According to the minimum distance assign each point to each cluster center;

(4) Calculate the mean of each cluster and determine it as the new cluster center;

(5) When the conditions for iterative termination are met, the calculation is terminated; otherwise, continue to step (2) to execute the step.

The classic $K$-means algorithm is fast, simple, has good scalability, and has high efficiency for large data sets. Its time complexity is /(nKT) and is nearly linear, which is suitable for mining large-scale data. However, the value of K is difficult to estimate, and the number of iterations and the final result will be greatly affected by the initial point. Due to the large amount of KDDCUP99 data set and the mixture of normal data and intrusion data, the training set is first divided into two categories by the K-means algorithm clustering module, which is called the attack data set AD (Attackdata set), the normal data set ND (Normaldata set), subdivided as the following Table 1:

*2.2. Convolutional Neural Network Training.* Convolutional Neural Network (CNN), as a semisupervised neural network, has a better effect in machine learning for data stream feature abstraction [11], and has excellent feature learning ability, so it is the focus of application research in the intrusion detection industry in recent years. CNN consists of two parts: convolutional layer and pooling layer [12]. The feature of the convolutional layer is that the data features of the original data are extracted by the convolution operation. Relatively speaking, the pooling layer pays more attention to the original data itself rather than its features [13]. It retains the main features of the original data, and reduces the dimension of the high-dimensional data space where the data is located to avoid the disaster of dimensionality. The mathematical characteristic of CNN is the result of multiplying two vectors and summing them up and the core idea of convolution is to extract data features. The initial convolution process is to first outline the features, such as classifying the samples, and then convolving the sample data features. The deeper the convolution layer, the higher the feature complexity and the more detailed the

Table 1: Data cluster identification.

| Category | Identification category | Implication | Segment specific identification |
|---|---|---|---|
| 1 | Normal | Normal data | Normal |
| 2 | DDOS | Distributed denial of service | Back, land, Neptune, pod, etc. |
| 3 | Probing | Probing attack | Ipsweep, nmap, portsweep, etc. |
| 4 | R2L | Remote-to-Login | ftp_w, guess_password, imap, etc. |
| 5 | U2R | User-to-Root | buffer_overflow, loadmodule, etc. |

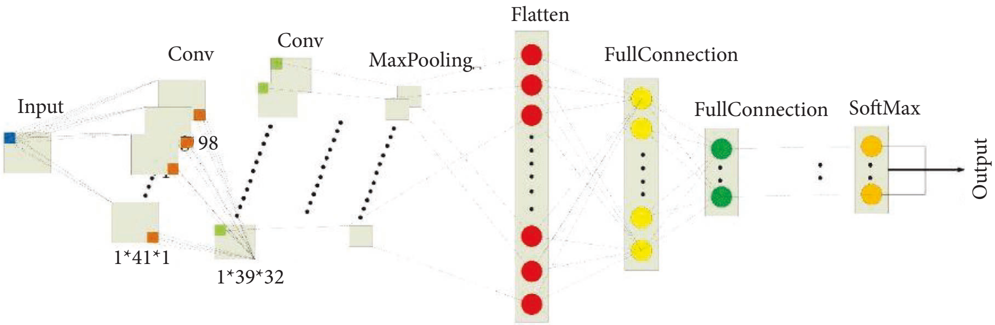

Figure 2: The structure of CNN mode.

difference. Through the volume 0 layer, on the premise of retaining the main characteristics of the data, the dimensionality of the data features is reduced to achieve the effect of removing redundancy. Usually, after the dataset passes through many convolutional layers and pooling layers, after being processed by the fully connected layer, it is finally classified by Softmax to obtain the classification result of intrusion detection. The author will use an eight-layer network model to build a neural network training system in an intrusion detection system, using two layers of convolution and pooling. The author defines the loss function with cross entropy that characterizes the gap between the predicted result and the real result, and uses the *L*2 regularization method to prevent the model from overfitting.

The *L*2 regularization formula is very simple, directly adding the sum of the squares of the weight parameters to the original loss function. Assuming that the network layer parameter to be regularized is *w*, the *l*2 regularization form is as follows:

$$L = E_{\text{in}} + \lambda \sum \left| w_j \right|. \tag{2}$$

Among them, *E*_in is the training sample error that does not contain the regularization term, $\lambda$ controls the size of the regularization term, and a larger $\lambda$ value will constrain the model complexity to a greater extent, and vice versa. In actual use, the regular term is generally added to the objective function (loss function), and the error of the overall objective function is backpropagated, so as to achieve the effect of the regular term and guide network training. *L*2 regularization is commonly called "weight decay" in deep learning, and *L*2 regularization is also called "ridge regression" or Tikhonov regularization in machine learning.

And he added the Dropout layer after the fully connected layer, and then obtained the classification probability through the Softmax layer. The final category is the category with the highest probability. The training model of the convolutional neural network in the intrusion detection system in this paper is shown in Figure 2. The convolution kernel of the first layer of convolution is 1 × 41 × 1, and the second layer of convolution kernel is 1 × 39 × 32. The pooling layer of the model adopts maximum pooling, and then connects two fully connected layers, and the number of nodes in the final output layer is 40.

*2.3. System Construction.* The overall process framework is shown in Figure 1. First of all, we need to use the *K*-means clustering algorithm to divide the preprocessing data set into *K* clusters (*K* is the number of different clusters). Next, the initial score data is processed by convolution, pooling, and other modules after passing through the convolutional neural network model. Then get the categorical data by SOFTMAX. Finally, by matching data features with behavior features or pattern features, it is possible to determine whether there is a network attack or abnormality, and the category of the abnormal behavior (normal (in the case of initial false exposure), DDoS, U2R, R2L, etc.). The result is that we get the intrusion detection result of the obtained data.

In the whole process, the *K*-means algorithm is used to initially screen the data, the *K* value is set to 2, and the initial data is marked as normal or abnormal cluster (Attack). The cluster anomaly data is then subdivided through a convolutional neural network, and the anomaly types are subdivided into 4 categories with a total of 39 attack types. Among them, 22 attack types appeared in the training set, and another 17 unknown attack types appeared in the test set. The purpose of this design is to test the generalization ability of the classifier model, and the ability to detect unknown attack types is an important indicator to evaluate the quality of the intrusion detection system. The four exception types are shown in Table 1. Each connection record in the KDDCup99 training data set contains 41 fixed feature attributes and a class identifier, which are used to indicate that the connection record is normal or a specific attack type. Among the 41 fixed feature attributes, 9 are symbolic, and the others are continuous. We use Python to preprocess the data, including numerical replacement text, numerical normalization, and label one-hot encoding. The numerical replacement text in this is mainly to convert the value of each 41 eigenvalues of each connection into a numerical form.

The specific steps of the intrusion detection framework are described as follows:

*Step 1.* Get the original network data packets from the network.

*Step 2.* Preprocesses the original network data preprocessing main.

It is divided into 3 steps:

(1) Feature extraction. Use feature extractor on raw network packets.

Feature extraction.

(2) Attribute mapping. Convert the extracted character network data to numeric values type of data.

(3) Data normalization. Due to the large difference between the data of the same attribute, the influence.

The training of the neural network is affected, so the data is normalized to the [0, 1] interval.

*Step 3.* K-means algorithm sorting: perform preliminary screening of the obtained standard data set.

*Step 4.* Data separation: divide the obtained image data into training data, test data, andcertification data. Among them, the training data is used to train the CNN model, and the test.

The proof data is used to test the effect of the CNN model training process, and the test data is used to.

It is used to test the effect of the trained model.

*Step 5.* Model Training: training and Parameters for Deep Convolutional Neural Networks Tuning.

(1) Initialize the CNN model parameters.

(2) Train the CNN model, using the validation data set to complete the training in each round.

The deep convolutional neural network is verified. After the neural network training is completed, according to the verification.

According to the verification results, the parameters in the deep convolutional neural network are adjusted until the model is optimal.

(3) Finally, a trained optimal deep CNN model is obtained.

*Step 6.* Input the test data set after preprocessing in Step 2.

To the trained deep CNN model, the classification prediction of each data is obtained. Test results.

Table 2: Experimental environment configuration.

| Configuration | Environment |
|---|---|
| Intel core i7-7700 CPU @ 2 80 GHz | Ubuntu16.0 |
| Nvidia GeForce GTX 1050(6 GB) | Python3.72 |
| 16 GB RAM | Pytorch1.10.0 |

The entire intrusion detection system has roughly four steps. Firstly, the captured network traffic is subjected to in-depth packet analysis, and secondly the $K$-means clustering algorithm is used to perform initial classification. And then, data preprocessing is performed on the abnormal data in the initial score data, and the convolutional neural network model is trained. And finally detect network traffic through the trained convolutional neural network model. The intrusion detection system model based on convolutional neural network is shown in Figure 1. The model is mainly composed of three parts: data preprocessing module, data hierarchy module, and convolutional neural network module. The principle of the model is as follows: first, the data to be detected is obtained by in-depth analysis of the data packets from the network traffic, and the data to be detected is processed by the preprocessing module. Secondly, the data to be detected enters the data layer change module, each layer of data is processed layer by layer and the convolutional neural network module, and the classification data is obtained through the excitation process. Thirdly, by matching the classified data with the network attack behavior characteristics or pattern characteristics, we can further judge whether there is a network attack or abnormality, and the category of the abnormal behavior (Normal, Probe, DoS, U2R, R2L, etc.). Finally, we can Get the IDS test results of the data. This model can help system administrators to detect network security vulnerabilities in their organizations and respond in a timely manner, repair the vulnerabilities to avoid being attacked, or lose important information and prevent the security of the network from being threatened.

## 3. Experimental Results and Analysis

In this study, the detection verification and comparative test experiments of the detection model are carried out on the Linux operating system. Use Python's deep learning library Pytorch to program the CNN in this article. In order to improve computing efficiency and reduce training time, the Pytorch-GPU version is used for parallel computing acceleration. The hardware and software configuration environment of the experiment is shown in Table 2.

*3.1. The Introduction of the Data Set.* The raw data for KDDCup99 comes from the DARPA Intrusion Detection Assessment Program in 1998, and all network data comes from a simulated US Air Force LAN with many simulated attacks added. The training data for the experiment was seven weeks of network traffic, which contains about 5 million network connections. The test data of the experiment was two weeks of network traffic and contained about two

Table3: Setting of CNN model parameter.

| CNN parameter | Value |
|---|---|
| Input layer | INput_dim = (11, 11) |
| Convolutional layer 1 | Number of convolution kernels = 16, Stride = (5, 5), activation = relu |
| Pooling layer 1 | Stride = (2, 2), |
| Convolutional layer 2 | Number of convolution kernels = 8, Stride = (5, 5), activation = relu |
| Pooling layer 2 | Stride = (2, 2), |
| Fully connected layer 1 | Number of nodes = 100 activation = relu |
| Fully connected layer 2 | Number of nodes = 50 activation = relu |
| Output layer | Number of nodes = 5 activation = Softmax |

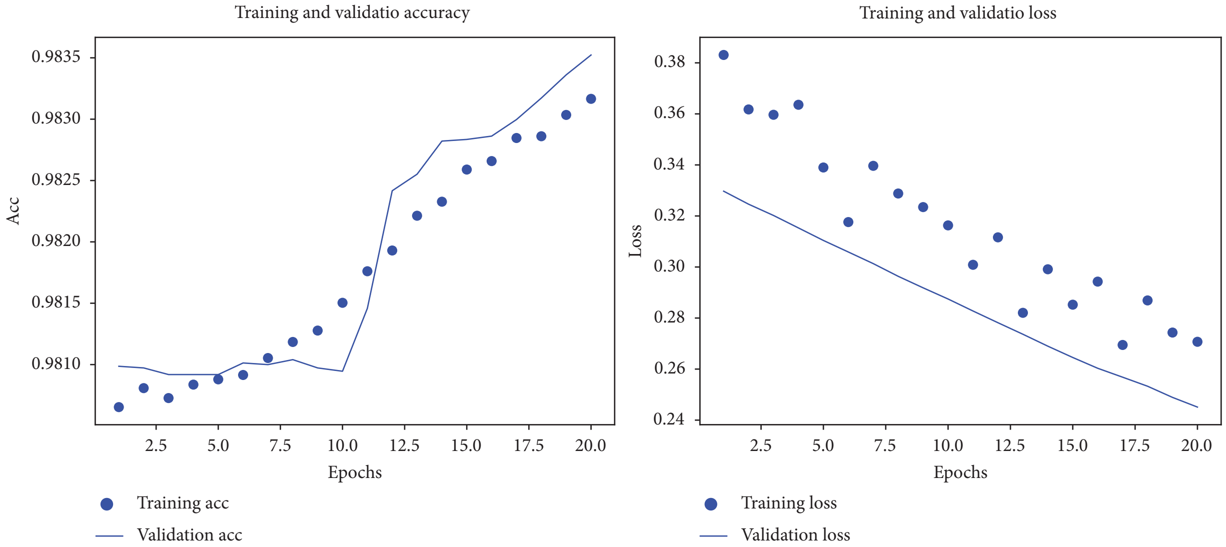

Figure 3: Accuracy and loss function and iterative relationship.

million connections. By processing the above datasets, a new data set can be formed. This data set was used in the KDDCUP competition held in 1999 and became the famous KDD99 data set. Although it has been a long time since now, the KDD99 dataset is still an important empirical dataset in the field of network intrusion detection.

*3.2. The Normalization of Data.* Based on the TFA model, the original data is processed by missing value cleaning, eigenvalues, and quartile outlier cleaning to generate model indicators, and the generated indicators need to be standardized. Common methods for this processing include min-max normalization, $Z$-score normalization, and mean normalization. In this paper, the $Z$-score standardization method is used to process the generated indicators. This method uses the Numpy.std function to substitute the model indicators into formula (3), respectively, to obtain the indicator properties of the clustering algorithm.

$$P^* = \frac{P - \mu_P}{\delta_P}, \tag{3}$$

where $\mu$ is the mean of all sample data, and $Y$ is the standard deviation of all sample data.

It ends up preprocessing into 4 files (train_x.csv, train_y.csv, test_x.csv, test_y.csv).

*3.3. Experimental Verification.* In order to test the effect of this model in the intrusion detection system, the convolutional neural network is tested on the KDD CUP99 data set. We first preprocess the KDD CUP99 data set, and then input it into the $K$-means clustering algorithm in this paper for initial classification, and then put the abnormal data after initial classification into CNN to train the model according to the proportion of 20% of the training data set. CNN settings are shown in Table 3.

The accuracy and loss function curves of the training set and validation set are shown in Figure 3, where acc and val_acc represent the accuracy of the training set and validation set, respectively, loss and val_loss represent the loss function curve of the training set and validation set, respectively.

What we need to do is to record the model accuracy of the first 20 iterations of the convolutional neural network, and then we can get the curve that the accuracy increases according to the number of iterations. The experimental results are shown in Figure 3. After studying the rendering of the convolutional neural network in Figure 3, we found that the convolutional neural network seemed to reach the optimal value after 12 iterations. In order to prevent overfitting, we chose to stop training at 12 rounds, which can quickly converge to best results. At this point in the experiment, all available data (training data + validation data) can be used for training to generate the final model. At the same time, we can pass the test set into the model to get the prediction result, and the prediction result is: Test loss: 2.06, Test accuracy: 90.57%.

When training the model for the first time, the training set is divided into 20% of the data as the validation set, so that the training set and the validation set are the same distribution, and the training accuracy and validation accuracy are high. When training the model for the second time, the training accuracy of each iterations is also high, most of which are around 97%. However, when we evaluated the model using never-before-seen data (test set), we found that there was a certain gap between the model testing accuracy and the training accuracy, and the model testing accuracy became about 91%. In my opinion I think the main reason for this is that the test set is not the same distribution as the training set. There are 17 new attack types in the test set that have never appeared in the training set, so that the model cannot learn the network traffic rules of these attack types from the training set. But from another point of view, this shows that the model still has a certain degree of overfitting and cannot generalize well.

## 4. Conclusions

In the increasingly huge network data environment, in the face of complex and changeable types of attacks, in order to improve the performance of intrusion detection, the author applies a variety of data mining techniques to the intrusion detection system, and uses the *K*-means clustering algorithm to analyze the initial data. After processing, the data is used for training the convolutional neural network, and finally the trained model is used as a classifier for the intrusion detection system to perform real-time detection of network traffic. The multi-layer convolution extracts the features of normal and abnormal network traffic, and the fully connected layer and Softmax are used to classify the network traffic after feature extraction. The training model is detected on the KDD CUP99 test set.

## Data Availability

The authors confirm that the data supporting the findings of this study are available within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by Science and Technology Department of Guizhou Foundation under grant no. LH[2016] 7023.

## References

[1] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[2] M. S. ElSayed, N. A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," *Journal of Network and Computer Applications*, vol. 191, Article ID 103160, 2021.

[3] M. S. Elsayed, H. Z. Jahromi, M. M. Nazir, and A. D. Jurcut, "The role of CNN for intrusion detection systems: an improved CNN learning approach for SDNs," in *Proceedings of the International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, pp. 91–104, Springer, Cham, 2021.

[4] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Generation Computer Systems*, vol. 113, pp. 418–427, 2020.

[5] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion CNN-LSTM-based approach," *Sensors*, vol. 21, no. 2, p. 626, 2021.

[6] J. Hu, C. Liu, and Y. Cui, "An improved CNN approach for network intrusion detection system," *International Journal on Network Security*, vol. 23, no. 4, pp. 569–575, 2021.

[7] Z. Zhi-feng, L. I. Ming-hui, and Y. Zhang, "Improved k-means clustering algorithm for adaptive k value in machine learning," *Computer Engineering and Design*, vol. 42, no. 01, pp. 136–141, 2021.

[8] L. Shao, X. Zhou, C. Zhao, and Xu Zhang, "Improved K-means clustering algorithm based on multi-dimensional grid space," *Journal of Computer Applications*, vol. 38, no. 10, pp. 2850–2855, 2018.

[9] Y. Chun and L. Lu, "Classifying non-life insurance customers based on improved SOM and RFM models," *Data Analysis and Knowledge Discovery*, vol. 4, no. 04, pp. 83–90, 2020.

[10] C. Yue, L. Wang, D. Wang, R. Duo, and X. Nie, "An ensemble intrusion detection method for train Ethernet consist network based on CNN and RNN," *IEEE Access*, vol. 9, pp. 59527–59539, 2021.

[11] E. Zhu, Y. Zhang, P. Wen, and F. Liu, "Fast and stable clustering analysis based on Grid-mapping K-means algorithm and new clustering validity index," *Neurocomputing*, vol. 363, pp. 149–170, 2019.

[12] P. Anitha and M. M. Patil, "RFM model for customer purchase behavior using K-means algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, 2019.

[13] S. H. I. Hai-yang, Y. U. Hui-qun, and G.-sheng Fan, "Customer segmentation and optimization based on clustering algorithm," *Computer Engineering and Design*, vol. 40, no. 11, pp. 3282–3287, 2019.