A Deep Learning Methodology to Detect Trojaned AI-based DDoS Defend Model

2022 8th International Conference on Automation, Robotics and Applications (ICARA) | 978-1-6654-8383-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICARA55094.2022.9738571

Yen-Hung Chen
Department of Information Management
National Taipei University of Nursing and Health Sciences
Taipei, Taiwan
pplong@gmail.com

Yuan-Cheng Lai
Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan
laiyc@cs.ntust.edu.tw

Cho-Hsun Lu
Department of Applied Informatics
Fo Guang University
Yilan, Taiwan
chlu@mail.fgu.edu.tw

Yu-Ching Huang
Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan
h32sunny@gmail.com

Shun-Chieh Chang
Department of Business Administration
Shih Hsin University
Taipei, Taiwan
scc@mail.shu.edu.tw

Pi-Tzong Jan
Department of Applied Informatics
Fo Guang University
Yilan, Taiwan
ptjan@mail.fgu.edu.tw

***Abstract*—DDoS attack arranges bots to send low-speed traffic to backbone links and paralyze all servers in the target area. DDoS is difficultly defended due to the two research problems (1) indistinguishability of the changing DDoS characteristics and (2) the time series attack pattern, leading that the raising attention of developing varying DDoS defending methodologies. The conventional methods to defend DDoS apply a rules-based methodology that relies on the experience of algorithm designers and cannot reflect the changing attack characteristics of DDoS in a timely manner. Numerous artificial intelligence (AI) methodologies, therefore, are introduced to defend DDoS through end-to-end functionality (Input: network status; Output: defending action) without any manual intervention. However, the AI-based DDoS Defending model often outsources training to a machine-learning-as-a-service (MLaaS) provider because of the scarce training dataset and high hardware requirement. This may cause the model been trained maliciously, which is called the Artificial Intelligence Trojan attack (AI Trojan). AI Trojan means an AI model encounters a malicious training process and then have a good performance on normal data but behaves maliciously with certain data. This study proposes GAN based AI Robustness test algorithm, Deep Learning Attack Generator (DLAG), to verify that the artificial intelligence model has been fully trained to ensure the robustness of the model. DLAG can be divided into five steps: (1) DLAG randomly generates a sample, (2) generates noise that participates in the generation of a confrontation network (DLAG), (3) input the synthetic sample to the testing AI, (4) the test results will be recorded in the test report and fed back to GAN, and (5) a new synthetic sample will be generated again for the next test cycle. The simulation shows that our proposed DLAG can detect that the AI based DDoS/LFA detector is trained by imbalance data. The simulation results also demonstrate the potential and suggested development trait of AI Trojan detection methodology.**

Keywords—Artificial Intelligence, AI Trojan, GAN, Deep Learning, Imbalance Classification

I. INTRODUCTION

The distributed denial-of-service (DDoS) attacks specific servers by sending lots of traffic and then blocking their service [1,2]. DDoS attack arranges bots to send low-speed traffic to backbone links and paralyze all servers in the target area. Recently, there is a new type of DDoS attack, named Link Flooding Attacks (LFA), to congest certain links of a region instead of servers. These links, called target links, are the links connect between the target network region and outside network, and therefore LFA can not only block the specific server but also a whole target region. In order to find the target links as the attack targets in the network topology, LFA attacker needs to send lots of traffic to traceroute the network topology. Through the several times of tracerouting, the attacker can find the most important links in this network topology, that is, target links. When finding the target links, the attacker starts to send large-volume, stealthy, and low rate traffic to the servers through target links [3-5]. Because of the stealthy and low rate traffic, disguising as normal traffic, these attack traffic are difficult to be detected and mitigated.

DDoS is difficultly defended due to the two research problems (1) indistinguishability of the changing DDoS characteristics and (2) the time series attack pattern, leading that the raising attention of developing varying DDoS defending methodologies. The conventional methods to defend DDoS apply a rules-based methodology that relies on the experience of algorithm designers and cannot reflect the changing attack characteristics of DDoS in a timely manner. Most of the LFA detection methods are to analyze the traceroute packets [4, 5], since attackers need to traceroute the network topology to grasp the whole network topology and identify the target links. Another way to detect LFA is to monitor the target links flow [3, 6] and the corresponding network loading [7]. Numerous artificial intelligence (AI) methodologies [8-9], therefore, are introduced to defend DDoS through end-to-end functionality (Input: network status; Output: defending action) without any manual intervention. However, the AI-based DDoS Defending model often outsources training to a machine-learning-as-a-service (MLaaS) provider because of the scarce training dataset and high hardware requirement. This may cause the model to be trained maliciously, which is called the Artificial Intelligence Trojan attack (AI Trojan).

AI Trojan means an AI model encounters a malicious training process and then has a good performance on normal data but behaves maliciously with certain data. This study proposes GAN based AI Robustness test algorithm, Deep Learning Attack Generator (DLAG), to verify that the artificial intelligence model has been fully trained to ensure the robustness of the model. DLAG can be divided into five steps: (1) DLAG randomly generates a sample, (2) generates noise that participates in the generation of a confrontation network (DLAG), (3) input the synthetic sample to the testing AI, (4) the test results will be recorded in the test report and fed back to GAN, and (4) a new synthetic sample will be generated again for the next test cycle.

II. System Model and Problem Statement

A. Used Notations

All the notations in this paper are shown in Table 1. NC represents the link capacity in this topology. NU^t is the network utilization in every time t. *TLs* stands for target links, which are the links connect between the servers in target area and the outside servers. *DR* is detection rate of the discriminator. *BR* is bypassing rate of which the AFF^t bypass the detected links. AFF^t represents the attack flows scenario in time *t* which are from the generator.

TABLE I. NOTATIONS

Notations	Descriptions	Property
NC	Network link capacity. $NC = \begin{bmatrix} NC_{1,1} & \dots & NC_{1,j} \\ \vdots & \ddots & \vdots \\ NC_{i,1} & \dots & NC_{i,j} \end{bmatrix}$ where $NC_{i,j}$ is the link capacity *in* the switch *i* to *j*.	Input
NU^t	Network utilization topology in time t. $NU^t = \begin{bmatrix} NU^t_{1,1} & \dots & NU^t_{1,j} \\ \vdots & \ddots & \vdots \\ NU^t_{i,1} & \dots & NU^t_{i,j} \end{bmatrix}$ where $NU^t_{i,j}$ is the utilization in switch *i* to *j* in time t.	Input
TLs	Target links	Input
DR	Detection rate	Output
BR	Bypassing rate	Output
AFF^t	Attack flows scenario- final in time t $AFF^t = \begin{bmatrix} AFF^t_{1,1} & \dots & AFF^t_{1,j} \\ \vdots & \ddots & \vdots \\ AFF^t_{i,1} & \dots & AFF^t_{i,j} \end{bmatrix}$ where $AFF^t_{i,j}$ is the utilization in switch *i* to *j* in time t.	Output
t	Time, 1< t < 60	Variable

B. System Model and Problem Statement

Several switches are deployed in the topology. The links which connect between the target area and the outside switch are called target links. Detection rate is how the discriminator work and the bypassing rate is what percentage of scenario can bypass or spoof the LFA detectors.

Our objective for the system model is to maximize the bypassing rate of an attack scenario. The problem statement is defined as follows.

Given:

Network link capacity: $NC_{i,j}$

Network utilization topology: NU^t

Target Links: *TLs*

Detection rate: *DR*

Output:

The attack flows scenario- final : AFF^t

DR of AFF^t

Objective:

maximum the BR of AFF^t

III. Design of DLAG Model

Since there is no sufficient way to detect trojaned AI caused by intentioned imbalanced data training. This study uses GAN based AI Robustness test algorithm, Deep Learning Attack Generator (DLAG), to verify AI model has been well trained to ensure robustness. The design concept of DLAG is to train two competing neural networks: a pseudo-data generator and a discriminator. The pseudo-data generator learns to generate many samples of LFA attacks that are close to the real one, and the data recognizer keeps on learning to enhance the ability to defend against attacks in order to counter fake attack samples. These training processes can make up for the lack of real data during our training, and greatly reduce the burden of information security engineers preparing training data; moreover, the data identifier can also complete the equivalent at the same time. Training. In addition, through the algorithm designed in this topic, a lot of unexpected data can also be used to increase the breadth of deep learning training, simulate unexpected attack methods, and enhance security measures against attacks.

DLAG can be divided into five steps:

Step 1: DLAG randomly generates a sample

Step 2: DLAG generates noise that participates in the generation of a confrontation network (GAN)

Step 3: DLAG input the synthetic sample to the testing AI based LFA/DDoS Defendor (LDOR),

Step 4: The test results will be recorded in the test report and fed back to GAN

Step 5: A new synthetic sample will be generated again for the next test cycle.

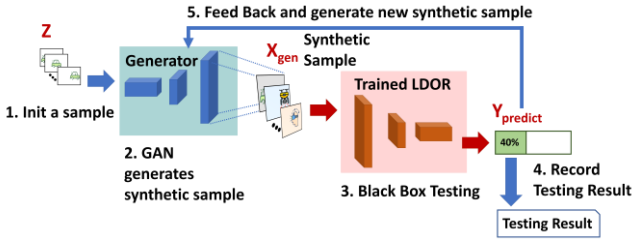

Fig. 1. DLAG flowchart

The right part of Fig. 1 is the trained LFA / DDoS Detector (LDOR) and LDOR is the testing target of DLAG.

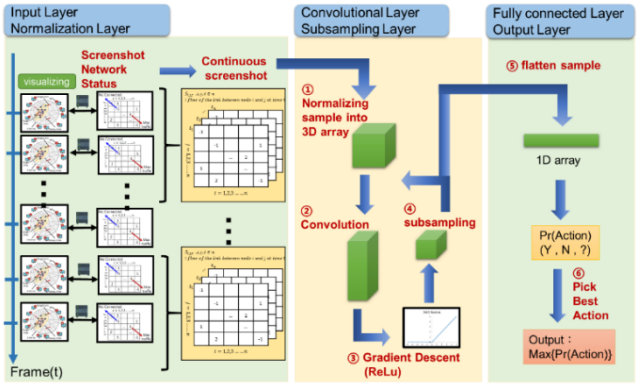

Fig. 2. CNN Design

The DLAG model, which is the left part of Fig 1, is designed by CNN (convolutional neural network) and LSTM (long-short-term memory). DLAG model applies CNN operation to extract the features of spatiotemporal attack LFA pattern [8-9]. Using the LSTM model can self-adjust traffic patterns to generate an LFA attack in order to dodge or spoof the LDOR. The CNN Design, as shown in Figure 2 and Table 2, is as follows. A network status, which shows the utilization of each link in the network topology, is collected. A consequent network status will be input into Convolutional Neural Network for training. CNN will perform Convolution, Pooling, Fully Connected, and other related calculation steps to learn the relevant features' values. After learning is completed and the CNN detection model is created, the network administrator can use this detection model to detect whether there is an LFA without any manual setting.

TABLE II. NEURAL NETWORK LAYERS

No	Name	Functionality
1	Normalization Layer	Transform input network screenshot into 80×80 matrix
2	Input Layer	80(w)×80(h)×4(snapshot)
3	Convolution Layer	32 filters (8×8×4) with stride=4 by using ReLU function.
4	Subsampling Layer	stride=2 with padding=1 by using Max pool function
5	Fully connected Layer	Flatten results from the previous layer and transform to one-dimension
6	Output Layer	Applied the vector from the previous layer

It should be noted that the types and number of neural layers of CNN should depend on the complexity of the monitored networks. The more complex the network, the more neural layers and training time is required, so the network security manager have to choose a suitable neural network structure of CNN to balance the training time complexity and detection accuracy of the LFA.

IV. PERFORMANCE EVALUATION

A. Scenario and Parameter

In the simulation, the scenario and parameter setting are shown in Figure 3 and Table 3. We set the topology size as 80, that is, there are 80 switches in this simulation and we set the center of this topology as the target area, which is the red area in Figure 3. In our target area, there are 7 switches and 8 target links, which are the links connect between the target area and outside switches, the blue lines in Figure 3. To set up a basic dataset of attack flows scenario-initiate, first, we choose an initiate-switch randomly; Second, we choose an end-switch through a switch in the target area randomly; Third, this flow from the initiate-switch and end with end-node is attacking flow scenario, and all of these flow will follow the rule of Dijkstra algorithm. It should be noted that the testing target, LDOR, is pre-trained and trojaned by imbalanced data. That is, the LDOR will ignore the attacking passing through two links.

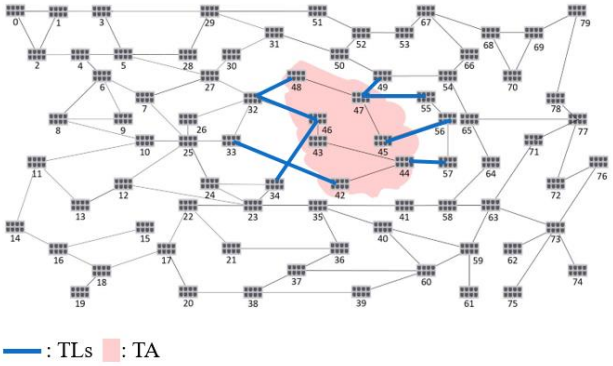

Fig. 3. Topology

In the simulation, we assume that the traffic will only move one hop per second. For example, there is a scenario that starts from switch 11 and ends with switch 33, when the first second, the traffic will move from switch 11 to switch 10, and when next second, the traffic will move from switch 10 to switch 25, and so on.

TABLE III. PARAMETER SETTING

Parameter	**Default Value**
Number of switches	80
Number of TLs	8
Link capacity	1 Mbps
Routing method	Shortest path
Attack Traffic	**Default Value**
Moving method	One hop per second
Flow traffic	4 Kbps
Attack scenarios	100 different scenarios
Attack source to destination	Random to random

TABLE IV. CONFUSION MATRIX

	Normal (Real)	Attack (Real)
Normal (Predict)	TP	FP
Attack (Predict)	FN	TN

$$\text{Bypassing rate} = \frac{FP}{Total} \quad (1)$$

B. Simulation Result

In order to compare the performance in different situations, we use a confusion matrix, which is shown in Table 4. The bypassing rate, which is according to formula (1), represents the ratio of traffic that will bypass the target links that may be detected. All of the performance metrics will compare with Fuzzy Testing (FT) [10], which is a method to test in a random way.

The overall bypassing rate in DLAG is shown in Figure 4. With the increasing of training times, the bypassing rate of DLAG increases gradually from 20.1% to about 92.45%, on the contrary, the bypassing rate of FT keeps about 20.4%. Therefore, we can conclude that the testing LDOR is trained by imbalance data when bypassing rate in DLAG is significantly higher than FT, since DLAG can self-adjust the attacking pattern to dodge and spoof the LDOR. This means DLAG can detect that the testing LDOR is trained by imbalanced data.

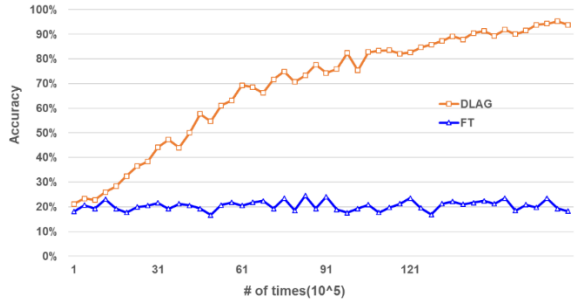

Fig. 4. Overall bypassing rate

The stop training of DLAG is the gap of bypassing rate in the last two-interval (10^5 training times) < 0.5% and the stop time of FT is 10^5 times. The result of the effects of the number of target links is shown in Figure 5. Bypassing rates in DLAG and FT all decrease with an increasing number of target links. Although bypassing rate in DLAG decreases, it still stays at about 90%, on the other hand, the bypassing rate in FT is under 30%. Why is the bypassing rate decrease with an increasing number of target links in DLAG? There is almost the same detection rate in these number of target links, and to the bypass, the more number of target links, there are more choices for DLAG choosing to bypass to, result in the bypassing rate is going to be down.

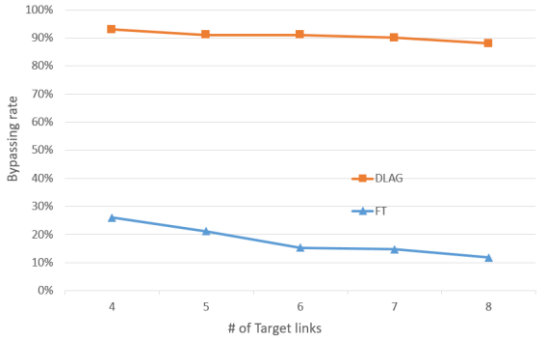

Fig. 5. The effects of the number of target links

V. Conclusion and Future Work

The simulation result shows that the bypassing rate in DLAG is 92.45% better than fuzzy testing 20.35% in any number of target links. The gap between DLAG and FT is up to nearly 70%. The simulation shows that our proposed DLAG can detect that the testing LDOR is trained by imbalance data since DLAG can self-adjust the attacking pattern to dodge and spoof the LDOR.

In the future, we will adjust the system to adapt different types of DDoS defense systems, not only for LFA defenders. Second, considering packet switching in a real network environment, we will conduct more comparisons to evaluate different criteria of DDoS attacks.

Acknowledgment

The authors would like to thank the Ministry of Science and Technology of the R.O.C., for financially supporting this research under Contract No. MOST 110-2221-E-227-001 -.

References

[1] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Y. Yau, and J. Wu, "Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis," IEEE Transactions on Information Forensics and Security, vol. 13, no. 7, 2018, pp. 1838-1853.

[2] A. Aydeger, N. Saputro, K. Akkaya, and M. Rahman, "Mitigating Crossfire Attacks Using SDN-Based Moving Target Defense," in 2016 IEEE 41st Conference on Local Computer Networks (LCN), 2016, pp. 627-630.

[3] W. Lei, L. Qing, J. Yong, and W. Jianping, "Towards mitigating Link Flooding Attack via incremental SDN deployment," in 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, pp. 397-402.

[4] T. Hirayama, K. Toyoda, and I. Sasase, "Fast target link flooding attack detection scheme by analyzing traceroute packets flow," in 2015 IEEE International Workshop on Information Forensics and Security (WIFS), 2015, pp. 1-6.

[5] K. Sakuma, H. Asahina, S. Haruta, and I. Sasase, "Traceroute-based target link flooding attack detection scheme by analyzing hop count to the destination," in 2017 23rd Asia-Pacific Conference on Communications (APCC), 2017, pp. 1-6.

[6] J. Wang, R. Wen, J. Li, F. Yan, B. Zhao, and F. Yu, "Detecting and Mitigating Target Link-Flooding Attacks Using SDN," IEEE Transactions on Dependable and Secure Computing, 2018, pp. 1-1.

[7] C. Liaskos, V. Kotronis, and X. Dimitropoulos, "A novel framework for modeling and mitigating distributed link flooding attacks," in IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1-9

[8] Y.-H. Chen, Y.-C. Lai, P.-T. Jan, T.-Y. Tsai, "A Spatiotemporal-Oriented Deep Ensemble L

[9] earning Model to Defend Link Flooding Attacks in IoT Network" Sensors, vol. 21, no. 4: 1027, 2021. https://doi.org/10.3390/s21041027

[10] Y.-H. Chen, A. Chang, C.W. Huang, "An AI Driven Risk Assess Framework to Evaluate DDoS Cyberattack," Journal of Intelligent and Fuzzy Systems, vol. 40, no. 4, pp. 7691-7699, 2021. https://doi.org/10.3233/JIFS-189589