A Deep Learning Approach for Network Intrusion Detection Based on NSL-KDD Dataset

Chongzhen Zhang, Fangming Ruan*, Lan Yin, Xi Chen, Lidong Zhai, Feng Liu
School of Big Data and Computer Science, Guizhou Normal University, Guiyang, China
1655143922@qq.com; 921151601@qq.com; 6329511@qq.com; 544196776@qq.com; zhailidong@ict.ac.cn; liufeng@iie.ac.cn

***Abstract*—Along with the high-speed growth of Internet, cyber-attack is becoming more and more frequent, so the detection of network intrusions is particularly important for keeping network in normal work. In modern big data environment, however, traditional methods do not meet requirement of the network in the aspects of adaptability and efficiency. A approach based on deep learning for intrusion detection was proposed in this paper which can be applied to deal with the problem to certain extent. Autoencoder, as a popular technology of deep learning, was used in the proposed solution. The encoder of deep autoencoder was taken to compress the less important features and extract key features without decoder. With proposed approach one can build the network and identify attacks faster, the benchmark NSL-KDD dataset can be evaluated with proposed model.**

Keywords—network intrusion detection; deep learning; autoencoder; NSL-KDD

I. INTRODUCTION

Intrusion is a series of actions that seek to damage the integrity, confidentiality or availability of information resources[1]. Network Intrusion Detection System (NIDS) is a reasonable supplement technology for firewall, which helps network security administrators to better ensure network security. Currently, the data analysis methods for summarizing various intrusion detection classified into two classes: misuse detection (signature-based detection) and anomaly detection. (1) Misuse detection, which is based on a known signature library formed by a network attack, matches the processed input data and signature library to determine whether it is an intrusion behavior. (2) Anomaly detection, comparing the activities to be detected with normal activities, and considers that it may be an intrusion behavior for activities that violate the statistical rules of normal activities. The former detection method shows low false alarm rate and high false negative rate, for a new intrusion behavior, it needs to be added to the signature library to identify the attack. The latter method does not need to analyze and extract features from each attack, and a new attack can be detected by using this detection method, but the detection result has a lower false negative rate and a higher false alarm rate. It poses a serious challenge to these traditional detection methods while new attack pattern arise.

There are two main challenges to the detection of attacks. First, network traffic is large and produces rapidly. Processing these large volumes of network data requires more effective data analysis techniques, and deep learning is an efficient data analysis technology widely used whthin various domains. Second, the feature selection problem, the variety of attacks pattern and the suitability of different scenarios. There are many methods of machine learning[2] applied to intrusion detection, such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forest (RF), etc. These classification techniques have greatly improved the accuracy of detection. Similarly, feature selection helps eliminate unrelated features and noise, generates a classification model with better classification performance and easier understanding, and selecting key attributes is a good way to reduce unnecessary data processing[3]. Formerly, when machine learning was used for real tasks, the features of describing samples were usually designed by domain experts, and specialized knowledge was needed to process data. The quality of features had a vital impact on generalization performance. It was uneasy to design good features, but also prone to make mistakes.The deep learning approach is designed to learn better feature representations from a great number of unlabeled data, and then apply these learned features to the classification.

Getting some resolution of the problem above mentioned is necessary. A deep learning method was proposed in this paper for NIDS, by which features can be learned and adjusted itself according to previously undefined attacks. In particular, the encoder of autoencoder to learn the features of the input data and extract the key features. Soft-max regression classification was applied at last. Using NSL-KDD intrusion detection benchmark dataset was evaluated to prove the usability of model proposed in this paper. This simple network architecture, as the result, can accurately predict a large amount of data in a short time.

II. RELATED WORK

Recently, deep learning and its applications have garnered great attention within various research fields, such as drug research[4],smart manufacturing[4], automotive design[6].

In this section, we will discuss related work in the application of deep learning within the NIDS domain. It is noted to state that we only discuss the work of evaluating the model performance with the NSL-KDD dataset, which is convenient to compare.

Yin et al. [7] used a recurrent neural network model in deep learning. This method was compared with other machine learning methods, such as J48, ANN, SVM, RF, which were applied to intrusion detection methods evaluated on NSL-KDD benchmark dataset. The experimental results show that the

978-1-7281-2458-2/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: AALTO UNIVERSITY. Downloaded on October 11,2021 at 07:06:07 UTC from IEEE Xplore. Restrictions apply.

performance is better than the previous classification method, and the model has a good performance in both binary and multi-class classification, and can be classified higher accuracy. Shone et al. [8] designed a stacked non-symmetric deep autoencoder, using random forest algorithm as classifier, and the novel deep learning classification model used for unsupervised feature learning. The model achieved high classification accuracy, precision, recall and f1-score while also reducing the required training time. Javaid et al. [9] proposed a Self-learning method combining sparse autoencoder and soft-max regression classification. The model is evaluated by using NSL-KDD benchmark dataset for 2-class, 5-class and 23-class, respectively. The results show that the model showed good performance. Potluri et al. [10] presented a method using Deep Neural Network (DNN) with all 41 features as input and three hidden layers (two autoencoders and one soft-max). The experimental results achieved higher accuracy, and those fewer classes are more accurate than with more classes. Tang et al. [11] proposed a method for monitoring network flow data, The authors claim that the classification accuracy based on the 6 features is 75.75% evaluated the proposed model with NSL-KDD dataset.

From the above work, it shows that most researchers are still trying to combine various network structures and classification algorithms to construct suitable and efficient for NIDS. Although it achieves high detection accuracy, there is still room for improvement. Moreover, the model can achieve high accuracy for the test data set, but when analyzing the actual network traffic, the accuracy always seems to be reduced. Therefore, we believe that the simple network framework proposed in below section can provide a solution or a new attempt for the current research domain.

III. NSL-KDD DATASET

This experiment uses the NSL-KDD benchmark dataset, which is improved version of the KDD Cup 99 dataset that overcomes some problems that have been discussed in detail in [12] and is better for evaluating the model we designed than the KDD CUP 99 dataset. Although this dataset still has the limitations discussed in [13], the researchers involved in the domains of intrusion detection are still in use, so we think this dataset is valid.

The dataset contains normal traffic and different kinds of Abnormal traffic. It can be classified into five categories: Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), Probing Attack, Normal, and its composition is detail shown in TABLE I.

Each record in the original NSL-KDD dataset contains 41 features, some of which cannot be better represented during the training process. We need to pre-process the data and extend three symbolic features of protocol_type, service, and flag with 1-N encoding. After processing, the data contains 122 features including 3 protocol_types, 70 services, and 11 flags. Then we normalize the NSL-KDD dataset to the range of [0, 1] with a max-mix operation.

TABLE I. COMPOSITION OF DATASET

Category		Training set	Test set
Attacks	*Dos*	45927	7458
	U2R	52	67
	R2L	995	2887
	Probe	11656	2421
Normal		67343	9711
Total		125973	22544

IV. MODEL DESIGN

A. Autoencoder

Deep learning develops from the artificial neural network of machine learning. It can represent complex relationships, concepts in multi-layers and learn from different levels. These levels can be seen as two components, the basic units of lower levels and the high-level concepts defined by these basic layers. Low-level combinations constitute a high level, and can also form higher-level concepts, which in turn help define higher-level concepts and express more complex relationships[14].

Autoencoder is a popular technique in deep learning, a type of unsupervised neural network, which takes vector as input and also takes a vector of the same dimension as output. By acquiring input, changing dimension, and reconstructing output, the data created in this process can be represented in higher or lower dimensions, and finally a similar result of input and output is expected, i.e. $X^{\sim} = X$, an example of a simple autoencoder is shown in Fig. 1. Relu function, $f(z) = \max(0, z)$, is used for activation, $h_{W,b}(x)$ used for the calculation of nodes in the hidden and output layers, as shown in equation (1).

$$h_{W,b} = f(Wx + b) \tag{1}$$

Here, *h* is a non-linear function with parameter weight W and bias b, which needs to be initialized.

The cost function minimized the reconstruction error in learning process is represented by equation (2), it tries to learn an output $X^{\sim}$ is similar to X.

$$J = \frac{1}{2m} \sum_{i=1}^{m} |x_i - \tilde{x_i}|^2 \tag{2}$$

These types of neural networks are very useful and its benefits are compression coding and feature learning in an unsupervised form. By compressing the input features and then attempting to restore the input, an effective model can be obtained through continuous training. When the dimension of the hidden layer is smaller than that of the input and output layers, it is used to encode the data (i.e. feature compression), reducing the computational resources required to build the model[15].

Authorized licensed use limited to: AALTO UNIVERSITY. Downloaded on October 11,2021 at 07:06:07 UTC from IEEE Xplore. Restrictions apply.

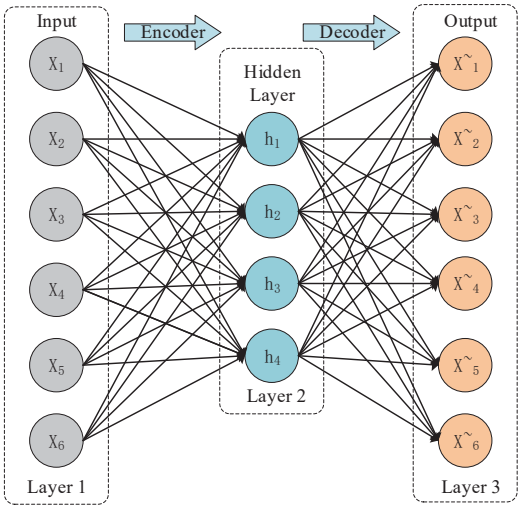

Figure 1. An example of a simple autoencoder.

We propose a autoencoder-based approach that uses only the encoder in the autoencoder without decoder, which helps us compress features, extract key features, build faster, simpler networks, and save computing resources, and make faster prediction of abnormal traffic.

B. Classification Model

Our model, as shown in Fig. 2, first preprocesses the data to be trained, then inputs the processed data into the encoder. The encoder learns the complex relationship between different features, extracts the key features, and finally classifies and predicts them by a soft-max regression.

The model we constructed has 1 input layer, 5 hidden layers and 1 output layer. According to the idea of encoder compression feature, the input of 122 features is reduced to 5 features and then classified. The detailed architecture is shown in Fig. 3.

The input layer fed all 122 features, and the hidden layer 1 selects 64 of the 122 features from the input data. The remaining hidden layers perform this operation in sequence until the learning feature is reduced to 5 at the end, and the hidden layer 5 is soft-max layers, these features are used as inputs to the softmax layer (it can be fine-tuned using supervised learning when we training model), the soft-max layer predicts and classifies the input data, and finally the output layer gets the results. These hierarchical network stacks implement a complete network model.

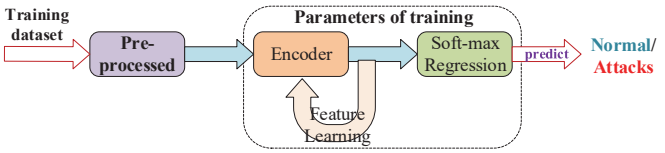

Figure 2. Classification model.

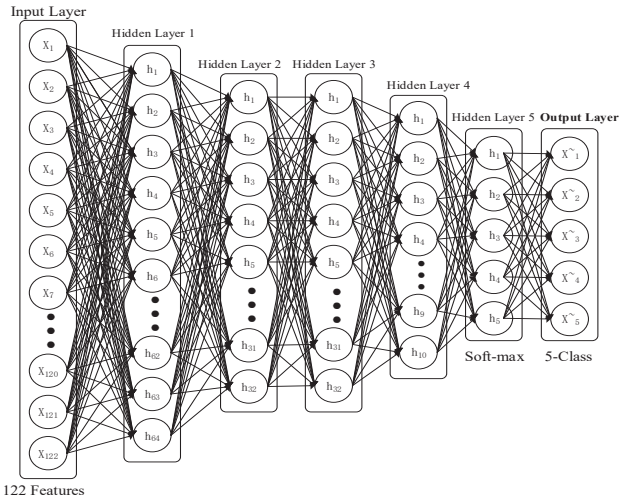

Figure 3. Detailed architecture for parameter training.

V. Experimental Results

A. Evaluation Metrics

The metrics we use in this section are based on the following definitions:

- True Positive (TP) - correctly classify attacks as attacks.
- False Position (FP) - incorrectly classifies the attack as normal.
- True Negative (TN) - correctly classify normal as normal.
- False Negative (FN) - incorrectly classifies normal as an attack.

Our proposed model will be using the below measures for evaluation:

Accuracy (Acc): measures the proportion of correct classification.

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Precision (Pr): the sample identified by the model is the proportion of true positive samples.

$$\text{Pr} = \frac{TP}{TP + FP} \tag{4}$$

Recall (Re): the ratio of positive samples are identified by model as positive samples.

$$\text{Re} = \frac{TP}{TP + FN} \tag{5}$$

F1-score (F1): the harmonic mean of precision and recall.

$$\text{F1} = 2 \cdot \frac{Pr \cdot Re}{Pr + Re} \tag{6}$$

Authorized licensed use limited to: AALTO UNIVERSITY. Downloaded on October 11,2021 at 07:06:07 UTC from IEEE Xplore. Restrictions apply.

Accuracy can evaluate the practicability of the model, but when unbalanced sample data is encountered, the model cannot be properly evaluated, so it is necessary to add precision and recall for the evaluation. High accuracy and high recall have different emphasis in practical applications. The F1-score is the harmonic mean of precision and recall. It reflects the generalization ability of the model. The model with high F1 - score is better applied to practical problems.

B. Performance Evaluation

The experiment is to evaluate the classification model performance within 5 classifications (Dos, Normal, Probe, R2L, U2R), the classification model tested 22544 samples, and the confusion matrix generated according to the experimental results is shown in Fig. 4.

From a dataset of 9711 normal samples, 9423 normal samples were correctly identified, while 8554 of the remaining 12833 attacks were correctly identified. According to equation (3) - (6), Accuracy, Precision, Recall, F1-score can be obtained, which is detailed in TABLE II. Because of the limited samples available for training, the detection rate of R2L and U2R attacks within five classes is low, this reduces the overall accuracy of NIDS.

We compare it with other similar methods deep learning-based to ensure that our model is effective.

In [9], the authors claim that they produced a 75.76% f1-score in the 5-class classification using NSL-KDD dataset. The result of recall and precision are not given, but their bar graph shows that they are about 69% and 83%, respectively. Our Precision is 82.22%, however, our model achieves better f1-score and recall values of 76.47% and 79.47%, respectively.

Tang et al. [11] claimed that their DNN method achieved 75.75% accuracy in performing the 5 classes based on the NSL-KDD dataset. This result is lower than the 79.74% accuracy we achieved.

The above comparison shows that our results are still very promising.

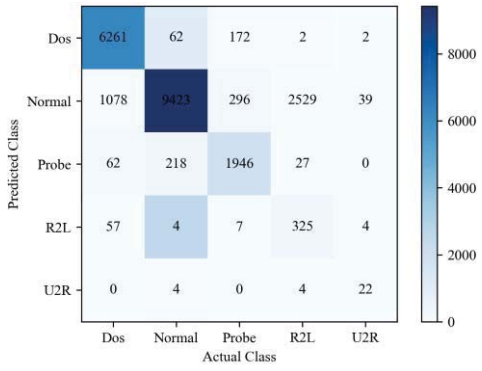

Figure 4. Confusion matrix.

TABLE II. 5-CLASS PERFORMANCE

Category	Accuracy	Precision	Recall	F1-score
Dos	83.95%	96.34%	83.95%	89.72%
Normal	97.03%	70.51%	97.03%	81.67%
Probe	80.38%	86.37%	80.38%	83.27%
R2L	11.26%	81.86%	11.26%	18.79%
U2L	32.84%	73.33%	32.84%	46.36%
Total	79.74%	82.22%	79.74%	76.47%

VI. CONCLUSION

A deep learning approach for NIDS has been proposed to detect any type of intrusion behavior in network. The method can also be used to network learn and adjust itself to the pattern undefined previously. Possibility of false alarm and false negative can be reduced with the method proposed. Taking encoder of autoencoder benefit to construct a network, simple network structure can make prediction more quickly, and is an exploration of a large-flow modern network. In upcoming job some effort will be made for improvement of detection accuracy to higher degree, modifying and adding other classifiers to existing models, and considering the problem of real-time detections.

ACKNOWLEDGMENT

This work was supported by Guizhou Province Project of Innovation Talents Teams of Electrostatic and Electromagnetic Protection (No.[2016]5653) , by Academician Liu Shanghe Fund of Electrostatic Protection Research (Grant No.BOIMTLSHJD20161004).

REFERENCES

[1] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The architecture of a network level intrusion detection system," Los Alamos National Lab., NM (United States), New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science, 1990.

[2] A. L. Buczak, and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol.18, pp.1153-1176, 2015.

[3] MAM, Hasan, M. Nasser, S. Ahmad, and K. I. Molla, "Feature selection for intrusion detection using random forest," Journal of Information Security, vol.7, pp.129-140, 2016.

[4] A. Aliper, S. Plis, A. Artemov, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov, "Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data," Molecular Pharmaceutics, vol.13, pp.2524-2530, 2016.

[5] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: methods and applications," Journal of Manufacturing Systems, vol.48, pp.144-156, 2018.

[6] F. Falcini, G. Lami, and A. M. Costanza, "Deep learning in automotive software," IEEE Software, vol.34, pp.56-63, 2017.

[7] C Yin, Y Zhu, J Fei, and X He. "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol.5, pp.21954-21961, 2017.

[8] 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, pp.41-50, 2018.

Authorized licensed use limited to: AALTO UNIVERSITY. Downloaded on October 11,2021 at 07:06:07 UTC from IEEE Xplore. Restrictions apply.

[9] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp.21-26, 2016.

[10] S. Potluri, and C. Diedrich, "Accelerated deep neural networks for enhanced Intrusion Detection System," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, pp.1-8, 2016.

[11] T.A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE, pp.258-263, 2016.

[12] M. Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE, pp.1-6, 2009.

[13] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactions on Information and System Security (TISSEC), vol.3,pp.262-294, 2000.

[14] L. Deng, and D. Yu, "Deep learning: methods and applications," Foundations and Trends® in Signal Processing, vol.7, pp.197-387, 2014.

[15] Y. I. Bengio, J. Goodfellow, and A. Courville, "Autoencoders" in Deep Learning, MIT Press, Cambridge, MA, pp.499-523, 2016.

Authorized licensed use limited to: AALTO UNIVERSITY. Downloaded on October 11,2021 at 07:06:07 UTC from IEEE Xplore. Restrictions apply.