A Deep One-class Model for Network Anomaly Detection

Songlin Dai1  Jubin Yan2  Xiaoming Wang2  Lin Zhang2
1 State Grid SiChuan Electric Power Company, Chengdu,610000, China
2 Chengdu Chengdian Electric Power Engineering Design Co. Ltd, Chengdu 610000, China
Lin Zhang, psplayer@126.com

Abstract. For traditional network anomaly detection system, the detection performance is related to the selected features and training dataset. But traditional methods adopt handcraft feature selection, which requires heavy human labour and relies on the experts' knowledge and experience. Besides, the collected dataset for training is not balanced, which makes the prediction of the trained model tends to be biased to the majority class. In this paper, a one-class network anomaly detection model based on the stacked autoencoders was proposed. We use the stacked autoencoders to select the prominent features from the raw collected data, then apply the one-class classification algorithm support vector data description to train a classifier to identify the network traffic into normal data and anomalous data. The experimental results demonstrate the promising results of our approach for network anomaly detection.

1. Introduction
Nowadays, the network has become an indispensable part of people's daily life. At the same time, the exponential growth of cyber-attacks has become a major threat to network security. For the users of the network, network security is an unavoidable problem, inadvertent negligence may lead to irreparable damage to them. Therefore, the need for cyber defense has become more and more urgent.

There are two main network security technologies [1]: firewall and intrusion detection system. Firewall is a passive security technology, which is designed to restrict and control the flow of traffic between the out network and the protected network and monitors the unauthorized data transmission to and from the protected network. Potential intruders can be blocked by installing a firewall at the edge of the protected network boundaries. But there is a limitation for a firewall that firewall cannot protect against insider attacks. If a packet allowed to the protected network contains malicious code, the entire network will be infected whether you install a firewall or not. Another technology is the intrusion detection system, and it is an effective complement to firewall technology. Intrusion detection is an active technology which monitors the behaviors of a host or a network and alerts the administrator when it detects the suspicious network behavior.

In general, the detection approaches for intrusion detection system can be divided into two categories: misuse detection and anomaly detection.

Misuse detection defines and collects the abnormal patterns firstly, then it will check every incoming and outgoing packet. Any action that conforms to the abnormal pattern is considered intrusive. The main advantage of misuse detection is its high detection accuracy, and any intrusion can be detected accurately if its pattern is collected in the rule library of the intrusion detection system. But misuse detection systems perform very poorly to detect new attacks [2–8]. Therefore, there are few intrusion detection systems use misuse detection approach alone.
On the other hand, the approach of anomaly detection is based on the normal behavior of the protected network rather than the patterns which are used in the approach of misuse detection. Any behavior which deviates from the established normal pattern will be regarded as an intrusion. Compared with the approach of misuse detection, the network anomalies and traffic anomalies generated by network attacks cannot be concealed in the approach of anomaly detection. Any attack can be detected if there establishes a suitable detection model. However, anomaly detection has a high positive false rate which makes anomaly detection cannot be effectively applied to the real network environment [9].

In order to address the above-mentioned problem, an improved anomaly detection method was proposed in this paper. The main contributions of the new proposed method in this paper are as follows:

1) Unlike the traditional methods regarding the anomaly detection as a multi-class classification task, we consider anomaly detection as a one-class classification paradigm and only collect the normal and legal data from the protected network to define the boundary about legal behavior of the protected network. So, in the method of the paper, there is no need to label the data and deal with the problem of the imbalanced dataset.

2) To the problem of the hand-craft feature selection, we introduce the stacked autoencoders in our anomaly detection method, which can filter the inessential features and select the meaningful features to represent the network data.

3) We performed extensive experiments which convincingly demonstrate the effectiveness of the method proposed in this paper.

The rest paper structure is as follows. The related works on network anomaly detection were provided in section 2. In section 3, a hybrid model for anomaly detection which is based on deep autoencoders and one-class SVM was proposed. The experiment and experiment result analysis were described in section 4. We conclude in section 5.

2. Related works
In this section, previous related works about network anomaly detection were reviewed before the approach proposed in this paper introduced.

Network anomaly detection refers to the method of finding the network packets which don’t conform to the expected network behavior pattern [10]. The main reason for these anomalies existing in the network is that the purpose and behaviors of hackers using the network are different from those of the normal user. The mainstream approaches to network anomaly detection can be divided into two categories: supervised network anomaly detection and unsupervised anomaly detection.

For the supervised based methods, the instances of the dataset have been labeled for normal as well as the anomaly class. A predictive model is built based on the labeled dataset, and unseen instances are judged by the model to determine which class they belong to. The advantage of supervised based anomaly detection is the high accuracy in their capability to detect the attacks[9,11]. But there two problems lie in supervised based methods. First, the data in the dataset need to be labeled. The annotation for the dataset is often done manually by a human which is time-consuming and labor intensive, moreover, the quality relies on the domain knowledge and experience of the annotators and it is a challenging task to obtain accurate labels for the data [12]. Second, because there are significantly fewer attack packets compare to the normal packets in the collected dataset. The judgment criterion which is trained by the machine learning model tends to be biased to the majority class and the minority class network behaviors are inclined to mislead model to incorrect judgment.

Unsupervised anomaly detection methods do not need the labeled train dataset and study rules from the grouping similar data instances [13]. The approaches for unsupervised anomaly detection usually are clustering algorithms, such as KNN [14], K-means [12] and etc. There is a basic assumption that normal network instances are far more frequent than the anomalies in the test data for the unsupervised anomaly detection methods. if this assumption is not true, these methods will supper from high false rate [15].

There is an issue both for supervised anomaly detection and unsupervised anomaly detection. It is the problem of hand-crafted features selection. Because the collected data is usually associate with a lot
of noise. It is an essential task to select a subset of highly discriminant features before train the model for anomaly detection. The most used approach for features selection is hand-crafted selection, which requires heavy human labor and the quality of the selected features relies on the experts’ knowledge and experience.

3. Our model
The proposed model exploits the nonlinear mapping abilities of stacked autoencoders to extract the prominent features, then the extracted features are taken as input to the one-class SVM identify the anomalies occurrences.

The architecture of the model proposed in this paper is illustrated in figure 1. There are three main layers of the model. Data preprocessing is the first layer of the model and it transforms raw data into a format that will be more easily and effectively processed. The layer of Stacked autoencoders is the second layer. It extracts the saint features from the data which is from the first layer. OCSVM is the third layer and it is a response to identify whether the input data is normal or not. The mechanics of the classification tasks is relatively straightforward: First, any network packet collected from the protected network will be processed by the layer of the data preprocessing and transform the raw data to the specified format. Then the preprocessed data will be mapped into a compact and informative feature vector by the layer of the stacked autoencoders. Finally, the layer of the OCSVM will verify whether the given network is normal or not. In the following subsections, the different components of the model are presented.

![Figure 1 The architecture for the model proposed in this paper](image)

3.1 Layer data pre-processing
Because the collected data from the protected network is susceptible to noise, data preprocessing is required to improve the quality of the raw data.

There are two main data types for the collected data. One is numerical, another is categorical or string. So, there are main two data preprocessing operations in the layer data preprocessing. For the type of categorical, one-hot encoder [16] is used to map the value of the data into numeric data. For example, the protocol value of the network packet has three categories: TCP, UDP and ICMP, and these values can be mapped into three vectors (0 0 1) (0 1 0) and (1 0 0). For the type of numerical, the min-max normalization [17] was applied to fit the data in a predefined boundary.

3.2 Layer stacked autoencoders
The power of deep learning lies in its ability that it can learn the different representations of the raw data layer by layer. Every layer can extract more abstract and complex features based on the former layer. The mechanics for stacked autoencoders is the same.
A stacked autoencoders consists of multiple layers of autoencoders, and every autoencoder is a kind of unsupervised learning structure which has three layers: input layer, hidden layer, and an output layer.

There are two parts in the autoencoder: encoder and decoder. Encoder transforms the raw data into low-dimensional representation. And decoder reconstructs the raw data based on the data which is generated by the encoder. Give a dataset \( X = \{X_1, X_2, X_3, \cdots, X_n\} \), where \( X_i \) is a \( d \)-dimensional feature vector. The encoder maps the raw data \( X_i \) from \( d \)-dimensional vector to \( m \)-dimensional (\( m < d \)) vector. The encoding process in Equation (1).

\[
H_i = S(WX_i + b) \tag{1}
\]

Where \( W \) is a \( m \times d \) weight matrix, \( m \) is the node numbers of the hidden layer, \( b \) is the bias vector and function \( S \) is the encoding function.

The decoder is the reversible process of the encoder. It transforms the result of the hidden representation \( H_i \) to reconstructed the input data \( X_i \). The decoding process is as follows:

\[
\hat{X}_i = D(W'H_i + b') \tag{2}
\]

Where, \( W' \) is a \( d \times m \) weight matrix, \( b' \) is the bias vector and \( D \) is the decoding function.

![Figure 2](image)

Figure 2: The training process schematic diagram for the stacked autoencoders (3-layer)

The goal of training autoencoder is to find the optimized parameter sets \( \{W, W', b, b'\} \) to minimize the difference between the input data \( X_i \) and the output data \( \hat{X}_i \). The loss calculation process can be described as follows:

\[
L(X_i, \hat{X}_i) = \frac{1}{n} \sum_{i=1}^{n} ||X_i - \hat{X}_i||^2 \tag{4}
\]

And the optimizing process is as follows:

\[
\theta = \{W, W', b, b'\} = \arg\min L(X_i, \hat{X}_i) \tag{5}
\]

As described above, the stacked autoencoders contain multiple autoencoders in the hidden layer and the training for the stacked autoencoders should be step by step. In our model, greedy layer-wise training [18] was applied to obtain good parameters for every layer. Illustrated by figure 2, the whole training process for the stacked autoencoders can be divided into three phrases [19]:

1) Using the raw data to train the first autoencoder and obtain the trained vector;

2) Take the formerly trained vector as input data to train the subsequent layer. Repeat this procedure until all the autoencoders have trained.

3) Use the backpropagation algorithm to minimize the loss function and update the weight matrixes and bias vectors to achieve fine-tuning.

3.3 Layer One-class classification

For the problem of the network anomaly detection, it is very difficult to obtain the attack instance compared to obtain the normal instances. The collected dataset for training classifier has unequal instances for a different class, which bias the prediction of the trained model towards the more common case. Learning the predicted model from unbalanced data is a one-class classification problem.

There are two main methods can be applied to the one-class classification problem. One is density estimation with parametric generative models based on the training data, such as the Gaussian model.
[20], Gaussian mixture model [21] or density estimation [22]. This method assumed that the data is distributed according to the normal distribution or to a mixture of Gaussian distributions and obtained the parameters using maximizing the likelihood function over the data. The advantage of method density estimation is that the model is very simple and is easy to implement. But this method is sensitive to the noise of the training data [23]. Another method is boundary method which defines a hypersphere with a minimum volume to cover the whole data of the dataset, such as one-class support vector machine (OCSVM) [24] or support vector data description (SVDD) [25] and this method is relatively resistant to noise. OCSVM and SVDD are very similar, and they are equivalent in the case that all sample \( s \) lie on a hypersphere centered at the origin. But the hyper-sphere model of the SVDD can be more flexible than that of the OCSVM if introducing kernel function [26]. So, layer one-class classification uses SVDD to verify whether the input data instance is normal or not.

For the given input dataset \( X = \{x_1, x_2, \cdots, x_l\}, x_i \in \mathbb{R}^n \), the aim of the SVDD is to find spherically shaped boundary around dataset \( X \). The problem can be defined as follows:

\[
\min \ R^2 + C \sum_{i=1}^{l} \xi_i
\]

\[
s.t. \| \phi(x_i) - a \|^2 \leq R^2 + \xi_i, i = 1, 2, \ldots, l
\]

(6)

Where, \( \alpha \) is center and \( R \) is radius of the hyperspherical model, \( \phi \) is a function which maps the input data to a higher dimensional space, \( \xi_i \) is the slack variables and \( C > 0 \) is a user-specified parameter which gives a trade-off between simplicity and the number of errors. If the error rate threshold is set, the parameter \( C \) is computed as:

\[
C = \frac{1}{\text{threshold}}
\]

(7)

Its Lagrange dual problem is Eq. (8).

\[
\max \sum_{i=1}^{l} a_i k(x_i, x_i) - \sum_{i,j} a_i a_j k(x_i, x_j)
\]

\[
s.t. \sum_{i=1}^{l} a_i = 1, \ 0 < a_i \leq C
\]

(8)

Where, \( a_i \) is Lagrange multipliers, \( K \) is the kernel function. For the dual problem, \( a_i = 0 \) means that the sample is in the hyperspherical, \( a_i = C \) means the sample is outside the hyperspherical and \( 0 < a_i < C \) means that the sample is on the hyper spherical. For any sample \( x \), its distance to the center of the hyper spherical is as follows:

\[
r^2 = (x \cdot x) - 2 \sum_{i} a_i (x \cdot x_i) + \sum_{i,j} a_i a_j (x_i, x_j)
\]

(9)

Therefore, we can use a discrimination function to determine whether a test point \( x \) is within the sphere. The discrimination function is as follows:

\[
f(x) = I[(x \cdot x) - 2 \sum_{i} a_i (x \cdot x_i) + \sum_{i,j} a_i a_j (x_i, x_j) \leq R^2]
\]

(10)

Where \( I(\cdot) \) is an index function. If \( I(\cdot) = 1 \), it means the test point \( x \) is in or on the hyper spherical. If \( I(\cdot) = 0 \), it means the test point \( x \) is outside of the hyper spherical.

4. Experiments

4.1 Dataset
We evaluate the model proposed in this paper on the well-known NSL-KDD dataset [27] which includes 125973 train and 22543 test records labeled anomaly and normal. The NSL-KDD dataset consists of 41 features and one class attribute. The details for NSL-KDD is given [30].

4.2 Evaluation Measures
In this paper, we use accuracy which is defined as the percentage of correctly classified instances over the total instance to evaluate the anomaly classifier. The computation of accuracy is as follows:

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
\]  

(11)

Where TP is true positive which is the number of anomalous data that is correctly classified as attack, TN is true negative which is the number of the normal data that is correctly classified as normal, FP is false positive which is the number of the normal data that is incorrectly classified attack, and FN is false negative which is the number of the attack data that is incorrectly classified as normal.

4.3 Feature selection and model training
The NSL-KDD dataset has 41 features with 38 numeric and 3 categorical features. We used min-max normalization and one-hot encoder to preprocess these features.

In our model, the encoded representations which are learned by the layer of the stacked autoencoders are used to train predictor to identify the network traffic into normal behaviors and anomalous behaviors. In order to have an efficient topology for both experiments, we employed a greedy layer-wise unsupervised learning algorithm proposed by Hinton et al [19] to train the layer for stacked autoencoders. The trained layer for the stacked autoencoders with 41-28-28-16-28-28-41 structure which is resulted from experimented with numerous structural compositions to achieve the best results.

After training of the layer for the stacked autoencoders, bottleneck layer reduces the dimensionality from 41 to 16-dimensional feature space. The layer for one-class classification was trained to identify the normal packets and anomalous packets. Unlike the layer for the stacked autoencoders using unsupervised learning, the layer for one-class classification uses the labeled data to train the predictor.

4.4 Performance Evaluation

Table 1. The overall performance of the three main methods and our approach.

| Classification method | Feature selection method | Number of the selected attributes | Detection accuracy |
|-----------------------|--------------------------|----------------------------------|--------------------|
| J48                   | Correlation              | 8                                | 68.39%             |
|                       |                          | 16                               | 73.67%             |
|                       |                          | 21                               | 78.19%             |
|                       | Info Gain                | 8                                | 78.05%             |
|                       |                          | 16                               | 78.18%             |
|                       |                          | 23                               | 80.96%             |
| SVM                   | Correlation              | 8                                | 68.03%             |
|                       |                          | 16                               | 71.61%             |
|                       |                          | 21                               | 74.38%             |
|                       | Info Gain                | 8                                | 74.64%             |
|                       |                          | 16                               | 72.34%             |
|                       |                          | 23                               | 74.32%             |
| Bayesian network      | Correlation              | 8                                | 65.58%             |
|                       |                          | 16                               | 70.49%             |
|                       |                          | 21                               | 75.61%             |
|                       | Info Gain                | 8                                | 71.05%             |
|                       |                          | 16                               | 72.63%             |
|                       |                          | 23                               | 72.10%             |
| Our approach          | The stacked autoencoders | 16                               | 91.32%             |
In order to evaluate the efficiency of our model proposed in this paper. We compare our approach with J48 decision tree [28], support vector machine [29] and Bayesian Network [30] with correlation-based feature selection [31], information Gain [32] two feature selection strategies. Table 1 demonstrates the overall performance of the above-mentioned methods and our approach.

Table 1 shows comparative experiments of four types of classification tasks on NSL-KDD. From the table 1, the approach proposed in this paper shows a very promising performance and achieved 91.32% accuracy rate in the binary classification as compare to the best accuracies achieved by J48 (80.96%), SVM (74.64%) and Bayesian network (75.61%), which proves the efficiency of our approach.

5. Conclusion
In this paper, a hybrid anomaly detection model which combined deep autoencoders and one-class SVM was proposed. The deep autoencoders are trained as an automatic feature selection and dimensionality reduction algorithm, which generate nonlinear changes for the raw data and transform these data into a lower dimensional set of features. Then, the trained features are taken as input to the one-class SVM to train the anomaly detection classifier. The experimental result on the benchmark dataset NSL-KDD demonstrates the effectiveness of the hybrid model proposed in this paper.

References
[1] Rai K, Devi M S and Guleria A 2018 Packet-based Anomaly Detection using n-gram Approach
[2] Muddu S, Tryfonas C and Bulusu R P 2018 Sharing model state between real-time and batch paths in network security anomaly detection (Google Patents)
[3] Zhou M, Wang Y, Srivastava A K, Wu Y and Banerjee P 2018 Ensemble based algorithm for synchrophasor data anomaly detection IEEE Transactions on Smart Grid
[4] Angelov P P and Gu X 2019 Applications of Autonomous Anomaly Detection Empirical Approach to Machine Learning (Springer) pp 249–59
[5] Habeeb R A A, Nasaruddin F, Gani A, Hashem I A T, Ahmed E and Imran M 2018 Real-time big data processing for anomaly detection: A Survey International Journal of Information Management
[6] Aljawarneh S, Aldwairi M and Yassein M B 2018 Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model Journal of Computational Science 25 152–60
[7] Asghar N, Ahmad W and Ashraf R 2018 A Comparative Study of Data Mining Algorithms for High Detection Rate in Intrusion Detection System Annals of Emerging Technologies in Computing (AETiC) 2
[8] Hamed T, Ernst J B and Kremer S C 2018 A survey and taxonomy of classifiers of intrusion detection systems Computer and network security essentials (Springer) pp 21–39
[9] Fernandes G, Rodrigues J J, Carvalho L F, Al-Muhtadi J F and Proença M L 2019 A comprehensive survey on network anomaly detection Telecommunication Systems 70 447–89
[10] Atli B G, Miche Y, Kalliola A, Oliver I, Holtmanns S and Lendasse A 2018 Anomaly-based intrusion detection using extreme learning machine and aggregation of network traffic statistics in probability space Cognitive Computation 10 848–63
[11] Ernst J, Hamed T and Kremer S 2018 A survey and comparison of performance evaluation in intrusion detection systems Computer and network security essentials (Springer) pp 555–68
[12] Wu J, Liu H, Xiong H, Cao J and Chen J 2015 K-means-based consensus clustering: A unified view IEEE transactions on knowledge and data engineering 27 155–69
[13] Buczak A L and Guven E 2016 A survey of data mining and machine learning methods for cyber security intrusion detection IEEE Communications Surveys & Tutorials 18 1153–76
[14] Deng Z, Zhu X, Cheng D, Zong M and Zhang S 2016 Efficient kNN classification algorithm for big data Neurocomputing 195 143–8
[15] Mohri M, Rostamizadeh A and Talwalkar A 2018 Foundations of machine learning (MIT press)
[16] Mou L, Jia R, Xu Y, Li G, Zhang L and Jin Z 2016 Distilling word embeddings: An encoding approach Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (ACM) pp 1977–80
[17] Patro S and Sahu K K 2015 Normalization: A preprocessing stage arXiv preprint arXiv:1503.06462
[18] Miotto R, Li L, Kidd B A and Dudley J T 2016 Deep patient: an unsupervised representation to predict the future of patients from the electronic health records Scientific reports 6 26094
[19] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks science 313 504–7
[20] Williams C K and Rasmussen C E 2006 Gaussian processes for machine learning vol 2 (MIT Press Cambridge, MA)
[21] Nasrabadi N M 2007 Pattern recognition and machine learning Journal of electronic imaging 16 049901
[22] Criminisi A, Shotton J and Konukoglu E 2012 Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning Foundations and Trends® in Computer Graphics and Vision 7 81–227
[23] Scott D W 2015 Multivariate density estimation: theory, practice, and visualization (John Wiley & Sons)
[24] Shang W, Li L, Wan M and Zeng P 2015 Industrial communication intrusion detection algorithm based on improved one-class SVM 2015 World Congress on Industrial Control Systems Security (WCICSS) (IEEE) pp 21–5
[25] Chalapathy R and Chawla S 2019 Deep Learning for Anomaly Detection: A Survey arXiv preprint arXiv:1901.03407
[26] Khan S S and Madden M G 2009 A survey of recent trends in one class classification Irish conference on artificial intelligence and cognitive science (Springer) pp 188–97
[27] Revathi S and Malathi A 2013 A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection International Journal of Engineering Research & Technology (IJERT) 2 1848–53
[28] Bhargava N, Sharma G, Bhargava R and Mathuria M 2013 Decision tree analysis on j48 algorithm for data mining Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering 3
[29] Suykens J A and Vandewalle J 1999 Least squares support vector machine classifiers Neural processing letters 9 293–300
[30] Friedman N, Geiger D and Goldszmidt M 1997 Bayesian network classifiers Machine learning 29 131–63
[31] Yu L and Liu H 2003 Feature selection for high-dimensional data: A fast correlation-based filter solution Proceedings of the 20th international conference on machine learning (ICML-03) pp 856–63
[32] Stachniss C, Grisetti G and Burgard W 2005 Information gain-based exploration using rao-blackwellized particle filters. Robotics: Science and Systems vol 2 pp 65–72