2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing (ICSP 2021)

Transfer-learning-based Network Traffic Automatic Generation Framework

Yanjie Li* , Tianrui Liu
Department of Electrical Engineering
Zhejiang University
Hangzhou, China
{yanjieli, terryliu18}@zju.edu.cn

Dongxiao Jiang, Tao Meng
Electric Power Research Institute
State Grid Jilin Company
Changchun, China
{913468136, 494344848}@qq.com

2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP) | 978-1-6654-0413-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICSP51882.2021.9408767

***Abstract*—Nowadays, there is an increasing number of attacks against the network system. The intrusion detection system is a standard method to prevent attack. In essence intrusion detection is a classification problem to judge normal or abnormal behaviors according to network traffic characteristics, and deep learning has been applied to intrusion detection recently. However, due to the lack of training data in some systems such as industrial control systems and smart grid, the deep-learning algorithm cannot give full play to its advantages. To solve this problem, we propose a transfer-learning-based network flow generation framework for deep-learning-based intrusion detection, which uses invariant extraction and sequence to sequence generation, to extract the attack invariant of the existing attack data set and transfer the knowledge to the target network system. We use the open-source data set on real systems and carry out relevant experiments, proving that our method can generate effective anomaly traffic as well as improve the accuracy of intrusion detection.**

Keywords- Network Network Automatic Generation; Network Intrusion detection; Transfer learning;

I. INTRODUCTION

With the development of network technology, an increasing number of attacks have appeared in the network system. Many of these attacks are carried out through communication protocols[1]. The deployment of an intrusion detection system is a standard method to prevent or detect this kind of attack. In the face of complex network behavior and massive data, the traditional network threat detection method based on rules is prone to many problems such as high false positives and long delays. As intrusion detection is a classification problem to judge normal or abnormal behaviors according to network traffic characteristics, deep learning technology has been applied to intrusion detection. However, due to the lack of training data in some systems such as industrial control systems, the deep learning algorithm cannot give full play to its advantages. Therefore, it is urgent and necessary to study the attack data generation and augment technology for the industrial network and tackle the dataset problem at its source to improve the system's security further.

To solve this problem, we propose a transfer-learning-based attack traffic generation framework. The framework transfers attack invariant between networks of different topologies and protocol families and generate new anomaly traffic for the target industrial control system. As far as we have known, it has not been done in previous studies. It should be mentioned that, as our following experiments show, our method can apply not only to traditional protocols but also to industrial control protocols, such as Modbus/TCP protocol traffic with different network topologies. The potential use of our method to generate fuzzy test cases is also presented.

II. RELATED WORK

A. Network Intrusion Detection System

There are four main steps in an anomaly-based network intrusion detection: data collection, feature extraction, model training, and anomaly detection. For example, Mirsky proposed Kitsune, which can learn to detect attacks on local networks without monitoring and adopt an efficient online method. [1].

However, these methods are currently aimed at the TCP/IP protocol family in the traditional network environment. In the traditional network environment, researchers have constructed a large number of anomaly detection data sets, such as KDDCup99[2], NSL-KDD[3]. Therefore, it is feasible to deploy deep learning anomaly detection methods. However, in some cases, such as industrial control systems or some less-used protocols, these methods cannot be directly applied due to the lack of reasonable and effective attack data. Researchers had proposed a few methods to solve this problem, mainly to generate new data through adversarial generative learning.

Ring et al. proposed IP2VEC[4] to convert the different fields in the protocol into vectors, which idea is borrowed from word2vec, and then use GAN to generate new data set, and conducted experiments on the CIDDS1 dataset[5]. However, their method is limited to generate data using the target network traffic and does not use cross-domain knowledge, so they can only generate attack data seen in the target network before. Moreover, the GAN-based method uses stochastic noise to mutate the traffic, which lacks clear goals. The attack knowledge in the industrial control field is relatively limited compared to traditional networks. Suppose we can use the knowledge of traditional networks. In that case, it will be possible to increase the quality of our generated data sets and improve NIDS's detection accuracy further.

B. Transfer learning

Traditional machine learning assumes that the training set and the test set have the same distribution, but in practical problems, this requirement is often difficult to satisfy because the application scenario is often different from the original training scenario.Transfer learning provides a way to solve this problem.Transfer learning has been widely used in small-sample learning. Depending on the different transfer learning objects, it can be divided into transferring knowledge of feature

978-1-6654-0413-6/21/$31.00 ©2021 IEEE

representations and transferring knowledge of parameters. The former focuses on the transfer of knowledge of the data itself, and the latter focuses on the transfer of knowledge of the model. The generative framework proposed in this work transfers the knowledge of the data itself, which can not only transfer the knowledge of attack invariant, but also can generate new traffic.

Feature knowledge transfer. The method of feature knowledge transfer can be divided into the supervised and unsupervised method. Due to the lack of paired data in network traffic, an unsupervised approach is required. At present, feature knowledge transfer has been widely studied in text style transfer[7] and image style transfer, but there are little researches in the field of network traffic generation[8]. Using the idea of the invariant representations of attack characteristics, we propose a network traffic generating framework that extracts the attack invariants of source network's traffic with different protocols and transfers the attack invariants to the target network, and generate new anomaly traffic of target network.

III. FRAMEWORK

The transfer-learning-based attack traffic generation framework composes two parts. The first part is a vector embedding model based on IP2VEC. The second part of our framework is a seq2seq model, which aims to realize end-to-end traffic generation. Due to the lack of paired data between the target network and the original network, we designed the seq2seq model with multiple decoders for unsupervised training. At the same time, in order to extract the optimal representation of attack invariants, we add the part of adversarial training in the network.

A. IP2VEC

IP2VEC is the first part of our model. IP2VEC is a neural network with one hidden layer. The generating method of training data used by IP2VEC is shown in Fig.1. Its main function is to embed the string-type fields like IP addresses into vectors and extract semantics so that "semantically" similar fields will have similar vector representations. For example, the distance between two embedding vectors from two devices with similar functions should be less than the distance from two embedding vectors from the normal device and the attacker's computer, even if the normal device and the attacker are in the same network segment. Because the IP2VEC algorithm calculates the joint probability density of the different fields in the same traffic, and many package fields from similar devices often have the same content in addition to some special fields, it can map the IP address of different devices with similar function to closer space, while mapping the attacker's IP field to a farther space.

We should mention that in the original algorithm, all fields are embedded using the neural network. However, in practice, we found that some fields are continuous quantities, which do not need to be embedded, such as the packet size. Moreover, if they are embedded, the dictionary will be extremely large because these fields have infinite value space. For these two reasons, we improved the embedding algorithm by modifying the embedding matrix. The modified embedding matrix is as:

$$\Phi_i \in \begin{cases} \mathrm{R}^{n \times k} & \text{the } i\text{-th field is categorical field} \\ \mathrm{R}^{k} & \text{the i-th field is numerical field} \end{cases} \quad (1)$$

Where k is the dimension of the vectors, n is the value space of a categorical field. Then the embedding vector of x_i^t is calculated by:

$$v_i^t = \begin{cases} \Phi_i[x_i^t] & \text{the } i\text{-th field is categorical field} \\ x_i^t \times \Phi_i[x_i^t] & \text{the } i\text{-th field is numerical field} \end{cases} \quad (2)$$

Where $\Phi_i[x_i^t]$ denotes the x_i^t-th row of Φ_i.

B. Multi-decoder seq2seq model

The original sequence to sequence model has been widely used in style transfer tasks, but need a large number of parallel data during the training stage. For example, in the text translation, it needs to train pairs of sentences in two languages with the same meaning. However, in the scenario of our problem, although there are some similar attack types in the traditional network and industrial control networks, such as DOS attack and probe attack, it requires much workforce to build these data sets manually. Besides, target networks, such as industrial control networks, may lack attack samples, so paired data sets cannot be constructed. For these reasons, the original seq2seq model cannot be used directly to generate attack traffic. We think this problem is similar to the unsupervised style transfer problem in traditional machine learning. To solve the unsupervised style transfer problem in traditional machine learning, Fu et al. proposed a multi-decoder structure seq2seq model, which can realize the transfer from the paper-style to the news-style title without paired supervised learning data, and obtained good results. We used a similar structure in the second part of the model. We use the auto-encoder idea to train our decoder to make it recover the X from D^{tgt} or D^{src} using intermedia representations.

At the same time, in order to extract attack invariants, the adversarial training process is added to the multi-decoder seq2seq model. The adversarial training process composes two parts. The first part wants to classify the network "style" of the optimal representations learned by the encoder. That is, we want the classifier to recognize whether the intermediate representation is learned from the source network or the target network. The second part is the encoder, making the classifier unable to identify the network "style" by maximizing the classifier's prediction entropy. As a result, only optimal representations of attack invariants have remained from embedded vectors of traffic, and redundant features like traffic formats are removed.

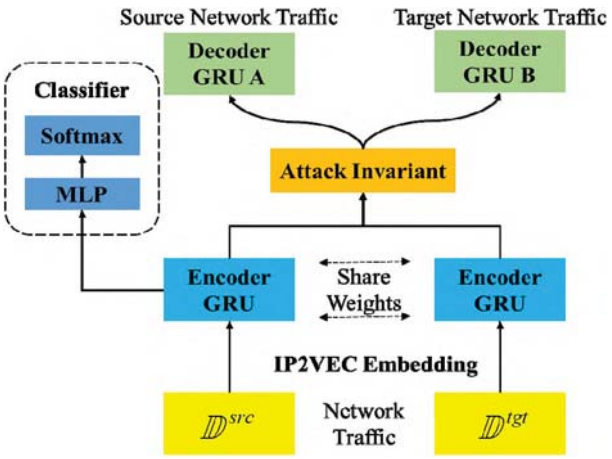

Figure 1. The transfer-learning-based attack traffic generation framework in this paper. The IP2VEC embeds the source and target network into vectors. The encoder encodes the vector and separates the network "style" feature and attack invariant through adversarial training. Then two different decoders trained by auto-encoder decodes the attack invariant into new abnormal network traffic.

The loss function of the classifier is:

$$L_{adv1}(\Theta_c) = -\sum \log p(l_i \mid Encoder(x_i;\Theta_e);\Theta_c), \quad (3)$$

The loss function of the encoder is:

$$L_{adv2}(\Theta_e) = -\sum_{i=1}^{M}\sum_{j=1}^{N} H(p(j \mid Encoder(x_i;\Theta_e);\Theta_c)), \quad (4)$$

The loss function of the decoders is:

$$L_{decoder}(\Theta_e,\Theta_d) = \sum_{i=1}^{L} L^i_{seq2seq}(\Theta_e,\Theta^i_d), \quad (5)$$

Finally, the total loss function of the model is:

$$L_{total}(\Theta_e,\Theta_d,\Theta_c) = L_{adv1}(\Theta_c) + L_{adv2}(\Theta_e) + L_{decoder}(\Theta_e,\Theta_d), \quad (6)$$

IV. EXPERIMENTS

In order to verify our proposed method, we conducted our experiments by transferring attack features between two datasets with the same protocol but different structure, and the second transfer attack features between datasets of two different protocols. we used CIDDS01 and CIDDS02 datasets. Both of them are TCP/IP protocols, but the network structure, IP addresses, and types of attacks they contain are different. In the future, we plan to transfer attack invariants between different protocols. Finally, several classifiers like KNN classifier and neural network classifier are trained respectively to verify that the auxiliary data set generated by our method can be used to improve the effect of intrusion detection.

A. Data preprocessing

We first preprocess the collected traffic data set to clean the data. It mainly includes eliminating missing values and outliers and normalization the numerical fields.

In addition, if the newly generated traffic does not conform to the traffic format, it should also be deleted. In the actual experiment, we found that less than 1% of the newly generated packets did not conform to the traffic format.

B. Transfer-learning-based traffic generation

We conducted experiments on CIDDS01 and CIDDS02 datasets. Both of these datasets are TCP/IP protocols, but the network structure, IP addresses, and types of attacks they contain are different.

TABLE 1. The orginal traffic from CIDDS02 dataset and generated traffic in the format of CIDDS01 datataset.

Network label	*srcIP, srcPt,dstIP,dstPt,proto,packets,bytes,duration*	*Label*
original X from CIDDS02	**192.168.220.51**,0_p,192.168.100.3,8_p,ICMP,1_k,42_b,0_d	attacker
generated Y for CIDDS01	**192.168.220.9**,48888_p,192.168.100.5,445_p,TCP,3_k,229_b,0.066_d	
original X from CIDDS02	192.168.220.51,138_p,**192.168.220.255**,138_p,UDP,1_k,262_b,0_d	attacker
generated Y for CIDDS01	192.168.200.5,53435_p,**192.168.100.5**,445_p,TCP,3_k,229_b,0.034_d	
original X from CIDDS02	**192.168.220.1**,0_p,192.168.220.51,3.1_p,ICMP,1_k,70_b,0_d	victim
generated Y for CIDDS01	**EXT_SERVER**,8000_p,192.168.200.9,60803_p,TCP,2_k,154_b,0.020_d	
original X from CIDDS02	192.168.220.1,0_p,**192.168.220.51**,3.1_p,ICMP,1_k,70_b,0_d	victim
generated Y for CIDDS01	EXT_SERVER,8000_p,**192.168.200.9**,60803_p,TCP,2_k,154_b,0.020_d	

We use CIDDS02 as source dataset and use CIDDS01 as target dataset. The generated traffic is compared with the original traffic in Table I. We can see that the IP address and other fields have been automatically transferred and the attack features have remained.

C. Evaluation

In order to compare the effect of the generated data sets, we use the unsupervised clustering algorithm kNN to evaluate the generated data sets. The result is shown in Fig 2. We use CIDDS02 as source dataset and use CIDDS01 as target dataset. The result shows that the generated data can be separated clearly.

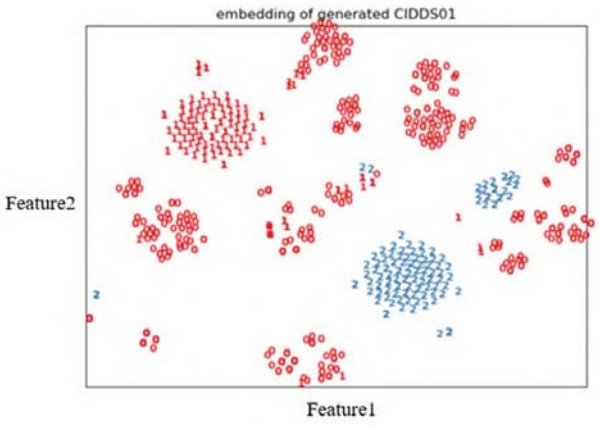

Figure 2. The embedding of generated CIDDS01 Dataset traffic. We use CIDDS02 as source dataset and use CIDDS01 as target dataset. The result shows that the generated data can be separated clearly. The label 0 is generated normal traffic. The label 1 is generated attaker's traffic, and the label 2 is generated victim's traffic.

V. Conclusion And Future Work

To verify our proposed method, we have conducted our experiments by transferring attack features between two datasets with the same protocol but different structures. Both of them are TCP/IP protocols, but the network structure, IP addresses, and types of attacks they contain are different.

As future work, we plan to transfer attack features between datasets of two different protocols, for example, TCP/IP and MODBUS/TCP. Furthermore, one can also use the framework to generate fuzz heuristics and fuzz the protocols for analyzing the robustness of server or grid terminal equipment.

Acknowledgement

This work was supported by the technology research project of State Grid Corporation of China" Research on Security Attack Evaluation and Monitoring Technology of Electric IoT Devices Based on Artificial Intelligence "(52230019000B).

References

[1] Mirsky Y, Doitshman T, Elovici Y, Shabtai A. Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089. 2018 Feb 25. (Unpublished)

[2] KDD Cup 99 Dataset. Accessed: Oct. 31, 2019. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3] Nsl-KDD Dataset. Accessed: Oct. 31, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[4] M. Ring, A. Dallmann, D. Landes and A. Hotho, "IP2Vec: Learning Similarities Between IP Addresses", 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, 2017, pp. 657-666, doi: 10.1109/ICDMW.2017.93.

[5] Ring M, Schlör D, Landes D, Hotho A. Flow-based network traffic generation using generative adversarial networks. Computers & Security. 2019 May 1;82:156-72.

[6] Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, 3104–3112.

[7] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan. Style transfer in text: Exploration and evaluation. In Association for the Advancement of Artificial Intelligence (AAAI), 2018.

[8] A. G. Voyiatzis, K. Katsigiannis and S. Koubias, "A Modbus/TCP Fuzzer for testing internetworked industrial systems," 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 2015, pp. 1-6, doi: 10.1109/ETFA.2015.7301400.