A Deep Belief Networks intrusion detection method based on Generative Adversarial Networks

Yi Wei
China Tobacco Guangxi Industrial Co., Ltd
Nanning, China
weiyi58830@gxzy.cn

Yingxiong Nong*
China Tobacco Guangxi Industrial Co., Ltd
Nanning, China
00083@gxzy.cn

Zhe Li
China Tobacco Guangxi Industrial Co., Ltd
Nanning, China

Jian Pan
China Tobacco Guangxi Industrial Co., Ltd
Nanning, China

Zhenyu Yang
China Tobacco Guangxi Industrial Co., Ltd
Nanning, China

Zezhang Zheng
School of Computer, Electronics and Information
Guangxi University
Nanning, China

Ningjiang Chen
School of Computer, Electronics and Information, Guangxi University
Guangxi Intelligent Digital Services Research Center of Engineering Technology
Key Laboratory of Parallel, Distributed and Intelligent Computing (Guangxi University)
Nanning, China

2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT) | 979-8-3503-8555-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/AINIT61980.2024.10581576

***Abstract*—In response to the substantial threat that Internet attacks pose to data center network security, researchers have proposed several deep learning-based methods for detecting network intrusions. However, while algorithms are constantly improving in terms of accuracy, their stability in the face of insufficient attack samples is a major obstacle. To solve the issues of insufficient attack samples and low detection accuracy in network intrusion detection, this paper proposes a deep confidence network intrusion detection method G-DBN based on GAN. The model is based on the malicious sample extension of the generative adversarial network, and it can produce adversarial samples using malicious network flows as original samples. Furthermore, this paper uses deep belief network technology to create and assess the efficacy of the G-DBN model in detecting network attacks, comparing it to standard DBN models and other network intrusion detection techniques. Experimental results show that compared to the standard three-layer DBN method, the G-DBN method described in this paper improves the detection accuracy of attack samples by 6.46% and better meets the performance requirements of current practical applications.**

Keywords—Intrusion detection, Generative Adversarial Networks, Deep Belief Networks

I. INTRODUCTION

Data centers frequently hold and handle massive amounts of sensitive information such as personal data, trade secrets, and organizational data, making cyber breaches to organizations and governments extremely damaging. It can cause data breaches, service disruptions, data tampering, malware proliferation, and network backdoors[1], all of which have a detrimental impact on an organization's reputation, economy, and operations. As a result, protecting data centers against network intrusions is a significant responsibility that necessitates suitable security measures as well as the implementation of effective network intrusion detection and prevention technologies[2].

Traditional intrusion detection systems use signatures from known attacks[3]. They are intended to detect a single attack or a group of strikes based on their unique properties. However, because a vast amount of data must be analyzed each day to create new signatures, creating signatures on time is a difficult operation[4]. At the same time, the researchers investigated behavioral analytic strategies for detecting intrusion attacks. The majority of these systems use machine learning approaches to simulate system behavior and detect malicious activities[5]. Typical classification machine learning techniques can be used to train malicious behavior models based on existing data sets of system activity markers, which can be binary (i.e., normal or malicious) or multi-class[6], with multi-class classification allowing the model to detect attack types based on classes/tags in the training set. However, traditional machine learning algorithms have some shortcomings in network intrusion detection, such as data imbalance, idea drift, difficulties processing high-dimensional data, resistance to sample attacks, and insufficient interpretation[7].

To address the problems of unbalanced data, difficulty processing high-dimensional data, and low detection accuracy in traditional machine learning, this paper proposes a multi-classification network intrusion detection method based on deep confidence networks that incorporates generative adversarial network technology. Compared to prior studies, this study offers an attack detection and adversarial sample generation approach that uses malicious network flow as the original sample, and the created adversarial sample can effectively ensure the sample's executability and aggression. At the same time, the experimental results of this work show that, in the case of effective sample generation, G-DBN has a significant improvement in the detection effect of a few categories of samples, and can further meet the practical application needs of data centers.

The remainder of this paper is organized as follows. Section 2 introduces the most recent work on NIDS. Section 3 presents the overall structure and implementation strategy based on G-DBN. In Section 4, the model architecture and experimental findings from the CICIDS2017 dataset are examined. Finally, in Section 5, we conclude this paper and suggest directions for further investigation.

II. Related work

Various machine learning methods have been investigated for the development of NIDS. Among them, the support vector machine (SVM) is the most commonly used one. Chitrakar et al. [8] combined support vector machines with ensemble learning to increase the generalization ability of the model and improve its performance on unknown data samples. However, both the fuzzy C-means (FCM) clustering method and the k-fold cross-validation method used to train and test the final support vector machine may have problems when applied to network traffic time series data sets.

Recent studies have shown that deep learning techniques are very effective in identifying cyberattacks on networks [9]. Ahmad et al. [10] selected the best elements from the data set according to the correlation between the elements, proposed an Adaboost-based network intrusion detection method. This method classifies normal activities and possible threats, and can effectively detect different forms of network intrusion on computer networks. In [11], the authors propose a deep learning approach using recurrent neural networks (rnn) for intrusion detection. In terms of binary classification and multi-class classification, the RNN-IDS proposed in this paper can effectively identify intrusion types, and the detection accuracy and detection rate are higher than the traditional ML-based intrusion types. However, it performs poorly against a few classes in a multi-class classification, a common problem in cybersecurity datasets. Sarhan et al. [12] comprehensively analyzed the importance and prediction ability of feature sets in network attack detection. The chi-square algorithm, information gain algorithm, and related algorithms are used to identify and sort the data features. In the same way, the unbalance of data will still lead to the low detection accuracy of the method on the attack data.

Although deep learning technology can effectively detect malware, malicious code, malicious behavior, etc., it also has two limitations: 1) The attack data in the training process is insufficient, far less than the normal data [13]. 2) With the development of technology, the attack means of attackers are constantly changing, and the attack media used, such as malware, malicious code, and malicious network flow, are constantly changing. To solve the above two problems, the researchers introduced Gans to generate usable attack data, enhance the training data set, and improve the performance of the detection model.

As a hot topic in the field of deep learning and artificial intelligence, the idea of "generation" and "confrontation" is the link between GAN and network security, and this feature also makes it shows great application potential and development prospects in the field of network security [14]. Kim et al. [15] addresses data imbalance by using generative adversarial network (GAN) models, an unsupervised learning approach to deep learning that generates new virtual data similar to existing data. It also proposes a model that would be classified as a random forest to identify detection performance after GAN-based solutions to data imbalances, but this approach has yet to expand the scope of the study of attack network flows. Lin et al. [16] proposed a generative adversarial network framework named IDSGAN to generate adversarial malicious traffic records, aiming to attack intrusion detection systems by spoofing and evading detection. Given that the attacker does not know the internal structure and parameters of the detection system, the adversarial attack example performs a black box attack on the detection system. However, the adversarial malicious records generated by the framework do not have a suitable way to verify them. Based on the current research work, aiming at the pre-processing of NIDS, feature extraction/selection, and data set balance, this paper uses GAN technology to generate executable malicious network flows, balance CICIDS2017 data set samples, and improve the detection accuracy of DBN model on attack samples.

This section mainly introduces the current work related to NIDS, and analyzes the current focus of work and problems to be solved. The next section will focus on the framework and flow of the proposed method.

III. The G-DBN method

This section will introduce the overall framework of the G-DBN method, as well as key techniques and steps such as data preprocessing, malicious network flow generation, and model training. By using DBN model for network intrusion detection and combining it with GAN to generate malicious network flow samples, the G-DBN method is helpful to increase the training data of network intrusion detection system, especially for some rare or new intrusion samples, to solve the problem of insufficient attack samples and low detection accuracy in current network intrusion detection, to improve the generalization ability of detection system. Figure 1 illustrates the overall architecture of our approach. The generation module uses an LSTM model to generate malicious network flow data, and the discrimination module uses an MLP model to classify and discriminate between real samples and generated samples.

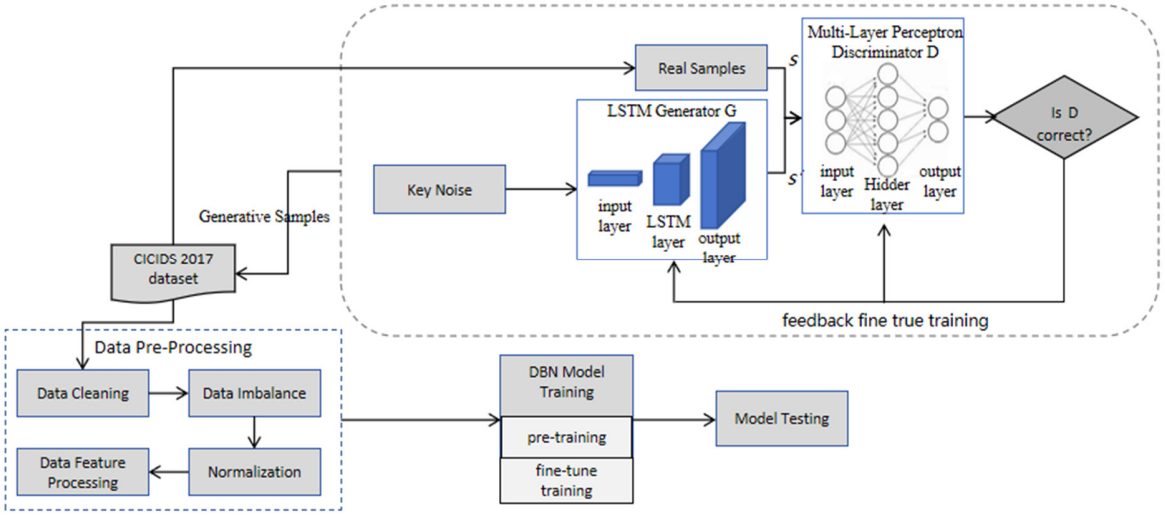

Figure 1. The Architecture of G-DBN

At the same time, due to the entire dataset being significantly imbalanced, as shown in Table 1, the normal data is very large, while the attack traffic is quite small. Therefore, given this situation, this paper first employs the malicious network flow generation method detailed in the following section A to generate executable attack samples. Subsequently, the generated fused dataset undergoes appropriate class balancing. That is, by adding weights in the algorithm's loss function, the model can account for underrepresented categories without resampling the training set. With the effects of malicious network flow generation and class balancing, the number of attack samples increases, and the penalty for misclassifying minority classes is greater than that for misclassifying majority classes. The dataset can achieve a good balance among all types of samples during training. In the following sections, we will introduce and analyze the key processes in the architecture.

Table 1 Class distribution of CICIDS2017

Lable	Counts	Lable	Counts	Lable	Counts
BENIGN	2273097	DoS Slowhttptest	5499	PortScan	158930
Bot	1966	DoS slowloris	5796	SSH-Patator	5897
DDoS	128027	FTP-Patator	7938	Web Attack Brute Force	1507
DoS GoldenEye	10293	Heartbleed	11	Web Attack Sql Injection	21
DoS Hulk	231073	Infiltration	36	Web Attack XSS	603

A. Malicious network flows are generated

As shown in Figure 1, the generative adversarial module consists of two models, the discriminant model and the generative model. The generation model captures the distribution of real samples and generates new fake samples according to the distribution; A discriminator is a binary classifier that determines whether the input is a real sample or a fake sample. Through continuous adversarial training, models G and D make D correctly identify the source of training samples, and make the fake samples generated by G more similar to the real samples. The real samples S of the generated module input are all from the CICIDS2017 data set, belonging to a small number of attack samples in the data set. GAN is a game problem between the generating network and the discriminant network. The discriminant network *D* hopes that the larger the probability value of the real sample x, the smaller the probability value of judging the fake sample *G (z)* as the real sample, and the generating network G hopes that the more similar the fake sample *G (z)* and *S*, the better. The higher the probability *D (G (z))* that the discriminant network will judge it to be a real sample, the better. The training optimization objective of the GAN network is as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[log\ D(x)] + E_{z\sim p_z(z)}[log\ (1-D(G(z)))] \quad (1)$$

where $V(D,G)$ represents the difference between the generated sample and the real sample, and the cross entropy loss of the binary classification (true and false categories) can be used. $\max_{D} V(D,G)$ represents updating the parameters of the

discriminator D by maximizing the cross-entropy loss $V(D,G)$ with the generator fixed.

$\min_G \max_D V(D,G)$ means that the generator should minimize the cross entropy loss $V(D,G)$ of true and false images if the discriminator maximizes the cross entropy loss $V(D,G)$. The right side of the equation expands the cross entropy loss formula on the left side of the equation and writes it in the desired form of the probability distribution. At the same time, to ensure that the generated sample has the same executability and aggressiveness as the real sample *S*, this paper retains all positions except the key bit *key* of the generated sample, that is, first obtain the key bit information of the real sample, empty the key bit information of the generated sample, and then compare the obtained sample with the processed generated sample *S'*. The processing formula is as follows: In the formula, *key** is the inverse result of *key*.

$$S' = S\&key\&(G(z)\&key^*) \tag{2}$$

B. G-DBN model training

The components of the DBN are Restricted Boltzmann Machines (RBM). RBM stands for Restricted Boltzmann Machine, which consists of only two layers of neurons. One layer is called the visible layer, composed of visible units, which are used to input training data. The other layer is called the hidden layer, composed of hidden units, which serve as feature detectors. The training process of Deep Belief Networks (DBN) is conducted layer by layer. In each layer, the hidden layer is inferred using the data vector, and then this hidden layer is treated as the data vector for the next layer (higher layer).

Figure 2 shows the architecture for implementing G-DBN. After fine-tuning, the three RBMS are stacked with the (49,128), (128,64), (64,9) visible/hidden nodes set for each RBM. The output of the last RBM is connected to a fully connected layer with 6 nodes, using the Softmax function for multi-class classification.

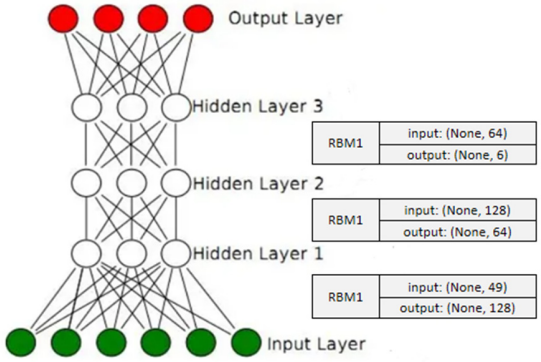

Figure 2. the architecture for implementing G-DBN

IV. Experiment and Analysis

A. Experimental environment and experimental data

To test the performance of G-DBN, this paper used the CICIDS2017 dataset created by the Canadian Cyber Security Institute (CIC), which contains benign and up-to-date common attacks, similar to Real World data (PCAP). It also includes the results of network traffic analysis using CICFlowMeter, which includes the flow of tags based on time stamps, source and destination IP, source and destination ports, protocols, and attacks (CSV files). The dataset identifies the 11 criteria needed to build a reliable baseline dataset, including full network configuration, full traffic, labeled data set, full interaction, full capture, available protocols, attack diversity, heterogeneity, feature set, metadata, and complete research papers that enable a complete test of G-DBN's network intrusion detection performance. All experiments were conducted in a Windows 11 environment using a 64-bit Intel(R) Core(TM) i5-12450U CPU and 16GB RAM. These models have been implemented in Python v3.7.0 using the PyTorch v1.13.1 library.

B. Experimental Results and Analysis

Figures 3 and 4 show the confusion matrix of G-DBN and normal 3-layer DBN on the test set. The number of incorrect and correct categories is summarized as the count for each label. As you can see, both models can correctly classify most network traffic samples. However, the two models differ significantly in terms of accuracy. The Standard 3-layer DBN recognized 13.57% of "Web attack" mistakes as other sorts of traffic and had an identification accuracy of 86.19% for offensive traffic, whereas the G-DBN enhanced that accuracy to 92.65%. This implies that the G-DBN model's capacity to detect attack traffic has improved as a result of the increased number of attack samples and improved data set balance. Significantly improved when compared to before the optimization.

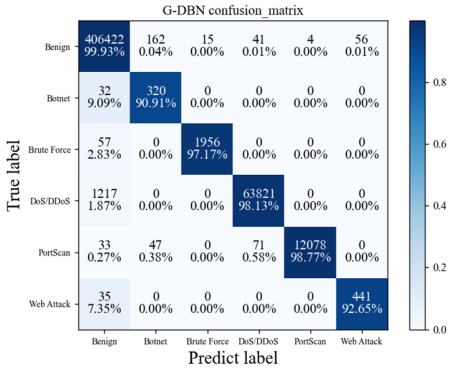

Figure 3. G-DBN Confusion Matrix

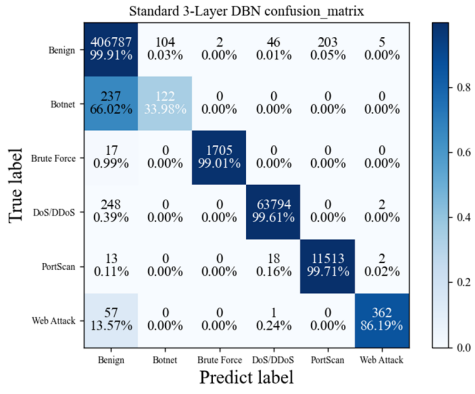

Figure 4. Standard 3-Layer DBN Confusion Matrix

Figure 5 shows the difference between the average ROC curve of G-DBN and ordinary 3-layer DBN. The AUC value of G-DBN is 0.18% higher than that of ordinary DBN, which proves that the G-DBN model has a higher true positive rate and a lower false positive rate, that is, the model has a better classification ability. Data samples that can more accurately distinguish between network intrusion and non-network intrusion.

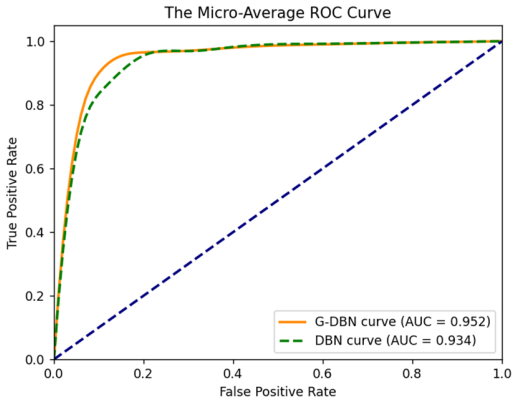

Figure 5. The ROC Curve of G-DBN and Standard 3-Layer DBN

This paper also compares the G-DBN model with other network intrusion detection methods using CICIDS2017 dataset. All of the methods studied used supervised learning methods to train their models. However, some studies use binary classification instead of multi-class classification. LSTM [17] and 1D-CNN [17] are deep learning models that perform binary classification. LUCID [18] used cnn to detect DDoS attacks. Table 2 summarizes the performance results of the above models and compares them with the model presented in this paper.

Table 2 Performance comparison against existing methods

Method	F1-score	Recall	Precision	Number of Classes
G-DBN	0.933	0.963	0.913	6
standard three-layer DBN	0.886	0.864	0.916	6
LSTM	0.895	0.898	0.984	2
1D-CNN	0.939	0.901	0.981	2
LUCID	0.996	0.999	0.993	2

V. Conclusion

Aiming at the problem of unbalanced data and low detection accuracy in traditional NIDS, this paper integrates the generation of adversus-network technology, introduces the method of making adversus-sample directly by using malicious network flow as the original sample, and proposes the G-DBN multi-classification network intrusion detection method. At the same time, the experimental results of this paper show that the detection accuracy of G-DBN is better than that of the same three-layer DBN detection model, and the accuracy is increased by 6%. At the same time, in other current NIDS research, G-DBN method also has many improvements compared with binary classification method, especially when the data set contains only a small number of attack samples, it can further meet the practical application requirements of data centers. In the subsequent research, it is necessary to increase the correctness of the generated attack samples and further improve the correctness of the generated samples for the verification link in the generated adversarial network.

Acknowledgment

This work was financially supported by “Innovative research and application project of intelligent operation and maintenance system of the computer room” of China Tobacco Guangxi Industrial Co., Ltd (2021450000340073).

References

[1] Zhang G, Lin J, Zhang Y, et al. Big data based intelligent operation and maintenance platform[C]//2020 IEEE 5th international conference on intelligent transportation engineering (ICITE). IEEE, 2020: 249-253.

[2] Zhao Y, Wang N, Liu Z, et al. Construction theory for a building intelligent operation and maintenance system based on digital twins and machine learning[J]. Buildings, 2022, 12(2): 87.

[3] Belarbi O, Khan A, Carnelli P, et al. An intrusion detection system based on deep belief networks[C]//International Conference on Science of Cyber Security. Cham: Springer International Publishing, 2022: 377-392.

[4] Nguyen H D, Tran K P, Thomassey S, et al. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management[J]. International Journal of Information Management, 2021, 57: 102282.

[5] Zhou X, Hu Y, Liang W, et al. Variational LSTM enhanced anomaly detection for industrial big data[J]. IEEE Transactions on Industrial Informatics, 2020, 17(5): 3469-3477.

[6] Xiang L, Wang P, Yang X, et al. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism[J]. Measurement, 2021, 175: 109094.

[7] Wang W, Sheng Y, Wang J, et al. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection[J]. IEEE access, 2017, 6: 1792-1806.

[8] Staudemeyer R C, Morris E R. Understanding LSTM--a tutorial into long short-term memory recurrent neural networks[J]. arXiv preprint arXiv:1909.09586, 2019.

[9] Smagulova K, James A P. A survey on LSTM memristive neural network architectures and applications[J]. The European Physical Journal Special Topics, 2019, 228(10): 2313-2324.

[10] Desuky A S, Hussain S, Kausar S, et al. EAOA: an enhanced archimedes optimization algorithm for feature selection in classification[J]. IEEE Access, 2021, 9: 120795-120814.

[11] Gao H, Zhang Z, Wang S, et al. Underdetermined blind source separation method based on quantum Archimedes optimization algorithm [J]. Applied Intelligence, 2022: 1-38.

[12] Ding Guiyan, Wang Wentao, Liu Hao, Tu Liangping. Defect of Archimedes optimization algorithm and its verification[J]. Soft Computing, 2022,27(2).

[13] Li Song, Wang Jie Sheng, Song Hao Ming, Zheng Yue, Zhang Xing Yue. Buoyancy energy driven archimedes optimization algorithm based on Lévy flight and tangent flight[J]. Journal of Intelligent & Fuzzy Systems,2022,43(6).

[14] Varol Altay E. Hybrid Archimedes optimization algorithm enhanced with mutualism scheme for global optimization problems[J]. Artificial Intelligence Review, 2022: 1-62.

[15] Kim H Y, Won C H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models[J]. Expert Systems with Applications, 2018, 103: 25-37.

[16] Lin Z, Shi Y, Xue Z. Idsgan: Generative adversarial networks for attack generation against intrusion detection[C]//Pacific-asia conference on

knowledge discovery and data mining. Cham: Springer International Publishing, 2022: 79-91.

[17] Roopak M, Tian G Y, Chambers J. Deep learning models for cyber security in IoT networks[C]//2019 IEEE 9th annual computing and communication workshop and conference (CCWC). IEEE, 2019: 0452-0457.

[18] Doriguzzi-Corin R, Millar S, Scott-Hayward S, et al. LUCID: A practical, lightweight deep learning solution for DDoS attack detection[J]. IEEE Transactions on Network and Service Management, 2020, 17(2): 876-889.