Research on ensemble model of anomaly detection based on autoencoder

Yaning Han
University of Chinese Academy of Sciences
Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences
Beijing, China
toxiaoning@163.com

Yunyun Ma
Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences
Beijing, China
myy@csu.ac.cn

Jinbo Wang
Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences
Beijing, China
wangjinbo@csu.ac.cn

Jianmin Wang
Technology and Engineering Center for Space Utilization, Chinese Academy of Sciences
Beijing, China
wangjm@csu.ac.cn

***Abstract*—In the fields of technology such as aerospace, anomaly detection is critical to the overall system. With the large increase in data volume and dimensions, the traditional detection methods have great limitations, and thus anomaly detection algorithms based on deep learning have received widespread attention. In this paper, based on autoencoder: standard autoencoder, denoising autoencoder, and sparse autoencoder, an ensemble detection model that can extract more feature information is proposed. To make more use of these feature information, inspired by the idea of pooling layer of the CNN, two feature fusion methods are proposed. Finally, the experiment verifies that the result of this model is better than the single autoencoder model.**

Keywords—anomaly detection, autoencoder, ensemble

I. INTRODUCTION

The Internet and information technology are developing rapidly. In the face of complex network environments, the amount of data is growing at an extremely fast rate, and data processing technology is also developing rapidly, which has also caused more and more network security exceptions. People build network security precautions from various levels, such as data packet encryption, access authentication, and anomaly detection. Data packet encryption and access authentication are passive defenses, which can detect and block most intrusions, but cannot detect and block intrusions inside the network system. Anomaly detection can analyze the information of the network system and can detect the intrusion behavior inside the system. Therefore, compared to data packet encryption and access authentication, anomaly detection can more effectively ensure network security.

Currently, scholars and engineers need to develop some methods to find valuable information in massive data and hidden safety hazards in the network. Data mining has gained the attention. It is based on statistical methods and uses machine learning, artificial intelligence and other technologies to analyze massive data and extract useful information.

II. RELATED WORK

So far, many experts and scholars have done a lot of research on anomaly detection problems and improved a lot of anomaly detection methods. Common methods of anomaly detection are clustering and support vector machine in data mining. The clustering algorithm has fast detection speed, but the false detection rate is relatively high [17]. Support vector machine (SVM) is a common machine learning classification model [18], however the performance of support vector machine is greatly affected by hyperparameters. For inexperienced scholars, the hyperparameters cannot be reasonably combined and the results cannot be guaranteed.

Neighbor-based methods assume that normal data has relatively more neighbors than the outlier data. In [19], Breunig adopted a density-based local outlier factor (LOF) to address this issue. In [20], Kriegel proposed the local outlier probabilities to detect outliers. However, the search for the nearest neighbors prohibits such methods to be applied to high-dimensional data due to the curse of dimensionality. High dimensional data will fool the algorithm to locate the improper neighbors, which will decrease the detection accuracy.

In practical applications, massive data and ultra-high dimensions have become important challenges for anomaly detection. The most common method for working with data dimensions is dimensionality reduction. The principal component analysis (PCA) method is representative of the dimensionality reduction method. However, in the face of ultra-high feature space, calculation of covariance matrix requires a lot of computational cost and time cost[5]. With the development of neural networks, deep learning technology has been received widespread attention. The accuracy of the neural network detection is relatively high, and accordingly the quality of the training data is higher. In [6], Li proposed a deep learning approach for intrusion detection using a multi-convolutional neural network (multi-CNN) fusion method. According to the correlation, the feature data are divided into four parts, and then the one-dimensional feature data are converted into a grayscale graph.

Although these traditional intelligent diagnosis methods such as Neural Network (NN) and Support vector machine (SVM) can obtain accurate diagnostic results, they have two inherent drawbacks: 1) They cannot generate features automatically, and fault features need to be designed by experts manually. 2) In practical applications, since the signal has non-linear, non-Gaussian or other characteristics, a lot of pre-analysis and comparison processes are required[7]. With the boosting of artificial intelligence, autoencoder provides an effective way to learn representative features, which overcomes the above drawbacks of manual feature extraction. In [9], Sakurada proposed an Autoencoder-based outlier

978-1-7281-8913-0/20/$31.00 ©2020 IEEE
DOI 10.1109/QRS51102.2020.00060

detection method. As autoencoder can capture the nonlinear correlations as well as the linear correlations, this method has better detection performance than PCA-based. However, when applying the autoencoder to image outlier detection, the detection performance is not always very palatable. This is because the single autoencoder fails to fully capture the correlations among features, especially in the high-dimensional datasets, and resulting in poor detection accuracy. Therefore, the autoencoder based on the ensemble came into being. In [10], Chen proposed a novel image outlier detection method by combining autoencoder with Adaboost (ADAE). By ensembling many weak autoencoders, the method can better capture the statistical correlations among the features of normal data than the single autoencoder. Therefore, the proposed ADAE is able to determine the outliers efficiently.

This paper is based on ensemble multiple autoencoders for feature extraction.

III. Proposed Model

A. Standard Autoencoder (AE)

Autoencoder is an unsupervised artificial neural network that is trained for automatic feature extraction Through training, the weight and bias parameters of each hidden layer can be adjusted under the optimal output to obtain different representations of the input (each layer represents a representation), So-called higher-order features. Studies show that automatic learning methods can greatly improve the accuracy, and thus achieve better classification results than other traditional classification algorithms. This method is called an autoencoder. The structure of the autoencoder is shown in Fig. 1.

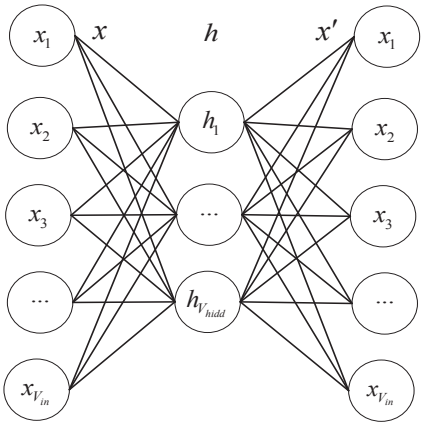

Fig. 1. The structure of AE

The autoencoder encodes the input x to obtain a new feature h. Assume that the original input x can be reconstructed from the new feature h. The encoding process is as follows:

$$h = f\left(W_e x + b_e\right) \tag{1}$$

W_e is a weight vector and b_e is a constant. Like the neural network structure, its encoding is a linear combination followed by a non-linear activation function. If there is no non-linear activation function, then the autoencoder is no different from ordinary PCA. With the new feature h, the input x can be reconstructed, that is, the decoding process:

$$x' = g\left(W_d h + b_d\right) \tag{2}$$

We want the reconstructed x' and x to be as same as possible, and this model can be trained using a loss function that minimizes negative log-likelihood:

$$Loss = -\log P\left(x \mid x'\right) \tag{3}$$

P is the loss function of x and x'. Sometimes we add more constraints or penalties to autoencoder, such as denoising autoencoder and sparse autoencoder. Because it is not meaningful to simply reconstruct the original input most of the time, we hope that the autoencoder can capture more valuable information of the original input in the case of approximate reconstruction of the original input.

B. Denoising Autoencoder (DAE)

Due to the existence of noise and outliers in the actual data, it is still difficult to learn the robust sample features using the above method to improve the applicability of the autoencoder. In order to force the hidden layer to acquire more robust features, a method of introducing noise to reconstruct the original input signal has been used to train the autoencoder. that is the denoising autoencoder(DAE).

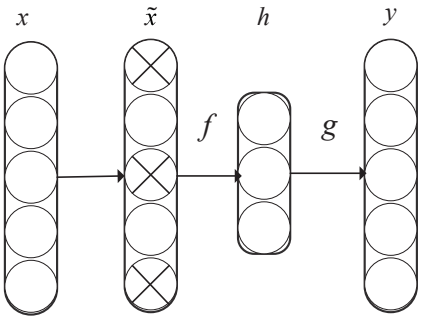

Fig. 2. The structure of DAE

As shown in Fig. 2, after introducing random noise, the input signal $\tilde{x}$ follows the distribution $\tilde{x} \sim q(\tilde{x} \mid x)$. the signal that reaches the hidden layer(4).

$$h = f(\tilde{x}) = f\left(W_e \tilde{x} + b_e\right) \tag{4}$$

$$y = g(h) = g\left(W_d h + b_d\right) \tag{5}$$

then (5) reconstructs the input data, and finally trains the parameters to minimize the error by (6).

$$L_H(x, y) = \left\| y - x \right\|^2 \tag{6}$$

so the model output y gradually approaches the real input x.

C. Sparse Autoencoder (SAE)

SAE is also based on AE. Make some restrictions on the hidden layer, such as reducing the number of neurons, the neural network will compress the data and extract features. When the penalty term is added to make the neuron inactive in most cases, the network can learn very interesting edge features.

Firstly, the objective function of the SAE is given as (7).

$$J_{sparse}(W,b)=\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right)-y^{(i)}\right\|^2\right)+P_1+P_2 \tag{7}$$

$$P_1=\frac{\lambda}{2}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\Theta_{ji}^{(l)}\right)^2 \tag{8}$$

$$P_2=\beta\sum_{j=1}^{s_2}KL\left(\rho\|\hat{\rho}_j\right) \tag{9}$$

(9) is a new penalty for the hidden layer. s_2 represents the number of neurons in the hidden layer. $\hat{\rho}_j$ represents the average activation degree of hidden layer neuron j for all training data.

$$\hat{\rho}_j=\sum_{i=1}^{m}\left[a_j^{(2)}\left(x^{(i)}\right)\right] \tag{10}$$

SAE can restrict the activation degree of each neuron node by adding sparseness restrictions, making the fault features extracted from the original input data more robust and less affected by interference factors such as noise, so to a certain extent Improve accuracy.

D. The proposed Ensemble Autoencoder(EAE) for anomaly detection

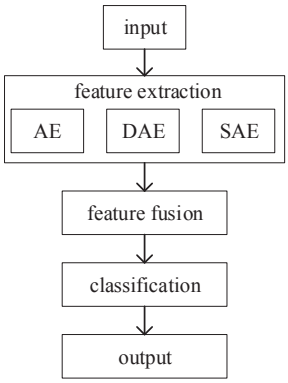

Fig. 3. The flowchart of EAE

This section introduces the proposed EAE method for anomaly detection in detail, and proposes the feature fusion method by analogy to the pooling layer of convolutional neural network(CNN).

EAE integrates three types of autoencoders above: AE, DAE and SAE in Fig. 3. Each autoencoder has the functions of dimensionality reduction and automatic feature extraction.

For CNN, the convolutional layer is mainly used to extract features. This paper uses autoencoder to extract features. The size and quality of the features by different convolution kernels are different, so the pooling layer is introduced. Essentially, the pooling layer is a feature fusion between multiple channels. Through the pooling layer, not only can the feature map be reduced, but also the amount of calculation can be reduced by reducing network parameters, and overfitting can be controlled to a certain extent. Similar to the pooling layer of a convolutional neural network, after the feature is extracted by the autoencoder, we also propose and define three feature fusion methods: feature non-fusion or single feature (SF), max feature (MF) and average feature (AF) in Fig. 4.

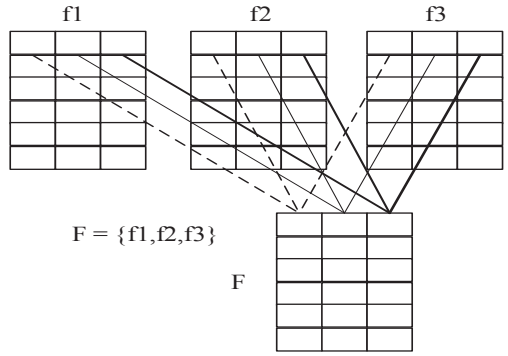

Fig. 4. Feature fusion

As shown in Fig. 4, assuming three encoders get three feature maps:f1, f2 and f3, all of which are $n*m$. For each feature map, $i,j\ (0\le i\le n-1, 0\le j\le m-1)$ represents row and column respectively. The feature fusion methods are as follows:

SF: No processing the feature map as (11).

$$\mathrm{F}(i,j)=\mathrm{f}_1(i,j)\ \text{or}\ \mathrm{f}_2(i,j)\ \text{or}\ \mathrm{f}_3(i,j) \tag{11}$$

MF: Take the largest feature at the same location as (12).

$$\mathrm{F}(i,j)=\max(\mathrm{f}_1(i,j),\ \mathrm{f}_2(i,j),\ \mathrm{f}_3(i,j)) \tag{12}$$

AF: Mean of the same location feature as (13). m is the dimension of these feature maps.

$$\mathrm{F}(i,j)=\frac{\mathrm{f}_1(i,j)+\mathrm{f}_2(i,j)+\mathrm{f}_3(i,j)}{m} \tag{13}$$

Finally get $\mathrm{F}()$ and input to the classifier. The overall experimental scheme is shown in Fig. 5. (SVM: Support Vector Machine, LR: Logistics Regression, KNN: k-Nearest Neighbor)

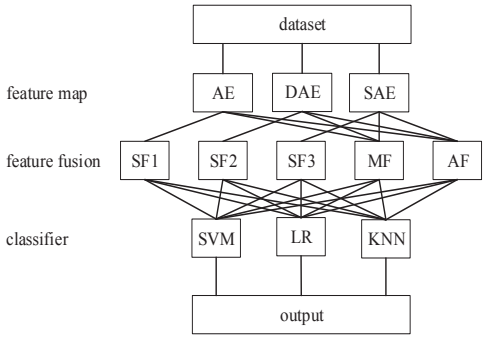

Fig. 5. Experimental schema

IV. Evaluations and Results

The proposed model is validated on some credit card fraud detection dataset with three autoencoders and three classifiers. In order to balance the positive and negative proportions, the positive samples are down-sampled and the negative samples are up-sampled. The data set is randomly divided into the train set and the test set according. normalize the train data and adjust network parameters to make the autoencoder reach the optimal state. The results are shown in TABLE I.

TABLE I. Results

	Accuracy		
	SVM	***LR***	***KNN***
SF1	0.784	0.775	0.79
SF2	0.796	0.817	0.809
SF3	0.811	0.803	0.812
MF	0.84	0.848	0.853
AF	0.845	0.854	0.837

As TABLE I, the accuracy of MF and AF are better than that of SF. It shows that the ensemble model with feature fusion extracts and uses more feature information than the single model (SF can be regarded as the single model). The generalization performance is better. In addition, the results of SF2 and SF3 are better than SF1, with less loss of feature information. This proves that the addition of penalty terms and denoising operation has helped improve the results. Because all the autoencoders use a single hidden layer, the accuracy results are not very high. If multiple hidden layers are used, the result may be higher.

V. Conclusion

In anomaly detection, this paper proposes an ensemble model based on autoencoder, and uses some data to conduct experiments. The verification results show that the results of the ensemble model are better than the single model, and more feature information is used. Autoencoder has better automatic feature extraction capability and can be widely used in more fields.

Acknowledgment

The relevant research done in this paper are supported by the Equipment Pre-Research Field Fund, China (No.61400020401).

References

[1] Yang, Q., et al, "Prediction of aptamer-protein interacting pairs based on sparse autoencoder feature extraction and an ensemble classifier," Mathematical Biosciences 311: 103-108, 2019.

[2] Shi, R., et al, "Boosting sparsity-induced autoencoder: A novel sparse feature ensemble learning for image classification," International Journal of Advanced Robotic Systems 16(3), 2019.

[3] Principi, E., et al, "Unsupervised Electric Motor Fault Detection by Using Deep Autoencoders," Ieee-Caa Journal of Automatica Sinica 6(2): 441-451, 2019.

[4] Li, Y., et al, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion." Measurement 154, 2020.

[5] Zhang, B., et al, "Network Intrusion Detection Based on Stacked Sparse Autoencoder and Binary Tree Ensemble Method," 2018 Ieee International Conference on Communications Workshops, 2018.

[6] Li, Y., et al, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," Measurement 154, 2020.

[7] Zhang, Y., et al, "Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method," Measurement 151, 2020.

[8] K. Tidriri, N. Chatti, S. Verron, T. Tiplica, "Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges," Annual Reviews in Control, 42 (2016) 63-81.

[9] M. Sakurada, T. Yairi, "Anomaly detection using autoencoders with non-linear dimensionality reduction," Proceedings of the MLSDA 2014 2ndWorkshop on Machine Learning for Sensory Data Analysis, ACM, 2014.

[10] Chen, Z., et al, "Evolutionary multi-objective optimization based ensemble autoencoders for image outlier detection," Neurocomputing 309: 192-200, 2018.

[11] Chen, L., et al, "Multiperiod-Ahead Wind Speed Forecasting Using Deep Neural Architecture and Ensemble Learning," Mathematical Problems in Engineering, 2019.

[12] Hu, J., et al, "An Efficient and Robust Unsupervised Anomaly Detection Method Using Ensemble Random Projection in Surveillance Videos," Sensors 19(19), 2019.

[13] Ibrahim, A. K., et al, "Classification of red hind grouper call types using random ensemble of stacked autoencoders," Journal of the Acoustical Society of America 146(4): 2155-2162, 2019.

[14] Khan, S. S. and B. Taati, "Detecting unseen falls from wearable devices using channel-wise ensemble of autoencoders," Expert Systems with Applications 87: 280-290, 2017.

[15] Wen, L., et al, "A New Snapshot Ensemble Convolutional Neural Network for Fault Diagnosis," Ieee Access 7: 32037-32047, 2019.

[16] Xu, F., et al, "Roller bearing fault diagnosis using stacked denoising autoencoder in deep learning and Gath-Geva clustering algorithm without principal component analysis and data label," Applied Soft Computing 73: 898-913, 2018.

[17] I. Chairunnisa, Lukas, and H. D. Widiputra, "Clustering base intrusion detection for network profiling using k-means, ECM and k-nearest neighbor algorithms," Konferensi Nasional Sistem dan Informatika 2009, nov 2009.

[18] Y. Bengio, "Learning deep architectures for AI," Foundations and Trends in Machine Learning, 2009,2(1): 1–127.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, "Lof: identifying density-based local outliers," ACM Sigmod Record, Vol. 29, ACM, 2000, pp.93–104.

[20] H.-P. Kriegel, P. Kr¨ oger, E. Schubert, A. Zimek, "LoOP: local outlier prob-abilities," Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 1649–1652.