DACNN: Deep Autoencoding Convolutional Neural Network in Network Intrusion Detection

Ruiwen Deng, Jie Yuan, Xiaoyong Li, Linghui Li, Yali Gao, Wenping Kong
Key Laboratory of Trustworthy Distributed Computing and Service (BUPT),Ministry of Education
Beijing University of Posts and Telecommunications
Beijing, China
Email:{bjjfcbj, buptyuanjie, lxyxjtu}@163.com, lilh92@foxmail.com, {gaoyalibupt, kwenping}@163.com

2022 7th International Conference on Big Data Analytics (ICBDA) | 978-1-6654-7938-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICBDA55095.2022.9760333

***Abstract*—In recent years, the Internet has become the main driving force of global economic growth. However, the ever-changing network security situation is not optimistic. With the development of network technology, the threat of network attacks to key network nodes has seriously affected the security of the entire network situation. In this article, We propose an effective intrusion detection method based on deep autoencoding convolutional neural network: DACNN. First, we trained multiple autoencoders with good performance. Then combine the dimensionality reduction representation of flow information obtained by the autoencoders with the original information, so that convolutional neural networks can use this kind of mixed information that contains the original data and the dimensionality reduction information processed by the autoencoder to analyze the network traffic information. The DACNN has better recognition performance in the field of intrusion detection than ordinary convolutional networks. Tests on multiple datasets show excellent intrusion detection performance:ACC 0.935 in KDDCUP and F_1 0.98 in IDS-2017.**

***Index Terms*—deep autoencoder, convolutional neural network, intrusion detection**

I. Introduction

With the rapid development of network technology, the network involves all aspects of human social life, and the issue of network security has attracted more and more attention. the number of Internet users worldwide had reached 3.8 billion, accounting for 51% of the global total, and 7 of the 10 companies with the highest market capitalization were Internet technology companies, namely Microsoft, Amazon, Apple, Alphabet, Facebook, Alibaba, and Tencent [1].

With the development of network technology and the prosperity of related industries,the threat of abnormal traffic attacks to the network has seriously affected the security of the entire network situation. in order to maximize profits, the black industry gangs constantly adjust their attack schemes and use different malicious network attacks to attack different targets. This has led to the deteriorating cyberspace environment.

the Network Intrusion Detection System [2] (NIDS) has been receiving attention from academia and industry, since Heberlein first proposed it in 1991. NIDS refers to a combination of software and hardware that detects behaviors that endanger computer system security, such as collecting vulnerability information, causing denial of access, and gaining system control rights beyond the legal scope. It's purpose is to identify potential attacks from the message flow on the network. There are constantly new samples or variants of malicious network attacks. The analysis and research work on malicious network attacks is a continuous process. In existing researches, traditional machine learning methods rely more on feature engineering, and have mediocre results. For the defects of the existing model, we propose a new anomaly-based network intrusion detection classification model, which combines the autoencoder and convolutional network to achieve performance improvements in our experiments.

The **contribution** of this article is as follows:

- **We propose a new model: deep autoencoding convolutional neural network(DACNN), and realize the application of deep autoencoding convlutional neural network in the field of intrusion detection.** DACNN consists of two parts: a compression network and an estimation network. The compression network contains multiple autoencoders that need to be pre-trained, and the estimation network consists of two-dimensional convolution Neural network composition. We use the KDDCUP dataset and IDS-2017 dataset to evaluate the effectiveness of the method and achieve exciting results.By comparing with related work, the model achieves the best intrusion detection performance: ACC is increased to 0.935 in KDDCUP and F_1 0.98 in IDS-2017.
- **Our experiments prove that the dimensionality reduction information generated by the autoencoder contains effective information for classification.** We use deep autoencoder to unsupervised learning flow feature information and then combined with two-dimensional convolutional neural network. This method improved the model's ability to abstract traffic data, and proved that the dimensionality reduction information obtained by autoencoder contains important classification information

II. Related Work

This section mainly introduces the research work related to our proposed DACNN model, which is mainly divided into three categories: traditional machine learning based approaches, autoencoder based approaches and others.

A. Traditional Machine Learning Based Approaches

In 2000, the KDDCUP 99 competition which aimed to build a network intrusion detector, a predictive model capable of

distinguishing between 'bad' connections, called intrusions or attacks, and 'good' normal connections. The top three winners used some variants of the decision tree. The winner used a C5 decision tree with a mixture of bagging and boosting [3]. The runner-up first constructed a optimal decision forest and then select an optimal subset of trees to give the final prediction [4]. The third-placed used two-layer decision trees in which the first layer was trained on the connections which cannot be classified by security experts and the second layer was built on the connections which cannot be classified by the first layer, and it is best summarized as recognition based on voting decision trees using 'pipes' in potential space [5]. Besides, Tsang et al. [6] present a multi-objective genetic fuzzy system for anomaly intrusion detection which extracts accurate and interpret able fuzzy rule-based knowledge from network data using an agent-based evolutionary computation framework. Zhang et al. [7] propose a systematic frameworks that apply a data mining algorithm called random forests in misuse, anomaly, and hybrid-network-based IDSs. Liu et al. [8] proposes an incremental unsupervised anomaly detection method that can quickly analyze and process large-scale real-time data.

B. Auto Encoder Based Approaches

Al-Qatf et al. [9] propose an effective deep learning approach, self-taught learning (STL)-IDS, based on the STL framework.The proposed model is built using the sparse autoencoder mechanism. Sparse autoencoder is an effective unsupervised learning algorithm for reconstructing a new feature representation. After the autoencoder stage, the new features are fed into the SVM algorithm to improve its detection capability for intrusion and classification accuracy. Shone et al. [10] detail their proposed nonsymmetric deep autoencoder (NDAE) for unsupervised feature learning,and also propose their novel deep learning classification model constructed using stacked NDAEs.

There are some similar studies, Mighan et al. [11] utilized a stacked autoencoder to decrease features, and also fed the new featuresto a SVM classifier for events classification into normal or attacks.Fararh et al. [12] present a deep learning approach for intrusion detection systems which uses deep autoencoder and trained by a greedy layer-wise method to prevent problems such as overfitting and local optima. Kim et al. [13] employ an reinforcement learning that permits a deep auto-encoder in the network, which achieve the maximum prediction accuracy in online learning systems while detecting intrusions by verifying whether the data is classified as normal or anomalous.

C. Other Deep Learning Based Approaches

About the use of deep neural networks in intrusion detection, Yu et al. [14] propose a session-based network intrusion detection model using a deep learning architecture which can achieve incredibly high performance to detect botnet network traffics. Niyaz et al. [15] propose a deep learning based multi-vector DDoS detection system in a software-defined network (SDN) environment . Abeshu et al. [16] implement a novel distributed deep learning scheme of cyber-attack detection in fog-to-things computing, and show that deep models are superior to shallow models in detection accuracy, false alarm rate, and scalability.

In order to solve the problems that traditional machine learning based abnormal flow detection methods rely heavily on features,and the detection methods based on deep learning are inefficient and easy to overfit, HANG et al. [17] propose an detection method based on one-Dimentional Convolutional Neural Network. Besides, Zhang et al. [18] combined the convolutional neural network with the gated recurrent unit, and used the attention mechanism to find the key features of a single data packet while using the gated recurrent unit to extract the features between the data packets, which improved the classification accuracy, real-time performance and training efficiency. Lei et al. [19] build a high-performance intrusion detection classifier model based on pruning deep neural network in three steps:train a deep neural network,pruning opration and retrain.

On the basis of these investigations, we propose an effective intrusion detection method based on deep autoencoding convolutional neural network, which aims to build a higher performance intrusion detection classifier model and promote the further development of intrusion detection research.

III. Methodology

The section below describe the proposed deep autoencoding convolutional neural network(DACNN) and the details of how DACNN works.

A. Overview

DACNN consists of two major components: a compression network which consists of multiple deep autoencoders and an estimation network which is essentially a 2d-convolutional network. As show in Fig 1, DACNN works as follows:

- **compression network** performs dimensionality reduction for input samples by multiple deep autoencoder, prepares their low-dimensional representations from the reduced space, and feeds the representations to the subsequent estimation network.
- **estimation network** takes the feed, splice with the original data and feed it into the neural network, and finally output the predicted value.

B. Compression Network

1) Deep autoencoders: The low-dimensional representations provided by the compression network contains multiple sources of features: the reduced low-dimensional representations learned by different deep autoencoders with different number of representation dimensions.Given a $input$, n autoencoders, the compression network computes the low-dimensional representations R as follows:

$$r_i = h(input; para_i), i \in [1, n] \tag{1}$$

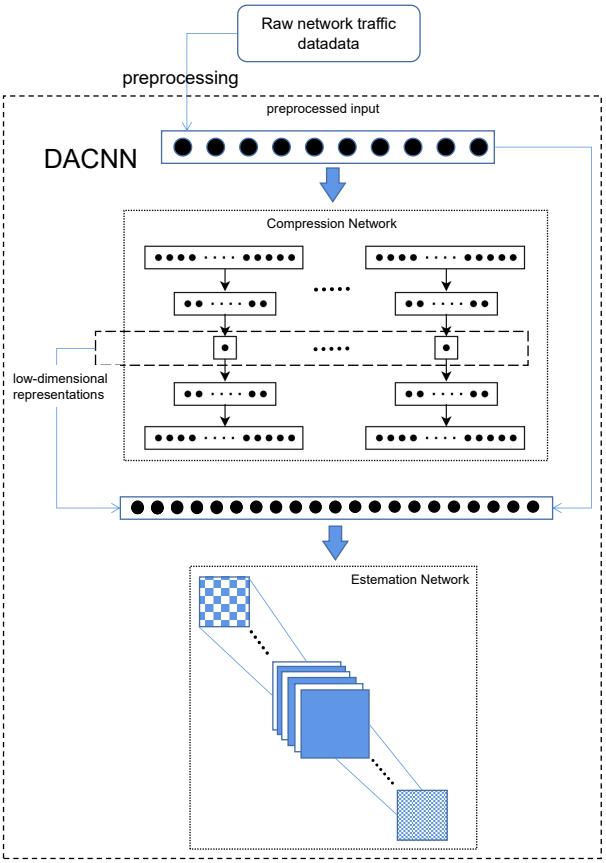

Fig. 1. An overview of DACNN

$$R = [r_0, \ldots, r_n] \tag{2}$$

$para_i$ is the parameters of i-th pretrained deep autoencoders, r_i is the reduced low-dimensional representation learned by the i-th pretrained deep autoencoder, r_i and r_j chould have different numbers of dimensions. $h(.)$ denotes the encoding function. After multiple autoencoder operations, the compression network feeds R to the subsequent estimation network.

2) Why use autoencoder: Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data, A simple and widely used method is principal components analysis (PCA).PCA transforms the original data into a set of linearly independent representations in each dimension through linear transformation,but as a purely mathematical method, PCA has certain limitations. Hinton et al. proposed a multilayer neural network which work much better than PCA as a tool to reduce the dimensionality of data named autoencoder [20].

In this article, we choose the autoencoder as part of the model. The following are the performance results of different pretrained autoencoders on the kddcup99 data set, we randomly selected 1000 pieces of data from the data set as input.

As we can see from Fig2 and Fig3, the abnormal/normal samples have relatively distinct distribution characteristics

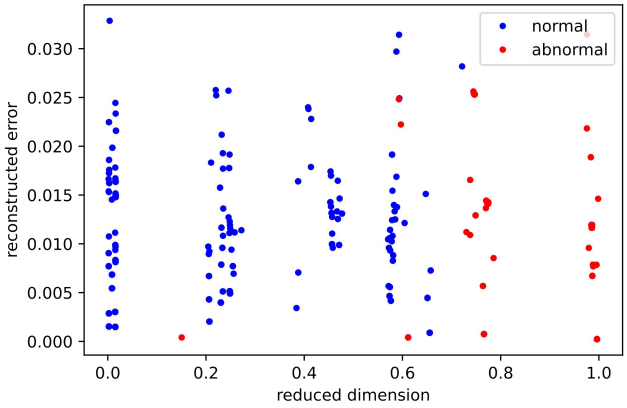

Fig. 2. Low-dimensional representations from autoencoder with reduced 1-dimensional space, the horizontal axis denotes the reduced 1-dimensional space learned by a deep autoencoder, and the vertical axis denotes the reconstruction error induced by the 1-dimensional representation.

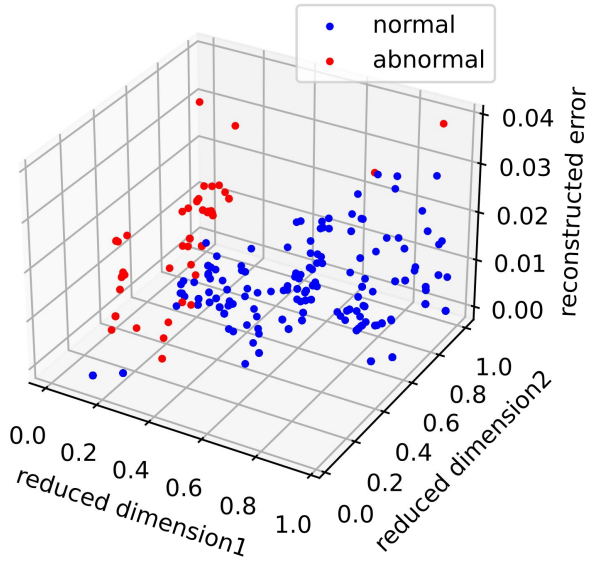

Fig. 3. Just like the Fig2, but low-dimensional representations from another autoencoder with reduced 2-dimensional space.

which proves that the use of autoencoders to obtain dimensionality reduction representations can provide a certain amount of traffic classification information.

C. Estimation Network

Given the low-dimensional representation of the input sample, the estimation network performs classification prediction under a 2d-CNN framework. Take the model built on kddcup as an example, the details of the network architecture are shown in Table I.

We used 4 ”conv-bn-relu” structures in the net. Many graphics-related cnn network structures will adopt a ”conv-pool” architecture, this is because of the close connection between adjacent pixels of the image, but the flow characteristics do not have this feature, so we canceled the local pooling operation. We also replaced fully connected layer with global

TABLE I
NETWORK ARCHITECTURE

layers type	output shape	parms
input	[-1,1,20,20]	/
conv2d	[-1,16,20,20]	160
batchnorm2d	[-1,16,20,20]	32
relu	[-1,16,20,20]	/
conv2d	[-1,32,9,9]	4640
batchnorm2d	[-1,32,9,9]	64
relu	[-1,32,9,9]	/
conv2d	[-1,64,4,4]	18496
batchnorm2d	[-1,64,4,4]	128
relu	[-1,64,4,4]	/
conv2d	[-1,5,2,2]	2885
batchnorm2d	[-1,5,2,2]	10
relu	[-1,5,2,2]	/
global-avg-pool	[-1,5,1,1]	/
softmax	[-1,5]	/

pooling.The strategy of global pooling was proposed by Lin et al. in [21], which is easier to interpret and less prone to overfitting than traditional fully connected layers.

D. Workflow

When we input traffic information to the model, the entire workflow of the DACNN model is shown in the following figure 4.

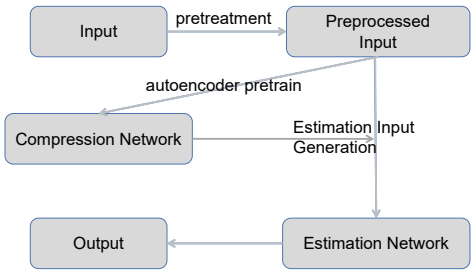

Fig. 4. Intrusion detection workflow

1) Autoencoder pre-train: Given a dataset of N samples, we first pre-train each autoencoder in the compression network. We use the following equation to evaluate the training effect.

$$\sigma = L(x_i, x_i^{'}) \tag{3}$$

x_i and $x_i^{'}$ are the input and the output of the autoencoder. $L(.)$ is the loss function that characterizes the reconstruction error caused by the autoencoder. Intuitively, the reconstruction error low, the low-dimensional representation could better preserve the key information of input samples.

2) Estimation input generation: After pre-training of the compression network, assuming the input is x_i, we get different low-dimensional representations of data generated by different autoencoders like $[r_0,\ldots,r_n]$. The input of the analysis network we constructed is as follows.

$$P_i = [x_i, r_0, x_i, r_1, \ldots, x_i, r_n] \tag{4}$$

$$I_i = resize(P_i, 0\ldots) \tag{5}$$

We alternately splice the data and the obtained dimensionality reduction representation into a one-dimensional vector P_i,and then reconstruct the one-dimensional vector into a two-dimensional vector I_i, if the lenth of vector is not enough, zeros will be added back.Finally, input I_i to the estimation network. We have tried different splicing methods on the data and dimensionality reduction representation. We found that the alternate splicing method $[x_i, r_0, x_i, r_1, \ldots, x_i, r_n]$ works better than direct splicing$[x_i, \ldots, x_i, r_1, \ldots, r_n]$. We guess this is because the classification information contained in the dimensionality reduction representation can interact with the original traffic data earlier and fully(under the same network depth).

IV. Experimental Result

In this section, we use public benchmark datasets to demonstrate the effectiveness of DACNN in anomaly detection.

A. Dataset

TABLE II
STATISTICS OF THE PUBLIC BENCHMARK DATASETS

	#Dimensions	#Instances	#Anomaly ratio
KDDCUP99 train	126	1074992	0.25
KDDCUP99 test	126	311029	0.8
IDS-2017	78	2412107	0.18

We uesd 2 datasets:KDDCUP99 and IDS-2017, and we will conduct the main comparison work based on the kdd dataset.

- KDDCUP:kddcup99 is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. the kddcup99 dataset has a training set and a test set, initially with 41-dimensional features, after onehot encoding, the feature is expanded to 126 dimensions.
- IDS-2017 [22]:CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files). It has 78-demensional features.

B. Performance indicators

In order to evaluate the effectiveness of our proposed method, we propose to use a confusion matrix III to get the performance indicators, in which, positive represents abnormal traffic and negative represents normal traffic. TP represents abnormal traffic that is correctly classified, TN is normal

traffic that is correctly classified, FP is normal traffic that is misclassified, and FN is abnormal traffic that is misclassified.

TABLE III
CONFUSION MATRIX

	predicted	
actual	positive	negative
true	TP	TN
false	FP	FN

- Accuracy(ACC).

$$ACC = \frac{TP + FN}{TP + TN + FP + FN} \tag{6}$$

Although ACC is a widely used classifier performance evaluation indicator. But when the data is unbalanced, the testing dataset is extremely unbalanced, the ACC will be misleading to the researcher.

- Precision(P), Recall(R), F_1.
Precision: calculate how many of the samples we predict are correctly predicted.

$$P = \frac{TP}{TP + FP} \tag{7}$$

Recall: for the original actual samples, how many samples have been correctly predicted.

$$R = \frac{TP}{TP + FN} \tag{8}$$

F_1: F_1 Score is an indicator used to measure the accuracy of a binary classification model in statistics. It is used to measure the accuracy of unbalanced data. It takes into account both the accuracy rate and recall rate of the classification model.

$$F_1 = \frac{2 * P * R}{P + R} \tag{9}$$

- Detection Rate(DR). The DR is represented by the proportion of the number of correctly classified instances to the total number of instances of this type. x_x represents the actual type x and classified as x type, x_x represents the actual any type and classified as type x.

$$DR_x = \frac{x_x}{\sum x_*} \tag{10}$$

C. Configuration

- KDDCUP. Because the data of kdd is extremely unbalanced, we first oversampled it and expanded the amount of data for smaller categories to more than 100,000. After that, onehot encoding was performed on six of the features, and the original 41-dimensional features were expanded to 126 dimensions, and normalized.
The compression network contains 3 autoencoders, Provide 1-dimensional, 2-dimensional and 5-dimensional dimensionality reduction representation respectively, then alternately splicing with multiple raw data to form the input of the estimation network like [x,γ_{1d},x,γ_{2d},x,γ_{5d}], x is the flow data, and γ_* is the reduction representation. In particular, the compression network runs with FC(126, 80,tanh)-FC(80, 40, tanh)-FC(40, γ, tanh)-FC(γ, 40, tanh)-FC(40, 80,tanh)-FC(80, 126, none), γ represents a dimensionality reduction representation of the number of different dimensions. The dimensionality reduction representations of the three autoencoders are respectively [1, 2, 5]. Then we get a 386-dimensional vector like (4), and resize it into a 20*20 two-dimensional vector after adding zeros as the estimation network input.
The estimation network performs like Table I.

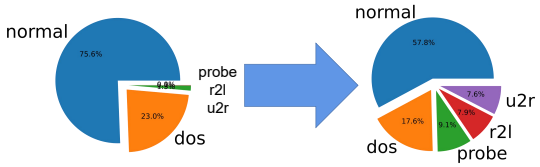

Fig. 5. Oversample the data of KDDCUP99

- IDS-2017.For this dataset, we simply perform two classifications of normal and abnormal, we normalize the data, and divide the data set into training and test sets, of which 20% is the test set. The compression network contains 4 autoencoders.
The compression network runs with FC(78, 50,tanh)-FC(50, 20, tanh)-FC(20, γ, tanh)-FC(γ, 20, tanh)-FC(20, 50,tanh)-FC(50, 126, none), γ represents a dimensionality reduction representation of the number of different dimensions which are respectively [1, 2, 3, 4].Then we get a 322-dimensional vector like (4), and resize it into a 20*20 two-dimensional vector after splicing an original traffic data again, The input form before resize is as shown in [x,γ_{1d},x,γ_{2d},x,γ_{3d},x,γ_{4d},x], x is the flow data, and γ_* is the reduction representation.
The estimation network structure is in Table IV.

TABLE IV
NETWORK ARCHITECTURE

layers type	output shape	parms
input	[-1,1,20,20]	/
conv2d	[-1,16,20,20]	160
batchnorm2d	[-1,16,20,20]	32
relu	[-1,16,20,20]	/
conv2d	[-1,32,9,9]	4640
batchnorm2d	[-1,32,9,9]	64
relu	[-1,32,9,9]	/
conv2d	[-1,64,4,4]	18496
batchnorm2d	[-1,64,4,4]	128
relu	[-1,64,4,4]	/
conv2d	[-1,2,2,2]	1154
batchnorm2d	[-1,2,2,2]	10
relu	[-1,2,2,2]	/
global-avg-pool	[-1,2,1,1]	/
softmax	[-1,2]	/

TABLE V
STATISTICS OF THE PUBLIC BENCHMARK DATASETS ON KDDCUP99

	DR_{normal}	DR_{dos}	DR_{probe}	DR_{r2l}	DR_{u2r}	ACC
KDDCup99 winner [3]	**0.995**	0.971	0.833	0.084	0.132	0.927
KDDCup99 runner-up [4]	0.994	**0.975**	0.845	0.073	0.118	0.929
P-DNN [19]	0.964	0.968	0.886	0.213	0.272	0.931
PNrule [25]	0.995	0.969	0.732	0.107	0.066	0.925
Multi-Classifier [26]	/	0.973	**0.887**	0.096	0.298	/
Decision Trees [27]	0.994	0.966	0.779	0.005	0.136	0.928
Naive Bayes [27]	0.977	0.967	0.883	0.087	0.110	0.921
MOGFIDS [28]	0.984	0.972	0.886	0.111	0.158	0.927
baseCNN	0.915	0.958	0.755	0.089	0.224	0.912
DACNN	0.972	0.957	0.886	**0.231**	**0.664**	**0.935**

$FC(a, b, f)$ means a fully-connected layer, a is input neurons and b is output neurons, f is the activate function.

All the instances are implemented by pytorch [23] and trained by Adam [24] algorithm with learning rate 0.0001. For KDDCUP and IDS-2017, the number of training epoch are 5 and 3 respectively, and the size of mini-batches are 128 and 256 respectively.

D. BaseLine Method

We considered a cnn model (baseCNN) like the estimation network in DACNN(Table I), without any input from the compression network as baseline. After training under the same data(kddcup99) and parameters, we compared the results of two models(baseCNN, DACNN) based on kddcup99 dataset in Table VI.

TABLE VI
DACNN COMPARED WITH BASELINE METHOD

	P	R	F_1	ACC
$baseCNN$	0.923	0.978	0.950	0.912
$DACNN$	**0.926**	**0.992**	**0.958**	**0.935**

It can be seen from the Table VI that the precision, recal, F_1, and accuracy values of DACNN are significantly improved compared to ordinary cnn without a compression network. This result proves that the classification information contained in the dimensionality reduction representation of the compression network has a positive effect on the performance of the model.

E. Comparision

The comparison in Table V included recent research using the KDD Cup 99 dataset and achieving satisfactory results.Evaluation indicators include ACC and DR.

It can be seen from the Table V that the classification results of DACNN have achieved excellent results in multiple classifications, especially in r2l and u2r type, which are far superior to other classification methods.This shows that the model has better classification results for data sets with extremely unbalanced data volume, and can effectively identify extreme data, reflect the excellent intrusion detection performance of the model. From the perspective of the indicator DR, compared with other research works, the method we proposed can not only get satisfactory results on DR_{normal}, DR_{probe}, and DR_{dos} but also make DR_{u2r} and DR_{r2l} achieve some improvement.

We also used IDS-2017, another commonly used data set in the cyber security field, for testing. It can be seen from the Table VII that the DACNN model proposed in this paper still performs well on the IDS-2017 dataset when compared with other methods.Here we use some commonly used traditional machine learning methods, such as KNN(K-NearestNeighbor), RF(Random Forest), qda(Quadratic Discriminant Analysis) and Adaboost to compare with the baseCNN and DACNN.

TABLE VII
DACNN COMPARED WITH TRADITIONAL MACHINE LEARNING METHODS ON IDS-2017

	P	R	F_1
KNN	0.96	0.96	0.96
RF	0.98	0.97	0.97
Adaboost	0.77	0.84	0.77
QDA	0.97	0.88	0.92
baseCNN	0.98	0.97	0.97
DACNN	**0.98**	**0.98**	**0.98**

V. CONCLUSION

Convolutional neural network can automatically abstract high-level features from raw data, learn the internal laws of samples, and has good adaptability to massive high-dimensional data. Therefore we propose an effective intrusion detection method based on deep autoencoding convlutional neural network: DACNN. We built a model architecture composed of two networks of compression and estimation. First, we pre-trained multiple autoencoder models with different compression levels, combined the output with the original data, and fed the estimation network for training to get the most excellent model. Finally, the results on multiple datasets show that our model can effectively extract the corresponding traffic characteristic information. This model combines the unsupervised learning model and the supervised learning model, shows good detection performance in the experiment(ACC increased to 0.935 in KDDCUP) and has good versatilitys.

In view of the traffic density characteristics of the current big data network era, we will consider further improving its real-time detection capabilities for traffic based on the DACNN model in the next step.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (62002028, 62102040), NSFC-General Technology Fundamental Research Joint Fund U1836215, China Postdoctoral Science Foundation under Grant 2020M680464, and Capital Science and Technology Leading Talent Training Project, China (Z191100006119030).

REFERENCES

[1] "Internet trends 2019." https://www.bondcap.com/report/it19/.

[2] L. Heberlein, B. Mukherjee, K. Levitt, G. Dias, and D. Mansur, "Towards detecting intrusions in a networked environment," 1991.

[3] B. Pfahringer, "Winning the KDD99 classification cup: Bagged boosting," *SIGKDD Explor.*, vol. 1, no. 2, pp. 65–66, 2000.

[4] I. Levin, "KDD-99 classifier learning contest: Llsoft's results overview," *SIGKDD Explor.*, vol. 1, no. 2, pp. 67–75, 2000.

[5] V. Miheev, A. Vopilov, and I. Shabalin, "The MP13 approach to the kdd'99 classifier learning contest," *SIGKDD Explor.*, vol. 1, no. 2, pp. 76–77, 2000.

[6] C.-H. Tsang, S. Kwong, and H. Wang, "Anomaly intrusion detection using multi-objective genetic fuzzy system and agent-based evolutionary computation framework," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 4 pp.–, 2005.

[7] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[8] L. Liu, M. Hu, C. Kang, and X. Li, "Unsupervised anomaly detection for network data streams in industrial control systems," *Inf.*, vol. 11, no. 2, p. 105, 2020.

[9] M. Al-Qatf, L. Yu, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[10] 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.

[11] S. N. Mighan and M. Kahani, "Deep learning based latent feature extraction for intrusion detection," in *Electrical Engineering (ICEE), Iranian Conference on*, pp. 1511–1516, IEEE, 2018.

[12] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 178–183, 2018.

[13] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security on neuromorphic computing system," in *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pp. 3830–3837, IEEE, 2017.

[14] Y. Yu, J. Long, and Z. Cai, "Session-based network intrusion detection using a deep learning architecture," in *Modeling Decisions for Artificial Intelligence - 14th International Conference, MDAI 2017, Kitakyushu, Japan, October 18-20, 2017, Proceedings* (V. Torra, Y. Narukawa, A. Honda, and S. Inoue, eds.), vol. 10571 of *Lecture Notes in Computer Science*, pp. 144–155, Springer, 2017.

[15] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based ddos detection system in software-defined networking (sdn)," *ICST Transactions on Security and Safety*, vol. 4, p. 153515, Dec 2017.

[16] A. Abeshu and N. Chilamkurti, "Deep learning: The frontier for distributed attack detection in fog-to-things computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018.

[17] Z. R. HANG Mengxin, CHEN Wei, "Abnormal flow detection based on improved one-dimensional convolutional neural network," *journal of Computer Applications*, vol. 41, no. 2, p. 8, 2021.

[18] L. P. C. Z. ZHANG Yanhui, LYU Na, "An encrypted traffic classification method based on convolutional attention gated recurrent networks," *Journal of Signal Processing*, vol. 37, no. 7, p. 9.

[19] M. Lei, X. Li, B. Cai, Y. Li, L. Liu, and W. Kong, "P-dnn: An effective intrusion detection method based on pruning deep neural network," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2020.

[20] Hinton, G., E., Salakhutdinov, R., and R., "Reducing the dimensionality of data with neural networks.," *Science*, 2006.

[21] M. Lin, Q. Chen, and S. Yan, "Network in network," *Computer Science*, 2013.

[22] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, 2018.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *CoRR*, vol. abs/1912.01703, 2019.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[25] R. C. Agarwal and M. V. Joshi, "Pnrule: A new framework for learning classifier models in data mining (A case-study in network intrusion detection)," in *Proceedings of the First SIAM International Conference on Data Mining, SDM 2001, Chicago, IL, USA, April 5-7, 2001* (V. Kumar and R. L. Grossman, eds.), pp. 1–17, SIAM, 2001.

[26] M. Sabhnani and G. Serpen, "Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context," in *Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications. MLMTA'03, June 23 - 26, 2003, Las Vegas, Nevada, USA* (H. R. Arabnia and E. B. Kozerenko, eds.), pp. 209–215, CSREA Press, 2003.

[27] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs decision trees in intrusion detection systems," in *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, March 14-17, 2004* (H. Haddad, A. Omicini, R. L. Wainwright, and L. M. Liebrock, eds.), pp. 420–424, ACM, 2004.

[28] C. Tsang, S. Kwong, and H. Wang, "Anomaly intrusion detection using multi-objective genetic fuzzy system and agent-based evolutionary computation framework," in *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), 27-30 November 2005, Houston, Texas, USA*, pp. 789–792, IEEE Computer Society, 2005.