INVESTIGATION OF POISONOUS ATTACKS AGAINST NETWORK INTRUSION DETECTION SYSTEMS

Van Quan Nguyen[1], *Van Cuong Nguyen*[1], *Tuan Hao Hoang*[1]

https://doi.org/10.56651/lqdtu.jst.v11.n01.359.ict

Abstract

Nowadays, deep learning is becoming the most strong and efficient framework, which can be implemented in a wide range of areas. Particularly, advances of modern deep learning approaches have proven their effectiveness in building next generation smart intrusion detection systems (IDSs). However, deep learning-based systems are still vulnerable to adversarial examples, which can destroy the robustness of the models. Poisoning attack is a family of adversarial attacks against machine learning-based models. Generally, an adversary has the ability to inject a small proportion of malicious samples into training dataset to degrade the performance of victim's models. The robustness of deep learning-based IDSs has been becoming a really important concern. In this work, we investigate poisonous attacks against deep learning-based network intrusion detection systems. We clarify the general attack strategy, perform experiments on multiple datasets including CTU13-08, CTU13-09, CTU13-10 and CTU13-13. Experimental results have shown that only a small amount of injected samples has drastically reduced the performance of the deep learning-based IDSs.

Index terms

Adversarial Attack, Robustness of Deep Learning, Network Intrusion Detection System.

1. Introduction

The dramatic increase of computer networks in terms of size, type, services and applications has made them more and more complicated and heterogeneous. They have been becoming the victims of huge number of critical threats including malicious activities, cyber criminals and also non-malicious behaviours. Detecting and preventing these harmful activities are important tasks in cyber security domain. However, monitoring network traffic and analyzing them to identify malicious threats are challenging tasks, especially in the context of large-scale networks [1]. In addition, the scope of attackers and malware programs have been changing significantly day by day [2]. Recently, many of artificial intelligence algorithms have long been applied for purpose of improving the performance of network IDSs [3], in which illegal intrusion and attacking data are can be called network anomalies. It is very crucial for automatically determining illegal activities and also variety forms of network attacks among network traffics. Many

[1]Faculty of Information Technology, Le Quy Don Technical University

machine learning-based methods have received a great success for building network anomaly detection systems.

Machine learning models are generally divided into three categories including supervised learning, unsupervised learning and semi-supervised learning depend on the availability of labeled data [4]. Supervised learning models are trained using labeled datasets, in which labelled data may be classified either normal network traffics or anomalies. Unsupervised learning models are trained using unlabeled data. Semi-supervised learning is a combination of supervised and unsupervised learning, in which some observations are labelled and the others are not.

Recently, deep learning algorithms have demonstrated huge success for IDSs in comparion with traditional machine learning methods [5]. In particular, this success becomes more and more evident as the amount of network data grows at an exponential frequency every day, However, in many cases, data from normal behaviors of network systems tend to be available and easy to collect. By contrast, outliers are scarce and expensive to capture [6]. Therefore, the semi-supervised learning algorithms are suitable approarch to develop network IDSs from only normal data.

Autoencoder (AE) has been using as the state-of-the-art method for network IDSs [6], [7], [8], [9]. In semi-supervised manner, AE-based models tend to capture the core characteristics of normal observations in order to find out the distinction between them and anomaly samples. However, it is very critical to improve the safety and robustness of the deployed AE-based models.

Adversarial samples are always huge concern when applying deep learning models in practical applications such as autonomous driving, fraud detection, face recognition...etc. Adversarial samples are usually impossible to detect by human eyes. However, they can lead the model to misclassify the output and pose critical threats to system's performance [10]. The safety and robustness of any deep learning-based models are determined by considering the adversarial purposes and their abilities. There are two main types of adversarial attacks against machine learning models [11]:

- Inference-time or evasion attacks: This kind of attack forces the trained model to misclassify carefully perturbed samples [12], [13], [14].
- Training-time or poisoning attacks: This kind of attack is exploited in the training time. In particular, the model is trained with malicious, crafted inputs to eventually compromise the whole training process [15], [16].

It is true that the adversarial attacks in the training time are difficult to accomplish. However, they are considered as very potential and powerful threats against machine learning-based models. With the goals to gain higher accuracy and also better performance, deep learning models require numerous training datasets from different sources. In many scenarios, the models are trained using outsourced malicious dataset produced by third party, in which adversarial samples are inserted into training datasets.

In this paper we investigate the poisoning attack against network IDSs, which are

trained in semi-supervised manner. Firstly, we train our model using only clean normal training data. Afterward we inject small portion of abnormal observations into clean dataset and train model in the same way. We compare behaviour of resulted models with similar testing datasets. The results have shown that only small percentage of injected malicious samples can cause serious problems for IDSs' performance. In particular, IDSs are built in a semi-supervised manner using only regular network data, the rate of accurate detection of anomalous data points decreases. In other words, the IDSs will classify outliers as normal with a higher rate. Attackers will exploit this vulnerability to deploy Advanced Persistent Threat (APT) attacks capable to bypass deep learning-based IDSs. To this end, they try to inject a small amount of malicious data into the training set of the models in a semi-supervised manner. In order to propose a solution to prevent poisoning attacks to training set of IDSs, in this paper, we conduct experiments to determine and analyze the effect of injected malicious data in the training set on the performance of the deep learning-based IDSs.

This paper is organized as follows: In section 2, background of adversarial threat model attacks and particular poisonous attacks against machine learning models are introduced. In section 3, we briefly review some recent works related with using poisoning attacks against artificial intelligence systems. In section 4, we explain our attack model against network IDSs. Experiments, results and discussion are presented in sections 5 and 6, respectively. Finally, we shall conclude our paper with highlights and future directions.

2. Background

2.1. Adversarial threat model

In this section, we aim to define threat surface [17] of machine learning-based systems to clarify where and how an attacker might tend to subvert the system under attack setting. A machine learning-based system is generally considered as data processing pipeline. The order of operations at the testing phase can be divided into four separated stages as follows: (1). Collection input data; (2). Transformation the input data into the appropriate values; (3). Calculation the produced output from the system; (4). Taking action based on the resulted output. Therefore, the attack surface can be defined with respect to the data processing pipeline. An attacker might have opportunity to harmfully manipulate data at collection or the calculation output time to degrade the victim's model. The main types of attack surface are introduced as follow:

- Evasion Attack: There is the most popular type of attacks in the adversarial setting. Generally, the system is evaded by using crafted malicious samples during testing time. The attacker's purpose is to create carefully crafted samples, which are misclassified by system with high confidence.
- Poisoning Attack: This kind of attack happens during the training time. Under attack setting, an adversary tries to inject carefully designed samples into training dataset to compromise the learning process.

- Exploratory Attack: This kind of attacks does not tend to access the training data. However, an adversary tries to gain knowledge about the learning algorithm.

The amount of information about system available to an adversary is the most important factor to define threat model. Next, we will discuss plenty of adversarial capabilities against machine learning-based systems during training stage. The goal of an attacker during training time is to compromise the model by altering the training dataset in some ways. Generally, there are three strategies for attacking training dataset as follows:

- Data Injection: In this case, the attacker has no access to the training sets as well as learning algorithm but has capability to insert new samples into training sets. By this way, he can inject some adversarial examples into training sets to target victim's model.
- Data Modification: In this case, the attacker has full access to the training data, but knows nothing about the learning algorithm. He has the ability to modify the whole data beforehand it is used for training process.
- Logic Corruption: In this case, the attacker has the ability to interfere learning algorithm. The goal of an adversary is to poison and change the way the model captering characteristics from data.

On the other hand, adversarial attacks at the testing time do not target training dataset but force the learned model to produce incorrect outputs. The amount of information available to the attacker is the core factor to determine the effectiveness of this attack methodology [10]. Generally, adversarial attacks at testing phase are classified into two categories, including White-Box and Black-Box attacks.

- White-Box: It is when an adversary has comprehensive knowledge about the model. This means ability to access to the structure, parameters, and complete training process of machine learning models. Information about training dataset, hyperparameters, weights, activation functions and the number of layers, the number of neurons at each layer are available to the attackers. The adversary exploit such information to scan and identify vulnerabilities of the models to launch the attack with the highest efficiency by their way.
- Black-Box: It is possibble when attackers have no knowledge and access to the model including structure, parameters and the training process. For instance, the adversary can target a model by giving a series of very carefully designed inputs and monitoring outputs. Then the attacker tries to create a fake model that closely resembles the victim's model based on produced outputs.

2.2. Poisoning attack model

Poisoning attack is considered one of the most popular threats among variety of different attacks against machine learning models [18]. Collecting training set is an essential process in machine learning project pipeline. However, the security of this process is often underdetermined that gives adversaries a chance to pollute the training data beforehand fitting to the model. The general architecture of poisoning attack is

shown in the Fig. 1. The original training datasets of the machine learning models in almost cases are confidential. Therefore it is generally impossible for an adversary has a chance to access and modify it. However, with the rapid growth of deep learning models in terms of sizes and architectures, many deep learning systems need more and more additional training data in order to improve their performance. In practice, this data may be collected from the Internet; hiring third parties for labeling or stored in the cloud, it creates a chance for attackers to manipulate the datasets by their way. An adversary may design carefully sophisticated malicious data either with wrong lables or adding noise into clean datasets. The goal of this type of attacks is to change or event destroy the probability distribution of the clean training dataset in order to decrease the accuracy or precision of the learned models. Such attacks have implemented in a plenty of applications, including malware detection, spam filter, handwritten digit recognition [19], [20], [21]. Generally, poisoning attacks are classified into two groups as poisoning labeled datasets and poisoning unsupervised clustering [22].

- Poisoning labled datasets: In this attack, an adversary may modify existing examples or create new damaging sample and add to the training datasets in some way. By using such methods, attackers tend to cause harmful effects on the trained models. In this category, there are two main techniques: Indiscriminate and targeted poisoning. The most common form of attacks that machine learning models are likely suffer is an indiscriminate attack, in which the attacker tries to poison the models to reduce its accuracy. For instance, the authors in [23] have modified 1% of the training data for the goal reducing the accuracy of spam classifier. On the other hand, targeted poisoning attacks tend to produce specific mis-classification of particular samples. For example, the researchers at [24] have modified 1% of training dataset in order to misclassify CIFAR-10 dataset.
- Poisoning unsupervised clustering: In unsupervised learning manner, there are no labled data and the models try to detect similar classes in the datasets without supervision. It is shown that, there are vulnerable points for an adversary to inject harmful unlabeled data to significantly reduce the performance of the models. For instance, Biggio at [25] has investigated the scenarios when an adversary may be able to subvert the model for clustering malware behaviors by injecting carefully samples with poisonous behavior.

3. Related works

In recent years, there are many effective neural network-based methods for anomaly detection. However, along with it, there are also a variety of attack techniques against machine learning-based anomaly detection systems. According to [26], attacks against learning algorithms can be classified into two types, causation (manipulation of training data) and exploratory (classifier mining). A poisoning attack refers to a pathogenic attack in which specially crafted attack points are fed into the training data. This attack is particularly important from a practical point of view, as attackers often cannot directly

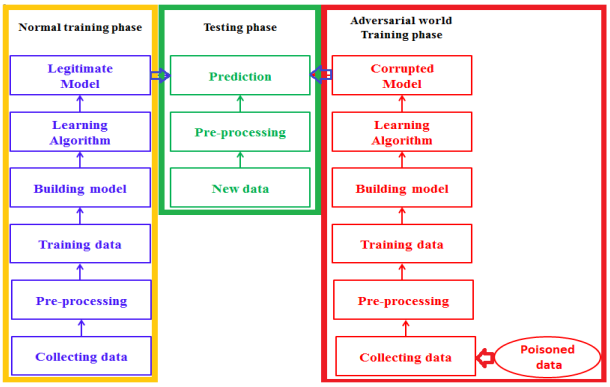

Fig. 1. Architecture of poisoning attack.

access existing training databases but can feed new training data. For example, web-based repositories and honeypots often collect examples of malware for training, which creates an opportunity for adversaries to poison the training data. Poisoning attacks have previously been studied only for simple anomaly detection methods [26], [27], [28]. However, there are many sophisticated techniques that use this type of attack to exploit deep learning models.

Gu et al. [29] showed that backdoor poisoning (BP) attacks have a great influence on the results of the model. BP attacks are actually an implicit threat to machine learning models. They still have an excellent performance during training, but when it comes to inference the results are not so good.

Kurita et al. [30] showed that language models' weights can be injected with vulnerabilities which can enable manipulation of finetuned models' predictions. They determined that, the RIPPLES method is capable of generating poison attacks with a high success rate even without access to the training set or hyperparameter settings. Another work presents different backdoor poisoning attacks against image classification models is Salem et al. [31]. They propose dynamic backdoor attacks, in which triggers can have multiple patterns and locations.

Recently, Chan et al. [32] proposed an attack method on the AE-based model in the text classification process. The paper has demonstrated that the Poison attacks have a serious effect on natural language inference and text classification systems. In this work, the authors utilized conditional adversarial regularized AE (CARA) to generate poisoned samples by poison injection in latent space. The experiments show that a victim BERT finetuned classifier's predictions can be steered to the poison target class with high success rates.

In recent years, many Autoencoder (AE)-based models have been developed to build IDSs networks, especially models are trained in one-class training manner, in which only normal network data is used. However, the safety and robustness of such models still have many concerns, especially when facing adversarial examples. Specifically,

the poisoning attack that injects malicious samples into training dataset for the during training process of IDSs in semi-supervised manner has received widespread attention in the information security community. In addition, with the increasing proliferation of APT attacks, when the attack data to the victim's system is very sparse, the poisoning attack becomes more and more dangerous. To clarify the influence of injected malicious samples into training set for AE-based network IDSs, in this work, we firstly develop an AE-based model to capture the normal behavior of network data. These AEs are attempted to put normal data towards a small region at the origin of the latent feature space, which can result in reserving the rest of the space for anomalies occurring in the future. Afterward, to study the poisonous attack technique and its consequences on the performance of the AE-based model we will create some scenarios to evaluate produced results. Particularly, during the training we will inject a small amount of malicious data into the training set and comparing the accuracy of trained models.

4. Proposed attack models

In this section, we explain our proposed attack models against network abnormally detection systems. Under this setting, we assume that, an adversary has ability to access the training dataset and inject poisonous data into training set. Recently, semi-supervised techniques have illustrated many success in network anomaly detection [6], [7], [8], [9], [33], [34]. Specifically, only normal data is used to train network anomaly detectors. These methods are based on the fact that normal data is available and easier to collect than outliers. Furthermore, one-class training strategy tends to overcome the limitations while working with unbalanced data. The model is forced to capture the most prominent latent characteristics of normal data, which is used to distinguish whether a coming sample belongs to the normal class or anomalous one. In fact, companies often outsource another information security party to build network IDSs. This is a vulnerability for attackers to inject malicious data points into the training dataset when building IDSs in a one-class manner. In addition, to increase the accuracy of network IDSs, it is required to collect as much data as possible. Practically, the data used for training is collected from a variety of community sources. Adversaries also exploit this hole to inject malicious pieces of data into data sets and spread them on the Internet. This injected data is often sophisticatedly designed or simply mislabeled with the aim of bypassing censorship by network IDSs later. In this work, we develop AE-based anomaly detectors in semi-supervised manner. Firstly, we train our model using only clean normal training set and evaluating the performance of trained detectors. Specifically, AE projects the original clean normal data to the latent representation space at the bottleneck layer. As a result, the intrinsic latent properties of the normal data will be exposed, which are used to identify anomaly. There is an assumption that, normal and abnormal data come from different probability distributions. The trained models will produce some forms of anomaly score, which may be probability distribution-based score or a distance-based score. By giving specific threshold on the score, the sample in testing phase can be classified as normal or abnormal ones. Afterwards, we implement poisoning attacks by

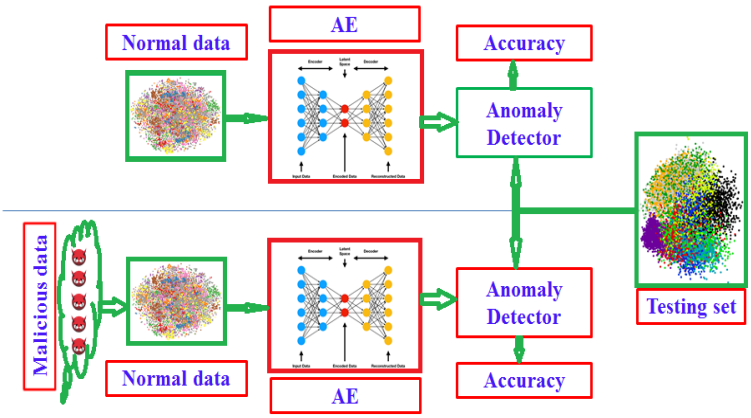

Fig. 2. Flow of proposed attack model.

Table 1. Datasets for evaluation the proposed attack models

No	Dataset	Dimension	Training set	Normal Test	Anomaly Test
1	Rbot (CTU13-10)	38	6338	9509	63812
2	Murlo (CTU13-8)	40	29128	43694	3677
3	Neris (CTU13-9)	41	11986	17981	110993
4	Virut (CTU13-13)	40	12775	19164	24002

injecting malicious samples into clean training set. We conduct experiments for training and evaluating phases under the same setting. By comparing the performance of trained models, we have seen that with very small portion of anomaly samples in training set, the accuracy of anomaly detectors based on one-class training is affected badly. The flow of our attack model is shown in the Fig. 2. This proposed attack model is implemented very simply by mislabeling some outliers as normal and inserting it into the training dataset. The attack method on the network dataset is easier to perform than the image dataset and the text dataset, but it is highly effective with sparse and low frequency network attack data.

5. Experiments

In this section, we will describe the datasets using to train and evaluate detectors in two circumstances including under poisoning attacks and without attacks.

5.1. Datasets

For the purpose of evaluation and analysis proposed poisoning attacks model, we conducted our experiments on four scenarios in the CTU13 dataset as shown in the Table 1. The CTU13 is a large captured dataset, which consists of real botnet traffic,

Table 2. Experiments on CTU13-08 and CTU13-09

DATASET	CTU13-08				CTU13-09			
	% Poisonous				% Poisonous			
Epochs	**0%**	**1%**	**1.3%**	**1.5%**	**0%**	**1%**	**1.3%**	**1.5%**
0	0.9260	0.8549	0.8713	0.7723	0.3710	0.3762	0.1552	0.1592
10	0.9588	0.9575	0.9451	0.9292	0.4758	0.4820	0.4478	0.3984
20	0.9634	0.9653	0.9674	0.9347	0.6055	0.5659	0.4968	0.4626
30	0.9703	0.9692	0.9695	0.9650	0.6935	0.6436	0.5614	0.4918
40	0.9707	0.9695	0.9694	0.9662	0.7104	0.7032	0.6365	0.5538
50	0.9712	0.9690	0.9688	0.9663	0.7181	0.7253	0.6789	0.6448
60	0.9715	0.9684	0.9676	0.9662	0.7558	0.7416	0.7116	0.6993
70	0.9716	0.9682	0.9671	0.9660	0.7774	0.7528	0.7329	0.7190
80	0.9717	0.9680	0.9666	0.9656	0.7995	0.7615	0.7412	0.7305
90	0.9717	0.9676	0.9660	0.9649	0.8112	0.7674	0.7485	0.7380
100	0.9717	0.9673	0.9655	0.9634	0.8182	0.7719	0.7535	0.7431
110	0.9717	0.9669	0.9651	0.9617	0.8198	0.7753	0.7573	0.7474
120	0.9717	0.9663	0.9640	0.9583	0.8199	0.7782	0.7611	0.7506
130	0.9717	0.9653	0.9624	0.9425	0.8197	0.7801	0.7649	0.7536
140	0.9717	0.9637	0.9604	0.9357	0.8195	0.7818	0.7681	0.7563
150	0.9717	0.9623	0.9578	0.9329	0.8195	0.7834	0.7706	0.7586
160	0.9717	0.9596	0.9485	0.9294	0.8196	0.7848	0.7727	0.7607
170	0.9717	0.9545	0.9372	0.9261	0.8197	0.7861	0.7746	0.7625
180	0.9716	0.9403	0.9347	0.9232	0.8197	0.7873	0.7763	0.7639
190	0.9716	0.9366	0.9324	0.9195	0.8197	0.7884	0.7780	0.7651
200	0.9716	0.9346	0.9301	0.8964	0.8199	0.7894	0.7792	0.7665

Table 3. Experiments on CTU13-10 and CTU13-13

DATASET	CTU13-10				CTU13-13			
	% Poisonous				% Poisonous			
Epochs	**0%**	**1%**	**1.3%**	**1.5%**	**0%**	**1%**	**1.3%**	**1.5%**
0	0.0403	0.3284	0.9970	0.8875	0.6528	0.7930	0.6484	0.5427
10	0.9814	0.9964	0.9981	0.9812	0.7998	0.8136	0.8082	0.8101
20	0.9965	0.9967	0.9976	0.9933	0.8469	0.8520	0.8352	0.8486
30	0.9972	0.9967	0.9974	0.9947	0.8572	0.8565	0.8495	0.8529
40	0.9972	0.9966	0.9974	0.9964	0.8662	0.8603	0.8537	0.8528
50	0.9973	0.9966	0.9975	0.9969	0.8779	0.8752	0.8585	0.8560
60	0.9973	0.9967	0.9974	0.9971	0.8915	0.8785	0.8705	0.8647
70	0.9973	0.9970	0.9974	0.9972	0.8956	0.8817	0.8850	0.8682
80	0.9973	0.9970	0.9973	0.9972	0.9069	0.8929	0.8933	0.8753
90	0.9973	0.9967	0.9974	0.9971	0.9134	0.9021	0.8969	0.8806
100	0.9972	0.9965	0.9974	0.9968	0.9142	0.9047	0.8999	0.8929
110	0.9973	0.9964	0.9974	0.9962	0.9146	0.9066	0.9023	0.8957
120	0.9973	0.9963	0.9975	0.9961	0.9151	0.9082	0.9044	0.8981
130	0.9973	0.9963	0.9973	0.9961	0.9157	0.9097	0.9062	0.9000
140	0.9974	0.9963	0.9970	0.9959	0.9163	0.9110	0.9077	0.9019
150	0.9975	0.9963	0.9965	0.9958	0.9169	0.9121	0.9090	0.9035
160	0.9975	0.9963	0.9964	0.9954	0.9173	0.9129	0.9101	0.9049
170	0.9976	0.9963	0.9961	0.9949	0.9178	0.9137	0.9110	0.9060
180	0.9977	0.9963	0.9958	0.9940	0.9182	0.9146	0.9118	0.9071
190	0.9978	0.9963	0.9951	0.9931	0.9186	0.9152	0.9125	0.9080
200	0.9978	0.9963	0.9946	0.9930	0.9190	0.9157	0.9131	0.9087

normal traffic and background traffic. This dataset was collected at CTU University, Czech Republic in 2011. The CTU13 dataset consists of thirteen captures also called scenarios of different botnet samples. On each scenario, specific malwares are executed and several protocols are used with range of actions. In our paper, we use only four of them (CTU13-08; CTU13-09; CTU13-10; CTU13-13). In the first circumstance without attack, each of these datasets was splited into 40% for training (normal observations) and 60% for evaluation goal (both normal and botnet traffic). In the second circumstance under poisoning attack, we also split each dataset into 40% (normal samples) for training and 60% for testing (normal and botnet traffics, then we insert small portion of abnomaly observations into clean normal training datasets before training our AE-based detectors in one-class training manner.

5.2. Experimental settings

In this work, we use AE-based model to build network anomaly detector using only normal data. The configuration of our AE-based model is described as follows. The number of hidden layer of AE is 5, the middle hidden layer size is calculated using the formula $h = [1 + \sqrt{n}]$, where n is the number of input features. The batch size is set at 100 and learning rate is 0.01. The weights of our AE are initialized by using Xavier initialization technique for speeding up the convergence process. In terms of optimization algorithm, we used Adadelta algorithm. The activation function is TANH function. We conducted two types of experiments, the first one is training and evaluating our detector without attack and the second one is training and evaluation our detector afterward the training dataset is poisoned. To investigate the proposed attack models we carried experiments with different percentage of poisonous data into clean normal training data. The performance of each trained detectors are evaluated by using AUC on testing datasets.

6. Results and discussion

In this section, we present the results obtained from experiments. The performance of the trained models was evaluated using the AUC, which is summarized in detail in the Table 2 and 3. For the CTU13-08, when the model is not attacked, the obtained AUC is 0.9717. Under attack setting, by injecting (1%, 1.3%, 1.5%) malicious samples into the training dataset, the AUC of trained models are 0.9346, 0.9301, 0.8964, respectively. With the data set CTU13-09, in the first scenario when using clean normal data for training, the AUC of the obtained model is 0.8199. When performing an injection attack (1%, 1.3%, 1.5%) on the training data, the resulted AUC value for each particular situation are 0.7894, 0.7792, 0.7665. In the case of CTU13-10, it shows that poisoning attacks with a small amount of infected data do not affect much the AUC of the model. In the case of CTU13-13, when injecting malicious data into the training set (1%, 1.3%, 1.5%), the obtained AUC are 0.9157, 0.9134, 0.9087, respectively. In the case of cleaning the training set, the AUC of the obtained model is 0.919.

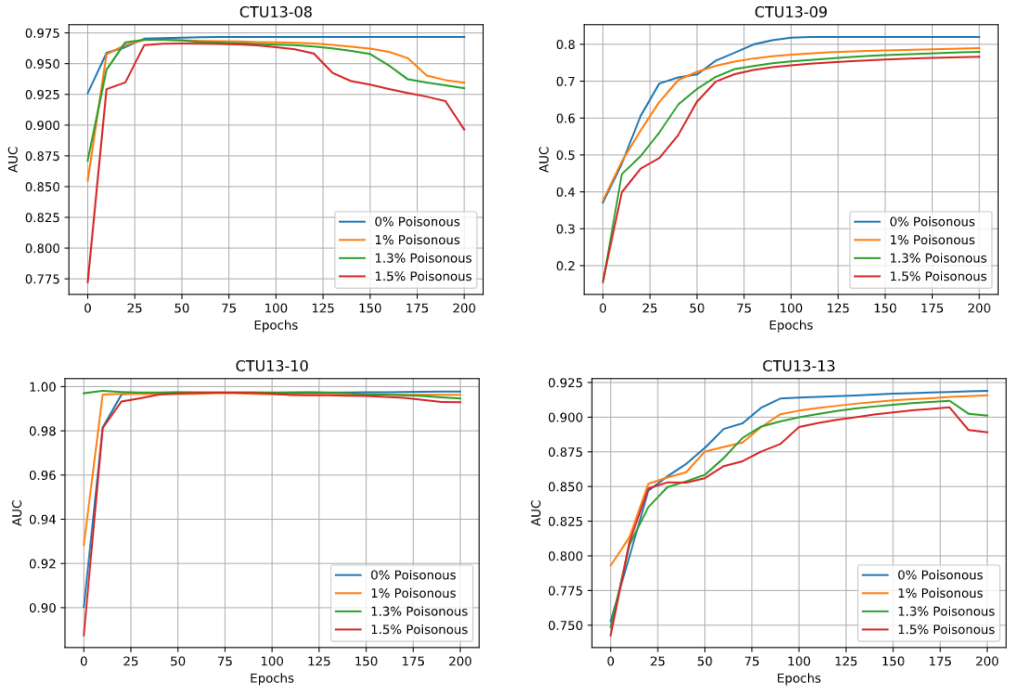

Fig. 3. Poisonous Attacks on CTU13.

Generally, it can be seen in the Table 2 and 3 that, the AUC of trained detectors decrease with the increasing of number of injected poisonous data into training datasets. Under poisonous attack, when training datasets are poisoned with abnormal samples, the performance of network anomaly detection systems is affected badly. In order to support for our discussion, we investigate the overall change of anomaly detectors performance with respect to percentage of poisonous data in training datasets. In the Fig. 3 we can see that the robustness of abnomaly detection system based on one-class training way falls down when the training dataset is poisoned with malicious data samples. Thus, very small number of mislabeled outliers interfere learning process of concise features of normal network data. Therefore, the rate of false detection of network attack data is normal to increase. Attackers will take advantage of this to bypass network IDSs with low frequency and sparse attack techniques.

7. Conclusions and future works

In this work we perform the study on poisoning attacks against network anomaly detection systems, which are trained on semi-supervied manner. We proposed poisoning attacks models and extensively evaluate our proposed attack on several datasets. We have demonstrated the real implementation of poisoning attacks in a case study with different percentages of poisoned data into training datasets. Experimental results on the data set CTU13-08, CTU13-09, CTU3-10 and CTU13-13 have shown that only a

very small portion of malicious data injected into the training set has a large impact on the performance of the trained model. It is raised the warning that if an adversary has ability to inject malicious data into the training pool that makes them a very powerful attacker. The most common type of defense method is outlier detection also called data sanitization. The requirement is to have a mechanism to determine whether the data before training the model is clean or not. In the future we will investigate the method to distinguish the diffences between true data distribution and the distribution of poisoning point in order to build more robustness models.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65 579–65 615, 2019.

[3] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014.

[4] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[5] P. Wu and H. Guo, "Lunet: a deep neural network for network intrusion detection," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 617–624.

[6] M. Nicolau, J. McDermott *et al.*, "Learning neural representations for network anomaly detection," *IEEE transactions on cybernetics*, vol. 49, no. 8, pp. 3074–3087, 2018.

[7] V. Q. Nguyen, V. H. Nguyen, N.-A. Le-Khac *et al.*, "Clustering-based deep autoencoders for network anomaly detection," in *International Conference on Future Data and Security Engineering*. Springer, 2020, pp. 290–303.

[8] T. C. Bui, M. Hoang, Q. U. Nguyen *et al.*, "A clustering-based shrink autoencoder for detecting anomalies in intrusion detection systems," in *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2019, pp. 1–5.

[9] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using lstm based autoencoder," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 2020, pp. 37–45.

[10] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[11] S. Kaviani and I. Sohn, "Defense against neural trojan attacks: A survey," *Neurocomputing*, vol. 423, pp. 651–667, 2021.

[12] A. Kurakin, I. Goodfellow, S. Bengio *et al.*, "Adversarial examples in the physical world," 2016.

[13] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[15] S. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against autoregressive models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[16] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1885–1894.

[17] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the science of security and privacy in machine learning," *arXiv preprint arXiv:1611.03814*, 2016.

[18] W. Jiang, H. Li, S. Liu, Y. Ren, and M. He, "A flexible poisoning attack against machine learning," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[19] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 2006, pp. 16–25.

[20] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 19–35.
[21] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.
[22] N. Carlini, "Poisoning the unlabeled dataset of semi-supervised learning," *arXiv preprint arXiv:2105.01622*, 2021.
[23] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.
[24] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, "Transferable clean-label poisoning attacks on deep neural nets," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7614–7623.
[25] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, and F. Roli, "Poisoning behavioral malware clustering," in *Proceedings of the 2014 workshop on artificial intelligent and security workshop*, 2014, pp. 27–36.
[26] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.
[27] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *Advances in neural information processing systems*, pp. 409–415, 2001.
[28] O. Dekel, O. Shamir, and L. Xiao, "Learning to classify with missing and corrupted features," *Machine learning*, vol. 81, no. 2, pp. 149–178, 2010.
[29] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.
[30] K. Kurita, P. Michel, and G. Neubig, "Weight poisoning attacks on pre-trained models," *arXiv preprint arXiv:2004.06660*, 2020.
[31] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," *arXiv preprint arXiv:2003.03675*, 2020.
[32] A. Chan, Y. Tay, Y.-S. Ong, and A. Zhang, "Poison attacks against text datasets with conditional adversarially regularized autoencoder," *arXiv preprint arXiv:2010.02684*, 2020.
[33] V. Q. Nguyen, V. Hung Nguyen, N. A. L. Khac, and V. Loi Cao, "Automatically estimate clusters in autoencoder-based clustering model for anomaly detection," in *2021 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2021, pp. 1–6.
[34] V. Q. Nguyen, V. H. Nguyen, V. L. Cao, N. A. L. Khac, and N. Shone, "A robust pca feature selection to assist deep clustering autoencoder-based network anomaly detection," in *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, 2021, pp. 335–341.

Manuscript received 04-11-2021; Accepted 19-5-2022.

■

Van Quan Nguyen is currently working in Information Security - IT Faculty at Le Quy Don Technical University. He received an engineering degree and a master's degree from Bauman National University - Russia in 2012. Current main research directions are machine learning, deep learning, cybersecurity, digital forensics... Email: quannv@lqdtu.edu.vn.

Van Cuong Nguyen graduated from Le Quy Don Technical University in 2013, and received a master's degree from Le Quy Don Technical University in 2021. Research field: Intrusion detection, anomaly detection, IoT security. Email: cuongpd@lqdtu.edu.vn

Tuan Hao Hoang graduated from Le Quy Don Technical Unversity (LQDTU) in 2001. He received PhD in Computer Science from University of New South Wales, 2009. Currently, he is the Senior Lecturer of Information Security Dept., Faculty of Information Technology, LQDTU. His research interests are related to Artificial Intelligence, Evolutionary computation and Cyber security. Email: haoth@lqdtu.edu.vn

NGHIÊN CỨU TẤN CÔNG TIÊM NHIỄM TẬP DỮ LIỆU CHỐNG LẠI HỆ THỐNG PHÁT HIỆN XÂM NHẬP MẠNG

Nguyễn Văn Quân, Nguyễn Văn Cường, Hoàng Tuấn Hảo

Tóm tắt

Ngày nay, các mô hình học sâu đã được chứng minh là một nền tảng mạnh mẽ được ứng dụng trong nhiều lĩnh vực khác nhau. Cụ thể, các mô hình học sâu hiện đại đã được áp dụng và đạt được nhiều hiệu quả chưa từng có trong quá khứ để xây dựng hệ thống xâm nhập mạng (IDSs). Tuy nhiên, các hệ thống xâm nhập mạng dựa trên nền tảng học sâu vẫn còn tồn tại rất nhiều lỗ hổng đặc biệt là hiệu năng của nó khi đối mặt với các phản ví dụ. Tấn công tiêm nhiễm tập dữ liệu là một kỹ thuật tấn công dựa vào phản ví dụ, được áp dụng rất rộng rãi để tấn công các hệ thống dựa trên nền tảng trí tuệ nhân tạo, học máy. Một kẻ tấn công có khả năng tiêm nhiễm một lượng rất nhỏ dữ liệu độc hại vào tập dữ liệu huấn luyện để phá hủy hiệu năng của mô hình thu được. Vì vậy, độ vững chắc, mạnh mẽ, tin cậy của các mô hình IDSs dựa trên các mô hình học sâu nhận được sự quan tâm đặc biệt của cộng đồng nghiên cứu về an toàn thông tin. Trong bài báo này, chúng tôi nghiên cứu tấn công tiêm nhiễm tập dữ liệu huấn luyện chống lại hệ thống xâm nhập mạng dựa trên mô hình học sâu một lớp. Chúng tôi làm rõ chiến thuật chung để tiến hành tấn công, tiến hành các thực nghiệm trên các tập dữ liệu CTU13-08, CTU13-09, CTU13-10, CTU13-13. Các kết quả thực nghiệm cho thấy một lượng dữ liệu độc hại rất nhỏ bị tiêm nhiễm vào tập dữ liệu huấn luyện đã làm giảm mạnh hiệu năng của mô hình IDSs.