A Novel Network Intrusion Detection Method Based on DSAE-PSOCNN Model

Qiaochu Sun, Hong Dai*, Yao Xu, and Tianwei Shi

***Abstract*—Network intrusion detection plays a vital role in information network security protection. To solve the deficiency of the feature dimension reduction and the detection performance, we propose a novel intrusion detection model, referred to as DSAE-PSOCNN. The proposed model is a deep learning model which fuses with the improved Sparse Auto-Encoder (SAE) and optimization of Convolutional Neural Network (CNN) based on Particle Swarm Optimization (PSO). The intrusion data has problems such as high feature dimension and noise data, which may cause overfitting. The model of DSAE designs SAE based on Directed Acyclic Graph (DAG) structure to solve the above problems. It extracts differentially the characteristics of each attack type by analytic hierarchy process and gets the high correlation features. The hyper-parameters of CNN are optimized by using PSO algorithm in PSOCNN model to select independently the best CNN structure without the guidance of experience. Finally, the excellence of the proposed model is verified on the CIC-IDS2017 dataset. DSAE-PSOCNN model achieves an accuracy of 98.6% and it compares with the other three models. We conclude that DSAE-PSOCNN model outperforms the comparative models in the precision and recall rate. The suggested model provides an effective solution to the feature dimension reduction.**

***Index Terms*—intrusion detection, sparse auto-encoder, convolutional neural network, particle swarm optimization, directed acyclic graph**

I. Introduction

MACHINE learning has the excellent performance in describing nonlinear complex systems, so many techniques have been proposed to resolve intrusion detection problems. Different techniques mainly include Support Vector Machine (SVM) [1, 2], Decision Tree (DT) [3], K-Means Clustering [4, 5], etc. However, with the development of the network, intrusion data has the characteristics of diversification, complication, and high dimensionality. The traditional machine learning technology can easily raise curse of dimensionality which is unable to meet the needs of current intrusion detection.

Deep learning can efficiently learn internal feature representations and excellently process intrusion data. Hinton first came up with Deep Belief Networks (DBN), which broke the bottleneck of Back Propagation (BP) neural network development [6]. Researchers were inspired to use the deep learning models including Multi-layer Perception (MLP) [7], Recurrent Neural Network (RNN) [8, 9], Long Short-Term Memory (LSTM), and CNN [10].

The intrusion data has problems such as high feature dimension and noise data. They may cause overfitting. The most common way to deal with problems is feature dimension reduction. It mainly includes Principal Component Analysis (PCA), Independent Component Analysis (ICA) [11], Locally Linear Embedding (LLE) [12], and Autoencoder (AE). Compared with other methods, AE has strongest generalization ability and is suitable for the complex and large dataset. It gets better performance by increasing the depth of its network.

To solve above challenges, a novel intrusion detection model of DSAE-PSOCNN is proposed in the paper. The model is validated by using the CIC-IDS2017 dataset. Our main contributions are as follow.

1) We present a feature dimension reduction technology, DSAE, which models SAE based on DAG structure for large scale dataset. Firstly, it is used to reduce the dimension of each attack class layer-by-layer. Feature fusion is performed on the dimension-reduced data afterwards. Strong correlation features are obtained. The method can not only excavate latent distribution of each attack type but also avoid the loss of the key information. It improves the detection rate significantly.
2) We propose a novel intrusion detection model DSAE-PSOCNN which integrates DSAE and PSOCNN. In the model, 2D image representations of the dataset are trained. It is a multi-classification model and can accurately and effectively predict the identification of attacks.
3) The CIC-IDS2017 dataset is used to evaluate the performance of different classification models. Simulation shows that our model is superior to the comparative models and provides an effective dimension reduction method for high dimension data.

II. Related Work

With the rapid development of computer technique, user requirements are increasing. Network intrusion detection plays an important role in defensive system of information security. Concerning the imbalance of the data, H. Zhang et

Manuscript received March 10, 2022; revised September 14, 2022. This work was supported by Graduate Education Reform and Innovation Entrepreneurship Project, University of Science and Technology Liaoning (2021YJSCX09), Natural Science Foundation project of Liaoning Province (2021-KF-12-06), and Project of Liaoning BaiQianWan Talents Program.

Qiaochu Sun is a Postgraduate of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: 912410855@qq.com).

Hong Dai is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (corresponding author to provide phone: +086-186-4226-8599; fax: 0412-5929818; e-mail: dear_red9@163.com).

Yao Xu is a Postgraduate of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: 2306387642@qq.com).

Tianwei Shi is an associate Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: tianweiabbcc@163.com).

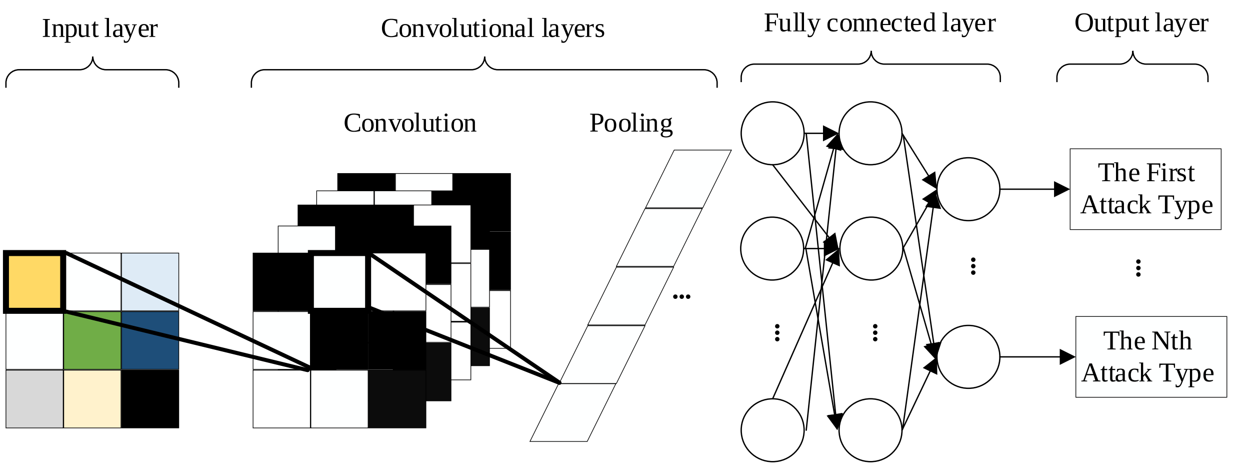

Fig. 1. The structure of Convolutional Neural Network

al. [13] designed a model named SGM-CNN. It avoided under-sampling caused by the loss of significant samples. However, the traditional method would train slowly and increase the complexity of space and time. M. T. Nguyen et al. [14] proposed a feature subset selected by Genetic Algorithm (GA) and fuzzy C-means clustering (FCM). The classification work used the CNN model. G. Andresini et al. [15] utilized 2D images of network traffic to train CNN model. It got the underlying data patterns that appeared in adjacent streams. Y. Zhang et al. [16] proposed a parallel cross convolutional neural network (PCNN). It reduced the quantity of useless elements in network learning and improved convergence speed by extracting stream features.

Feature dimension reduction learns original data and focuses on new key features out of the original features. Data redundancy can affect the accuracy of the classification. Curse of dimensionality about the intrusion detection dataset makes it difficult to classify. S. Zavrak et al. [17] used Variational Auto-Encoder (VAE) and AE to concentrate on the detection. M. Al-Qatf et al. [18] proposed a self-learning framework, STL-IDS, which used SAE for the dimension reduction and utilized SVM for the prediction. Although the experiment achieved a high accuracy, the model couldn't be accurately evaluated only by comparing with a single SVM model. A. H. Mirza et al. [19] processed the sequential nature of network data by using LSTM. They reconstructed features by utilizing AE. The experiment had achieved certain results, but the feature dimension was carried out only through a single SAE. It was difficult for a single SAE to learn strongly corrected features.

On the basis, we propose a novel intrusion detection model DSAE-PSOCNN. Above all, DSAE is used to effectively reduce dimension of the CIC-IDS2017 dataset. Then CNN is optimized by using PSO algorithm in the classification module. It can improve the detection rate of all attack types.

III. Methodology

A. *Sparse Auto-Encoder*

Sparse Auto-Encoder (SAE) adds a sparsity penalty term based on the traditional AE. SAE is proposed by A. Ng. The traditional AE is composed of three layers: input layer, hidden layer, and output layer. It is mainly used for data dimension reduction. SAE is proposed to avoid sample features learning automatically when there are more hidden-layer nodes than input-layer nodes. It mainly adds sparsity constraints on hidden-layer nodes to suppress hidden-layer neurons.

The traditional AE reconstructs the input data x through encoding and decoding. It gets output data y finally. The process of the encoder is shown in Eq. (1) and the process of decoder is shown in Eq. (2).

$$h = \sigma(W_1 x + b_1) \quad (1)$$

Among them, σ is the representation of the activation function; W_1 is the weight of the encoding procedure; b_1 is the bias unit of the encoding process.

$$y = \sigma(W_2 h + b_2) \quad (2)$$

Among them, W_2 means the weight of the decoding process; b_2 is the bias unit of the decoding procedure.

Reconstruction error calculation of AE is the difference between x and y, as shown in Eq. (3).

$$L_{AE}(W,b) = \frac{1}{n}\sum_{i-1}^{n}(\frac{1}{2}\|y_i - x_i\|^2) \quad (3)$$

n is the quantity of training sample.

The input data is expressed as the less neuron. SAE introduces sparse constraints in the hidden layer to inhibiting the activation level of nodes. Its loss function is shown in Eq. (4).

$$L_{SAE}(W,b) = L_{AE}(W,b) + \lambda\sum_{j=1}^{m}(\mu\log\frac{\mu}{\hat{\mu}_j} + (1-\mu)\log\frac{1-\mu}{1-\hat{\mu}_j}) \quad (4)$$

Among them, λ means the sparse weight; m is the number of hidden-layer nodes; μ is the activation level of hidden-layer nodes; $\hat{\mu}_j$ is an average activation of hidden-layer node j.

B. *Convolutional Neural Network*

Convolutional Neural Network (CNN) is a type of feedforward network with deep structure. It is widely used in Natural Language Processing (NLP), image recognition, etc. CNN is biology-inspired like Artificial Neural Network (ANN). Up to now, the architecture of CNN has several improvement strategies. They are mainly composed of input layer, output layer and hidden layer. The hidden layer includes convolutional layer and fully connected layer. The structure of CNN is shown in Fig. 1.

The convolutional layer is the core layer of the whole framework. It enhances the characteristics of input data and improves learning ability, as shown in Eq. (5). The pooling layer prevents too many parameters. It is usually interspersed between continuous convolution layers. The full connected layer mainly applies for classifications, as shown in Eq. (6).

$$x_j^t = \sum_i y_i^{t-1} \otimes k_{ij}^{t-1} + b_j^t \tag{5}$$

Among them, x_j^t represents the output of position j in layer t; y_i^{t-1} is input matrix i in layer $t-1$; k_{ij}^{t-1} is the joint kernel between the previous layer i and the new layer j; b_j^t represents the bias unit of layer t in position j.

$$h_{w,b}(x) = f(W^T x + b) \tag{6}$$

Among them, x represents input of the neuron; $h_{w,b}(x)$ is output of the neuron. Common nonlinear activation functions are *Sigmoid*, *Tanh* and *ReLU*.

IV. Build Model

The schematic diagram of network intrusion detection system (NIDS), DSAE-PSOCNN, is shown in Fig. 2. The system is mainly composed of two modules: a data preprocessing module and a classification module. The first part is the data preprocessing module that is more conducive to the subsequent operations. It is primarily responsible for data cleaning, imbalance preprocessing, data normalization, feature dimension reduction and image processing on the CIC-IDS2017 dataset. In the part of the feature dimension reduction, we present DSAE model. It can obtain representative features according to the features of each attack type. In the end, providing good initial points for the subsequent module. The second part is the classification module which principally classifies the processed dataset. We select the best CNN structure by PSO algorithm. It can avoid wasting a lot of resources by relying on manually configure hyperparameters.

A. *Directed Acyclic Graph of Sparse Auto-Encoder*

Feature dimension can avoid data redundancy by focusing on new key features out of the original features. It can also enhance the accuracy of the classification. Based on above, we propose SAE based on DAG structure, referred to as DSAE. If applying a single SAE, it can’t learn a great many of representative features. Furthermore, analytic hierarchy process is suitable for network traffic. Therefore, we present DSAE model for feature dimension in the paper, and its basic framework is shown in Fig. 3.

Categories are divided into tree-structure, each node corresponds to a dichotomy in DAG. Firstly, the preprocessed data is divided into several subsets. Each of subset contains only two attack types and inputs into DSAE. DSAE is used for dimension reduction. It inputs corresponding models before encoding and decoding. Meanwhile, the reconstruction loss is minimized by adjusting network parameters. Finally, feature fusion is performed. The results are inputted into PSOCNN algorithm for classification subsequently.

The feature dimension model of DSAE is shown in Fig. 4. DSAE is composed of several SAE, each of which reduces dimension for only two attack types in the dataset. Test data firstly enter the apex of DSAE. Each individual SAE only reduces dimension for BENIGN and Infiltration in the dataset. Then it divides them into normal data or attack data. According to the above method, the dimension is reduced layer by layer until it is divided into the final type. By analogy, the feature dimension reduction model of DSAE contains 56 groups.

B. *Convolutional Neural Network Based on Particle Swarm Optimization*

In 1995, Eberhart and Kennedy first proposed Particle Swarm Optimization (PSO) [20] that had the advantages of simple structure, easy to implement and fast convergence. The proposal of PSO algorithm was inspired by foraging of birds. They found that birds must be able to find the largest food in the area because they would deliver messages to each other during the search process. PSO algorithm is also the same theory, it likes birds to particles. Each particle represents a solution to the optimized problem and is assigned an adaptive value. Particles go through the search space to find the most optimal solution, and other particles also follow the current best particle to search.

In the search space, the position and the velocity of particles are key conditions. Their updating formulas are shown in Eq. (7) and Eq. (8). They may contain multiple iterations before the final location is found.

$$v_{id}^k = v_{id}^{k-1} + c_1 r_1 (P_{best} - x_{id}^{k-1}) + c_2 r_2 (G_{best} - x_{id}^{k-1}) \tag{7}$$

$$x_{id}^k = x_{id}^{k-1} + v_{id}^{k-1} \tag{8}$$

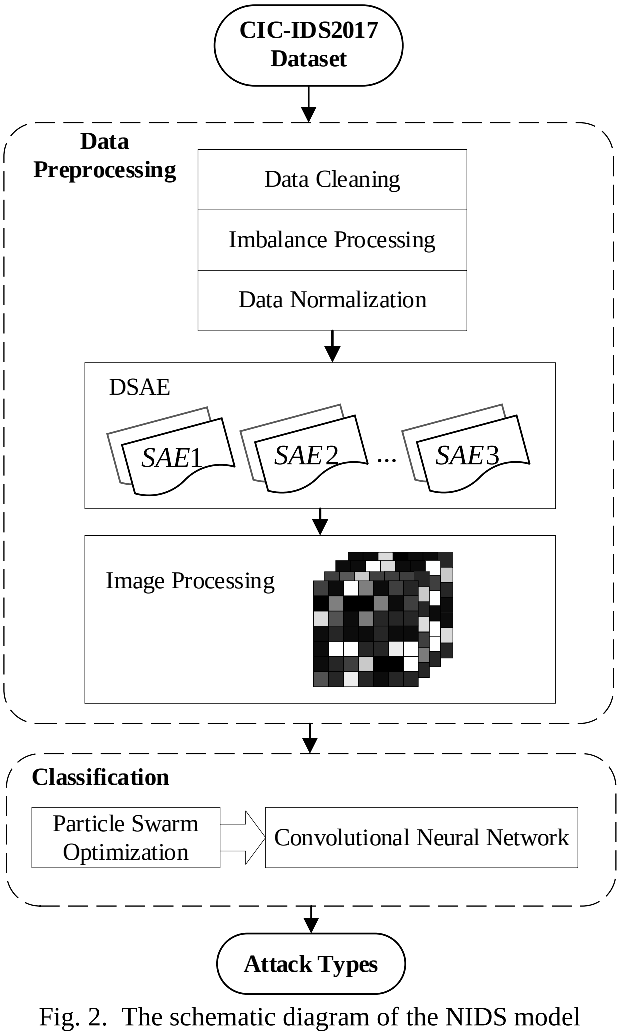

Fig. 2. The schematic diagram of the NIDS model

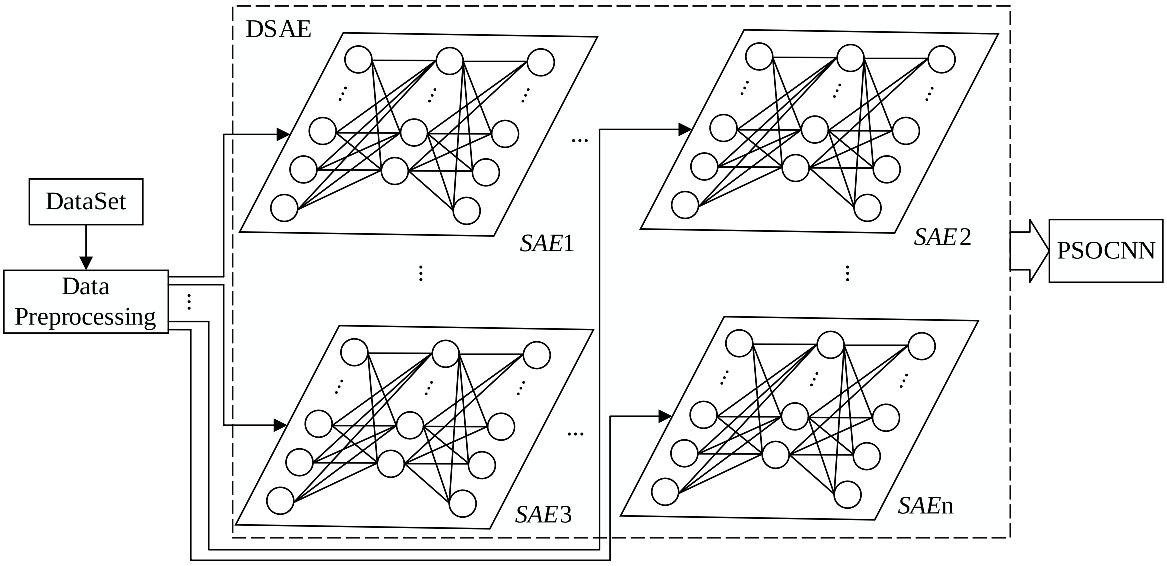

Fig. 3. The network structure of the feature dimension reduction method

Among them, i is the particle; v_{id}^{k} is the velocity of particle i in d-dimension at iteration k ; c_1 and c_2 are acceleration factor; r_1 and r_2 are a random number of between 0 and 1; P_{best} is the individually best solution; G_{best} is the global optimal solution; x_{id}^{k} is the location of particle i in d-dimension at iteration k .

We design an eight-layer 2D-CNN model which improves the classical model of LeNet-5. It includes four-layer convolutional layers, two-layer pooling layers and two-layer fully connected layers. Firstly, the input data is 7*7 grayscale of network traffic. The first four layers are convolution layers, pooling layers are inserted after each two convolution layers. Subsequently, the data is changed single-dimensional data through the flatten layer. It is the transition between convolutional layers and full connected layer. The last two layers are the fully connected layers. The second fully connected layer is classified by the function of *softmax*. We add a dropout layer between two layers to avoid overfitting. The performance of CNN model is closely related to configure the hyper-parameters. However, the best hyper-parameters often depend on configuring artificially. It consumes a lot of time and computing resource. PSO algorithm shows good search ability, so we use PSO to find more outstanding performance so as to select suitable hyper-parameters of CNN model. The PSOCNN algorithm is shown in Fig. 5. The flow of PSOCNN algorithm is described as follows.

Step 1. Firstly, determining CNN parameters that needs optimizing by PSO algorithm. Setting up the value range.

Step 2. Initializing PSO parameters such as number of iterations, population size, etc.

Step 3. Candidate solutions are randomly generated as the

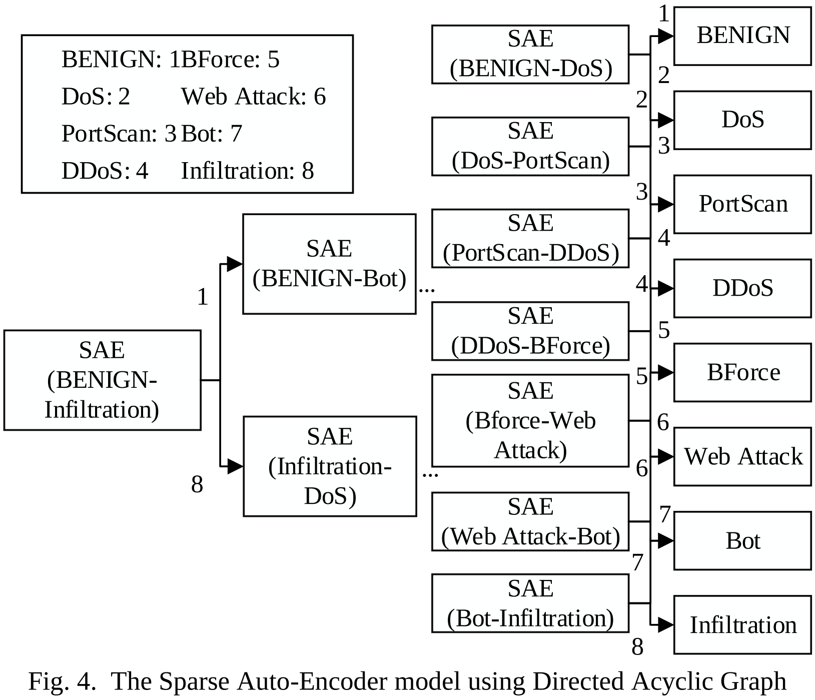

Fig. 4. The Sparse Auto-Encoder model using Directed Acyclic Graph

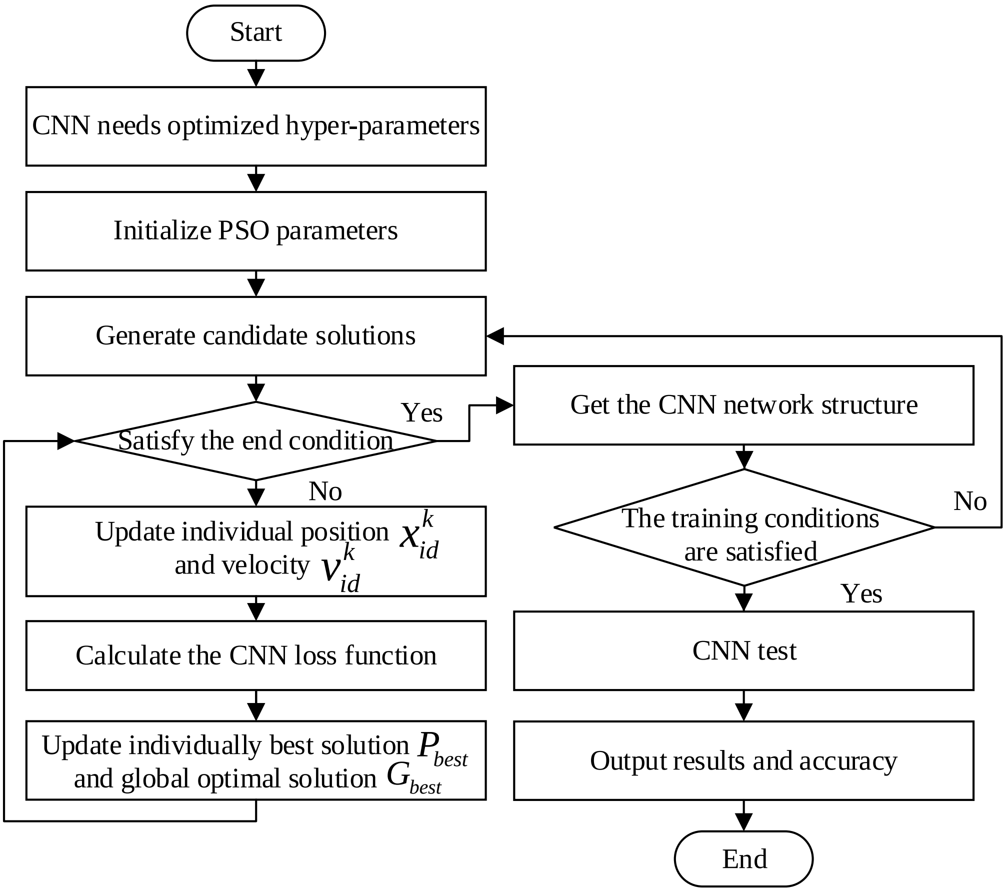
Start
CNN needs optimized hyper-parameters
Initialize PSO parameters
Generate candidate solutions
Satisfy the end condition
Yes
No
Update individual position x_{id}^{k} and velocity v_{id}^{k}
Calculate the CNN loss function
Update individually best solution P_{best} and global optimal solution G_{best}
Get the CNN network structure
The training conditions are satisfied
No
Yes
CNN test
Output results and accuracy
End

Fig. 5. Convolutional Neural Network process based on Particle Swarm Optimization

CNN structure.

Step 4. Updating the location x_{id}^{k} and the velocity v_{id}^{k} of the particles according to Eq. (7) and Eq. (8).

Step 5. The loss function is calculated through training CNN model.

Step 6. Comparing with the fitness value of candidate solution and P_{best}. Updating P_{best} when the fitness value of candidate solution is less than it. Then the fitness value of candidate solution compares with G_{best}. If it is less than G_{best}, G_{best} is updated.

Step 7. Estimating whether the current iteration reaches the maximum iteration. If the condition is met, the best CNN network structure is obtained; otherwise, repeating Step 4 to 6.

V. Experiments

We design the proposed model in Python which uses TensorFlow 2.6.0 machine learning library. Training and testing implement on a PC with Inter Core i5, RAM 16GB, Window 10 and 64-bit operating system.

A. Model Parameters Setting

We improve the classical LeNet-5 model and optimize hyper-parameters by PSO algorithm. The input of PSOCNN model is 7*7 grayscale. PSO algorithm finds the best parameters to improve the predictive ability such as the number of convolution kernels, the size of pooling kernels, etc. The parameter setting models of the standard CNN and PSOCNN are shown in Table I. The iterations of PSO algorithm are 50 and the number of the population is 20.

B. Data Preprocessing

Data preprocessing module primarily contains five parts on the CIC-IDS2017 dataset: data cleaning, imbalance processing, normalization, feature dimension reduction, and image processing.

i. *Dataset Description*

The CIC-IDS2017 dataset is used for simulation in the experiment. The dataset was collected by Information Security Center of Excellence in University of New Brunswick [21]. The stage of data capture has lasted for five days. The dataset comprehensively covers all eleven criteria necessary for a reliable benchmark dataset. It provides eight common attack types.

ii. *Class Imbalance Processing*

CIC-IDS2017 dataset is an imbalanced dataset. There are many instances on the labels. If you look at Bot that only comes in at 0.07%. Training without imbalance processing will make it difficult to deal with few instances of attack types. Finally, it leads to poor results in an experiment.

Firstly, the fifteen labels of the CIC-IDS2017 are combined and reduced to eight main labels. They are replaced with 1 to 8. The sampling process is divided into oversampling and under-sampling. Synthetic Minority Oversampling Technique (SMOTE) is used for oversampling. It can improve the few types of Web Attack, Bot, and Infiltration in data samples. Random sampling scheme without replacements extracts eight kinds of labels respectively from the processed dataset. Finally, the training set accounts for 80%, and the testing set accounts for 20%.

TABLE I
PARAMETER OPTIMIZATION RANGES OF CNN AND PSOCNN

Layer	Type	Parameter	Range	CNN	PSOCNN
L1	Conv2D	Filters	[4,101]	32	36
		Kernel Size	[3×3,5×5,7×7]	5×5	7×7
L2	Conv2D	Filters	[4,101]	16	69
		Kernel Size	[3×3,5×5,7×7]	5×5	5×5
L3	MaxPooling2D	Pool Size	[2,4]	2	3
		Strides	[2,4]	2	4
L4	Conv2D	Filters	[4,101]	16	68
		Kernel Size	[3×3,5×5,7×7]	5×5	5×5
L5	Conv2D	Filters	[4,101]	32	72
		Kernel Size	[3×3,5×5,7×7]	5×5	3×3
L6	MaxPooling2D	Pool Size	[2,4]	2	2
		Strides	[2,4]	2	4
L7	Dense	Neurons	[4,201]	128	134

iii. *Normalization*

Normalizing the numerical values for reasonably scaling the data in a certain range. It facilitates the learning and training of the subsequent model. The values of numerical features are mapped between zero and one, as shown in Eq. (9).

$$x^{*} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{9}$$

Among them, x is the original value of numerical feature; $x_{\min}$ is the minimum value of sample data; $x_{\max}$ is the maximum value of sample data.

iv. *Image Processing*

High-dimensional data can slow down the training process and increase the complexity of time and space. Therefore, the original data is visualized. Visualization dataset is convenient for the observation and CNN has excellent detection effect in image recognitions. The matrix dimensions are constructed according to the dataset after the DSAE algorithm. The dimension of each data is 49-dimension. Afterwards we convert into 7*7 pictures. Eight types of the dataset are shown in Fig. 6.

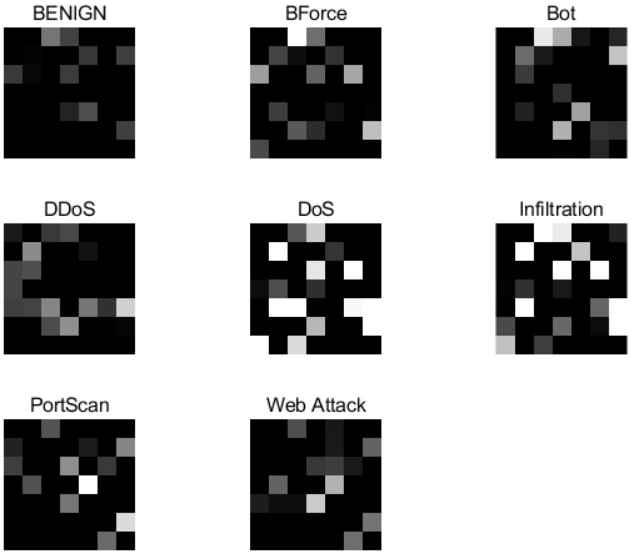

Fig. 6. Gray scale images of CIC-IDS2017 dataset

TABLE II
CONFUSION MATRIX

Confusion Matrix		Predicted Class	
		Positive	Negative
Concrete Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

C. *Evaluation Metrics*

The confusion matrix is used to evaluate the detection effect of the classification model, as shown in Table II.

True Positive (TP): Positive examples are correctly predicted by the model.

False Positive (FP): Negative examples are incorrectly predicted as positive examples by the model.

False Negative (FN): Positive examples are wrongly predicted as negative examples by the model.

True Negative (TN): Negative examples are correctly predicted by the model.

The results of the experiment are evaluated based on four indexes: *accuracy*, *precision*, *recall*, and *F1-score*. All of them are between zero and one. As they approach one, performance of model increases; otherwise, performance of model degrades.

Accuracy refers to the proportion of correct data predicted by the classification model to the total data, as shown in Eq. (10).

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

Precision refers to the proportion of the data correctly predicted as positive cases to all the data classified as positive cases, as shown in Eq. (11).

TABLE III
TRAINING SET CONFUSION MATRIX

Concrete class / Predicted class	1	2	3	4	5	6	7	8
1	6870	12	31	22	26	11	18	12
2	5	1009	0	0	0	0	0	0
3	5	0	978	0	0	0	0	0
4	1	0	0	975	0	0	0	0
5	26	0	0	0	1006	0	0	0
6	8	0	0	0	0	1034	0	0
7	8	0	0	0	0	0	982	0
8	20	0	0	0	0	0	0	941

$$Precision = \frac{TP}{FP + TP} \tag{11}$$

Recall refers to the proportion of the data correctly predicted to be positive to all the data of actual positive, as shown in Eq. (12).

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

F1-score is a comprehensive evaluation that combines *Accuracy* and *Recall* as one criterion, as shown in Eq. (13).

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precison}} \tag{13}$$

D. Result Comparisons and Discussion

Accuracy and loss of the model are important indexes to judge the model performance. In the paper, the accuracy and loss function variation trends of iterations in the CIC-IDS2017 dataset are plotted, as shown in Fig. 7. With the increase of iterations, the accuracy rate improves. It levels off after five iterations and steadily improves finally. The accuracy is 0.986 and loss function is 0.059 in 25 iterations.

Meanwhile, confusion matrix is used to evaluate the accuracy of classification results, as shown in Table III. There is a misclassification between attack data and normal data because some values of normal traffic can’t differentiate from attack data.

Verifying the property of the proposed model, the following three models are compared in the experiment: 1) original CNN model, 2) PSOCNN model: The initial parameters of CNN are optimized by PSO, 3) DSAE-CNN model: DSAE is used to reduce the dimension of the dataset. They are evaluated by the above four evaluation performance: *accuracy*, *precision*, *recall*, and *F1-score*. The index scores of different models are shown in Table IV.

It can be seen from Table IV that among all models, DSAE-PSOCNN model has the best effect with an accuracy of 98.6%. This fully shows that proposed model can play an advantage in feature reduction and the selection of CNN structure. Based on DSAE model, the accuracy of DSAE-CNN model and DSAE-PSOCNN model are 2.4% and 3.0% better than the previous models. In terms of PSO optimization, the accuracy of PSOCNN model and DSAE-PSOCNN model compare with the models before improvements that are improved by 1.8% and 2.4% respectively. To sum up the above, DSAE-PSOCNN model can predict the attack identification accurately and effectively.

F1-score is used to express the overall success rate, so we also maintain our focus on it when analyzing results. In order to demonstrate the success rate of each tag, we compare the above three models with the presented model, as shown in Fig. 8.

As can be seen from broken lines in Fig. 8, DSAE-PSOCNN model basically tends to the top and has the best effect. Except for Bot, other types of data show excellent effects. F1-score of BENIGN normal flow increases by 13.3%, 10.7%, and 6.8% compared with the other three models, respectively. BForce increases by 5.7%, 2.7%, and 4.6%. From the perspective of F1-score improvement, DSAE-PSOCNN model has an excellent effect on processing issues of diverse data and high latitude. The overall success rate is high. It is proved that the proposed model in the paper

TABLE IV
PERFORMANCE COMPARISON OF MULTICLASS CLASSIFICATION

Indicators (%) / Model	Accuracy	Precision	Recall	F1-score
CNN	93.8	93.9	94.0	93.8
PSOCNN	95.6	95.6	95.7	95.5
DSAE-CNN	96.2	96.2	96.4	96.1
DSAE-PSOCNN	**98.6**	**99.3**	**98.1**	**98.6**

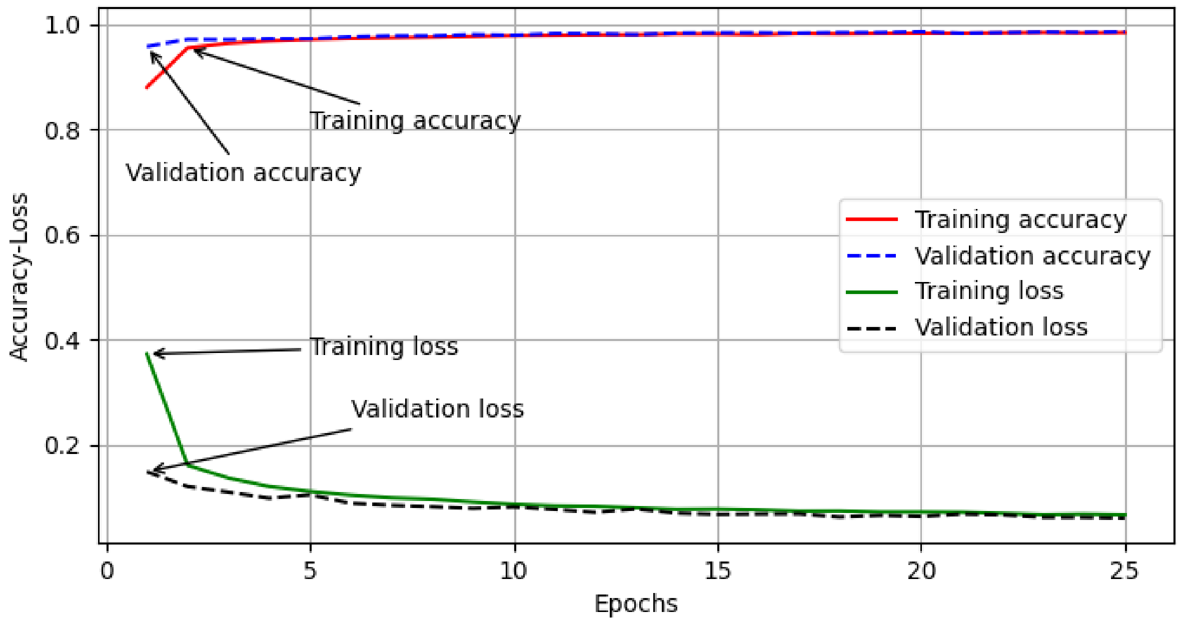

Fig.7. Time distribution of DSAE-PSOCNN model

can not only learn better feature distribution, but also better effect in high dimensional and unbalanced data. Table V specifically shows the detailed results of DSAE-PSOCNN model and other three models on the CIC-IDS2017 dataset.

As can be seen in Table V, DSAE-PSOCNN model has good performance in four evaluation metrics. The proposed model not only significantly improves accuracy and F1-score, but also has excellent properties in Precision and Recall. In terms of DDoS data, the precision of the proposed model is 2.4%, 1.6%, and 3.9% higher than that of the other three models. In the Infiltration data, the recall of the proposed model improved by 2.8%, 2.6% and 0.7%, respectively. In summary, the suggested model in the paper performs feature dimension reduction through DSAE and extracts more representative features. The original data can be visualized for easy observation. It can also form an effective multidimensional feature space and improve the accuracy of the model. The CNN structure was optimized by PSO algorithm. PSO algorithm can take full advantage of the optimum.

TABLE V
DSAE-PSOCNN AND OTHER CLASSIFICATION MODELS

Indicators Model	Performance Index	Model Performance (%)							
		1	2	3	4	5	6	7	8
CNN	Accuracy	77.5	99.1	98.4	96.3	93.7	97.4	99.7	89.5
	Precision	76.4	99.7	98.7	97.5	94.5	97.7	99.5	89.8
	Recall	94.8	88.5	96.7	92.0	89.1	96.1	97.9	97.0
	F1-score	85.3	93.5	97.5	94.1	91.3	96.7	98.7	93.1
PSOCNN	Accuracy	97.1	99.1	99.6	99.3	98.1	99.4	99.7	98.8
	Precision	81.1	99.5	99.7	98.3	95.2	99.4	99.5	92.4
	Recall	95.9	93.6	97.4	96.0	90.0	96.3	98.0	98.0
	F1-score	87.9	96.5	98.5	97.1	92.5	97.8	98.7	95.1
DSAE-CNN	Accuracy	88.5	99.0	100.0	97.2	95.4	98.8	99.6	90.3
	Precision	96.3	97.5	98.4	96.0	95.7	98.9	97.3	90.0
	Recall	95.3	90.5	97.3	97.4	94.3	98.2	97.1	99.3
	F1-score	91.8	94.6	98.6	97.3	94.8	98.5	98.3	94.6
DSAE-PSOCNN	Accuracy	98.5	99.9	99.7	99.8	99.6	99.9	99.8	99.8
	Precision	98.1	99.5	99.5	99.9	97.5	99.2	99.2	97.9
	Recall	98.9	98.8	96.9	97.8	97.5	98.9	98.2	98.7
	F1-score	98.5	99.1	98.2	98.8	97.5	99.0	98.7	98.3

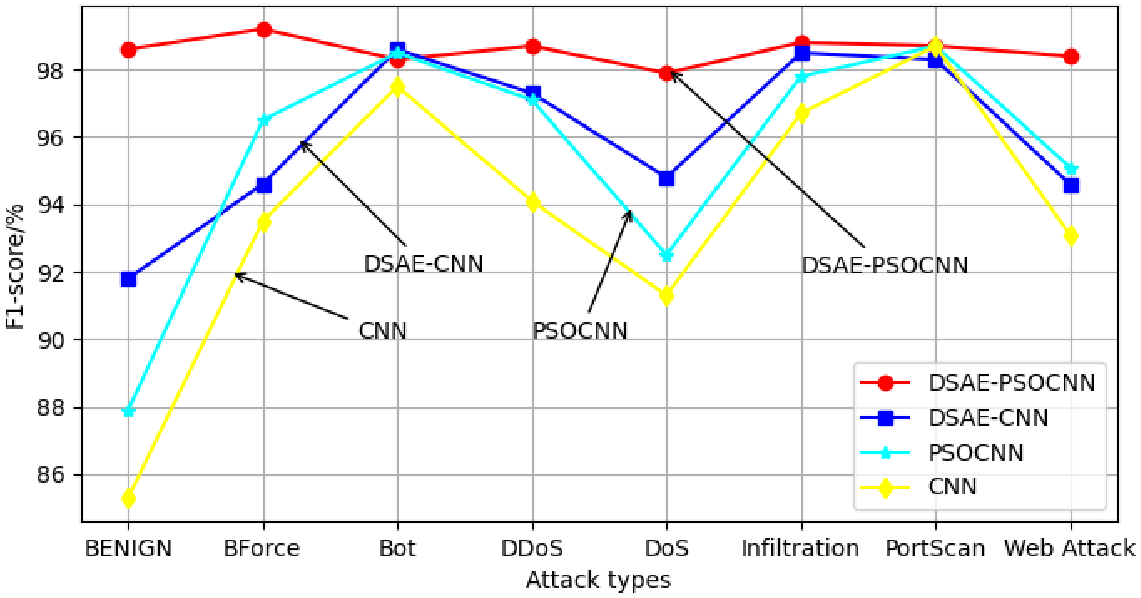

Fig. 8. Comparison diagram of F1-score

VI. Conclusion

We build a network intrusion detection model. We solve the problems of the dataset and parameters with deep learning technology. Aiming at the issue that network traffic data has various data and large characteristic dimension, we apply Directed Acyclic Graph of Sparse Auto-Encoder. DSAE model can reduce dimension layer by layer according to each attack category and learn better feature distribution. It acquires more representative features and lays a solid foundation for subsequent classification work. High-dimensional data will also slow down the training process and increase the time complexity. A multi-dimensional feature space suitable for intrusion detection model is constructed by manipulating the dataset graphically. This also facilitates analysis of the results. Finally, Convolutional Neural Network with optimal performance is searched through Particle Swarm Optimization. The improved model avoids the trouble that the most hyperparameters depend on manual configuration. And it can also save a lot of resources and time and have higher detection accuracy. On the basis, proposed model and three classification models are compared from multiple perspectives through four evaluation indexes. Experimental results show that DSAE-PSOCNN model can learn best feature distribution. It has excellent effect in high dimensional and unbalanced data.

In the future research, we will further consider the convergence of particle parameters about PSO algorithm. We will try to improve the PSO algorithm to achieve better effect. We also plan to explore other data dimension reduction methods for more intrusion detection tasks.

References

[1] Wathiq Laftah AI-Yaseen, “Improving Intrusion Detection System by Developing Feature Selection Model Based on Firefly Algorithm and Support Vector Machine,” IAENG International Journal of Computer Science, vol.46, no.4, pp534-540, 2019.

[2] Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai, and Citra Dwi Perkasa, “A novel intrusion detection system based on hierarchical clustering and support vector machines,” Expert Systems with Applications, vol.38, no.1, pp306-313, 2011.

[3] Vegard Engen, Jonathan Vincent, and Keith Phalp, “Enhancing network based intrusion detection for imbalanced data,” International Journal of Knowledge-Based and Intelligent Engineering Systems, vol.12, no.5-6, pp357-367, 2008.

[4] Vipin Kumar, Himadri Chauhan, and Dheeraj Panwar, “K-Means Clustering Approach to Analyze NSL-KDD Intrusion Detection Dataset,” International Journal of Soft Computing and Engineering, vol.3, no.4, 2013.

[5] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri, “Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system,” Expert Systems with Applications, vol. 67, pp296-303, 2017.

[6] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, “A Fast Learning Algorithm for Deep Belief Nets,” Neural Computation, vol.18, no.7, pp1527-1554, 2006.

[7] Hongpo Zhang, Chase Q. Wu, Shan Gao, Zongmin Wang, Yuxiao Xu, and Yongpeng Liu, “An Effective Deep Learning Based Scheme for Network Intrusion Detection,” 2018 24th International Conference on Pattern Recognition (ICPR), 20-24 August, 2018, Beijing, China, pp682–687.

[8] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He, “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks,” IEEE Access, vol.5, pp21954-21961, 2017.

[9] Muder Almiani, Alia Abughazleh, Amer Al-Rahayfeh, Saleh Atiewi, and Abdul Rzaque, “Deep recurrent neural network for IoT intrusion detection system,” Simulation Modelling Practice and Theory, vol.101, 2020.

[10] Wei-Zhong Sun, Jie-Sheng Wang, Bo-Wen Zheng, and Zhong-Feng Li, “A Novel Convolutional Neural Network Voiceprint Recognition Method Based on Improved Pooling Method and Dropout Idea,” IAENG International Journal of Computer Science, vol.48, no.1, pp202-212, 2021.

[11] James V. Stone, “Independent Component Analysis: an introduction,” Trends in Cognitive Sciences, vol.6, no.2, pp59-64, 2002.

[12] Sam T. Roweis, and Lawrence K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” Science, vol.290, no.5500, pp2323-2326, 2000.

[13] Hongpo Zhang, Lulu Huang, Chase Q. Wu, and Zhanbo Li. “An Effective Convolutional Neural Network Based on SMOTE and Gaussian Mixture Model for Intrusion Detection in Imbalanced Dataset,” Computer Networks, vol.177, 2020.

[14] Minh Tuan Nguyen, and Kiseon Kim, “Genetic convolutional neural network for intrusion detection system,” Future Generation Computer Systems, vol.113, pp418-427, 2020.

[15] Giuseppina Andresini, Annalisa Appice, and Donato Malerba, “Nearest cluster-based intrusion detection through convolutional neural networks,” Knowledge-Based Systems, vol.216, 2021.

[16] Yong Zhang, Xu Chen, Da Guo, Mei Song, Yinglei Teng, and Xiaojuan Wang, “PCCN: Parallel Cross Convolutional Neural Network for Abnormal Network Traffic Flows Detection in Multi-class

imbalanced Network Traffic Flows," IEEE Access, vol.7, pp119904-119916, 2019.

[17] Sultan Zavrak, and Murat Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," IEEE Access, vol.8, pp108346-108358, 2020.

[18] Majjed Al-Qatf, Yu Lasheng, Mohammed Al-Habib, and Kamal Al-Sabahi, "Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection," IEEE Access, vol.6, pp52843-52856, 2018.

[19] Ali H. Mirza, and Selin Cosan, "Computer network intrusion detection using sequential LSTM Neural Networks autoencoders," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2-5 May, 2018, Izmir, Turkey.

[20] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," Proceedings of ICNN'95 - International Conference on Neural Networks 1995, 27 November - 1 December, 1995, Perth, WA, Australia, vol.4, pp1942-1948.

[21] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," 4th International Conference on Information Systems Security and Privacy (ICISSP) 2018, 22-24 January, 2018, Funchal, Portugal, pp108-116.