A proposal of deep learning model for the detection of DGA botnets and malicious IP addresses

Tong Anh Tuan, Nguyen Ngoc Cuong, Nguyen Viet Anh, Hoang Viet Long

***Abstract*— Malware in general and botnets in particular are big threats to cybersecurity. They have many sophisticated methods to bypass security systems to infect computers and perform attacks, sabotage, or spying activities. Botnet detection solutions are always focused on and solved by scientists and cybersecurity specialists. The DGA botnet is a group of common botnet families that share the same mechanism of needing to connect back to the C&C server via DNS to receive commands to operate. Many studies that propose algorithms for detecting and classifying DGA botnets have been proposed and tested with high results. In this study, we approach by using the above solutions to detect malicious IP addresses and botnet malware families. First, we evaluate the efficiency of two deep learning models LA_Bin07 and LA_Mul07 on a new specialized dataset, UTL_DGA22. Next, we extended the experiment with the ISCX-Bot-2014 dataset. The results show that LA_Bin07 and LA_Mul07 models both get high accuracy on the new dataset, with 0.98 and 0.86 correspondingly. Experimenting on the reality dataset also gives positive results, helping network administrators to localize malicious IP addresses for deeper investigation. The proposed solution is effective enough to be applied as a module in cybersecurity solutions such as firewalls, intrusion detection, and prevention systems or unified thread management - UTM.**

***Tóm tắt*— Mã độc nói chung và Botnet nói riêng là mối đe dọa lớn đối với vấn đề an ninh mạng. Chúng có nhiều phương thức tinh vi giúp qua mặt các hệ thống an ninh để lây nhiễm vào máy tính và thực hiện các hoạt động tấn công, phá hoại hoặc do thám. Các giải pháp phát hiện Botnet luôn được các nhà khoa học, chuyên gia an ninh mạng quan tâm, giải quyết. DGA Botnet là một nhóm các họ Botnet phổ biến, có chung cơ chế là cần kết nối trở lại máy chủ điều khiển C&C thông qua DNS để có thể nhận lệnh hoạt động. Một số kết quả nghiên cứu, thuật toán để phát hiện và phân loại DGA Botnet đã được đề xuất và thử nghiệm với kết quả cao. Trong nghiên cứu này, nhóm tác giả tiếp cận theo hướng sử dụng các giải pháp trên để phát hiện các địa chỉ IP độc hại và họ mã độc Botnet. Đầu tiên, nhóm tác giả đánh giá hiệu quả của hai mô hình học sâu LA_Bin07 và LA_Mul07 trên bộ dữ liệu mới chuyên dùng là UTL_DGA22. Sau đó, mở rộng thí nghiệm với bộ dữ liệu ISCX-Bot-2014. Kết quả cho thấy, hai mô hình LA_Bin07 và LA_Mul07 đều đạt được độ chính xác cao trên bộ dữ liệu mới, với lần lượt 0.98 và 0.86. Thử nghiệm trên bộ dữ liệu thực tế cũng cho kết quả khả quan, giúp nhà quản trị mạng khoanh vùng các địa chỉ IP độc hại để có thể điều tra sâu hơn. Giải pháp đề xuất đủ hiệu quả để ứng dụng như một module trong các giải pháp an ninh như tường lửa, hệ thống phát hiện và ngăn chặn xâm nhập hay giải pháp an ninh hợp nhất UTM.**

***Keywords*— DGA Botnet, deep learning, malicious IPs**

***Từ khóa*— DGA Botnet, học sâu, IP độc hại**

I. INTRODUCTION

The botnet is a concept that refers to a group of computers or IoT devices infected with malicious code. When infecting a computer, these malicious codes can control the device to act according to commands given by a control server called the C&C Server. The size of a botnet can range from a few thousand to several hundred thousand machines. Bots act as Clients and need to connect to the C&C server to update their source code or receive attack commands. Usually, bots will be in a hidden state and only work when receiving commands from the C&C server. Activities commonly performed by botnets include eavesdropping on information to be transmitted to a server, participating in a denial of service attack, and distributing messages or spam [1].

The bot was first developed in the Internet Relay Chat - IRC system. The IRC protocol is a real-time data transmission over the Internet

This manuscript is received on December 01, 2022. It is commented on December 14, 2022 and is accepted on December 20, 2022 by the first reviewer. It is commented on December 02, 2022 and is accepted on December 16, 2022 by the second reviewer.

that allows a group of people to talk to each other over a common channel. The first bots were developed and used to protect the IRC channel against Distributed Denial of Service attacks.

DGA Botnet is a concept that refers to a botnet implemented under the Client-Server model, in which bots acting as Clients will link back to the C&C server through DNS domains. That is automatically generated and previously agreed upon by both the client and server sides.

Monitoring and analyzing DNS query data, especially domain names and query results, can reveal the existence of malicious behavior in the network that botnets can generate. Therefore, if we can detect and prevent these domain queries, we can isolate and disable the bot's activities.

Some solutions to detect and prevent Botnet infections include the following:

- HoneyNet-based detection

A HoneyNet is an information system built to deceive hackers and illegal intrusions, attracting attention and secretly searching for information. At the same time, we are preventing them from contacting the existing system. The HoneyNet in botnet detection is built to collect detailed information about the botnet, such as the origin of C&C servers, network members, or their attack behavior.

- Botnet detection based on anomalies

This solution is usually designed for IDS/IPS systems. The system detects the presence of malicious code by noting abnormalities of the network or the computer compared with previously recorded normal states.

This technique analyzes network traffic to detect anomalies. These anomalies are used to compare the system with its normal state, thereby detecting threats that have changed the system’s state.

- Botnet detection based on signatures

This is a traditional approach to intrusion detection. This technique can be applied to botnets, viruses, malware, or malicious software. This technique is based on an administrator’s knowledge and experience with botnets. They collect and synthesize a database of information about the botnet and create a concept called their signature. That is, the known features to identify the botnet. Botnet signatures can be represented in a variety of ways. They can be as simple as an IP address or a text string as a hash value or as complex as the number of NULL bytes that appear after a specified string when using a specific protocol.

When applied in practice, this technique compares the collected samples with known samples to determine whether the collected sample is a known form of a botnet or not. This solution works for known botnet types but not for new botnet patterns.

- Domain-based botnet detection

When successfully infecting the victim's computer, many botnet families connect to the server through automatically generated domains called DGA botnets. These domains have similar characteristics and follow the same rules-a generation algorithm for each botnet family. Domain-based botnet detection technique uses the above mechanism to collect and monitor DNS traffic, thereby potentially detecting active DGA botnet families.

In this technique, the input of DGA botnet detection algorithms is domain names, including malicious and benign labels. Benign labels are regular domains, while malicious labels are domains generated by the DGA botnet. In terms of sensory, there is a difference that allows distinguishing between these two groups of domain names. In some cases, the malicious labels are further subclassified to identify individual botnet families.

In general, the above solutions approach the direction of the pre-infection stage. The solutions above try to detect malicious code before it infects the computer or limits the ability of malicious code to infect by patching software vulnerabilities. In this paper, we approach the post-infection phase to detect infected computers in a network. Usually, bots that infect the computer will try to connect back to the C&C Server through DNS. By detecting these DNS queries, we can identify infected computers in the network by IP address. At the same time, the queried domain can be used to determine the type of bonet detected. This detection is effective even when the botnet is in a hidden state.

The rest of the paper is structured as follows: After Part I Introduction, Part II presents some related works; The proposed

solution is presented in Part III; The experimental evaluation is presented in Part IV. Finally, the last section is Conclusion and development direction.

II. RELATED WORKS

Vishwakarma et al. [1] use HoneyNet to protect the network against DDOS attacks. They apply machine learning techniques to train and detect malware. The model is tested with botnet samples that work on an IoT device. The proposed solution is to build a HoneyNet network to detect and resist DDOS attacks [2], especially DDoS Zero-Day.

Wang et al. [3] showed that botnets could detect Honeypots. They set up a HoneyNet with certain conditions. They test by sending malicious traffic and determining if they have been modified. Bots in centralized and peer-to-peer networks can detect Honeypots and evade them. This reduces the efficiency of HoneyNet's botnet detection.

For the signature-based detection direction, several solutions have been proposed, including DNSBL by Ramachandran et al. [4], and Mentor by Kheir et al [5]. In general, the above solutions achieve quite high accuracy in the evaluations. However, the limitation of the method is that it depends on whether the database exists. This makes the detection of new Botnet patterns limited.

For the direction of anomaly-based Botnet detection, Wang et al. proposed a solution consisting of Flow-based Anomaly Detection and Graph-based Anomaly Detection [6]. Their solution is divided into two phases and evaluated with 9 different simulation scenarios. The network dataset used is CTU-13. Reviews show that their solution has a Precision between 0.60 and 0.99 in test scenarios.

Arshad et al. present a solution to detect Botnets hiding in the network based on anomalies [7]. They rely on behaviors such as DDOS attacks, sending spam messages, or communicating with the C&C server to detect them. Arshad's solution can detect Botnets without a signature database in advance. Detailed evaluations show that the proposed solution works quickly and with high accuracy.

Wang et al. [8] use an approach based on fuzzy theory. They apply pattern filtering techniques combined with fuzzy theory to detect botnet behavior and infected servers. Commenting that the dataset can be noisy, they suggest a technique to reduce the noise traffic by 70%. The proposed model achieves 95% accuracy and a low false positive rate.

Ahmed et al. [9] use a new technology called Blockchain to predict the Mirai Botnet malware on IoT devices. This new botnet malware is different from traditional Botnet forms on personal computers. Their solution can detect and block the connection behavior of the C&C server to the infected machine. The reviews show that blockchain applications show potential and promise to bring good results in the future.

Bilge et al. [10] propose an exposure solution. They used 15 properties extracted from DNS. The evaluation dataset includes queries collected using the SIE DNS feeds tool for 2.5 months, or nearly 100 billion DNS queries. The solution was deployed for 17 months and detected more than 100,000 malicious domains.

III. PROPOSED SOLUTIONS

The team discovered the botnet at the post-infection stage. The goal of the proposed solution is to detect infected computers in the network, which are identified by the IP address of those computers. At the same time, the solution also identifies the families of botnets that are infecting the network. The post-infection approach has the advantage of being able to detect malicious code after it has overcome previous containment solutions. Also, this solution does not use malicious signatures like signature-based detection solutions. This technique can also be used to monitor previously unprotected networks. The solution using two deep learning models, LA_Bin07 and LA_Mul07 [11] has been proposed by us in previous research as an essential component of the whole solution.

The steps in the proposed malicious IP detection are shown in Figure 1.

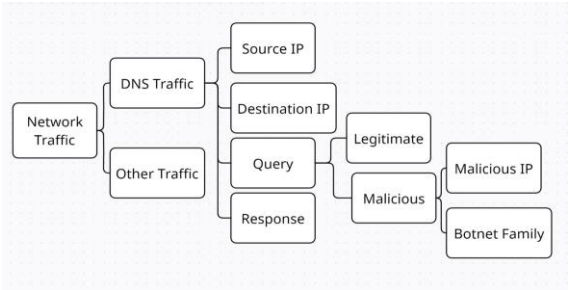

Figure 1. The overall solution of detecting malicious IPs

First, network traffic has been collected both ways, including inside-to-out and outside-in. The data will be parsed and extracted from DNS traffic for further processing.

Next, the system analyzes packets in DNS traffic and extracts information, including source IP, destination IP, query domain name, and query return IP (if any). This information will be stored based on their respective relationships in each DNS packet flow, helping to remap malicious IPs from malicious domain name queries.

The extracted domain name queries are passed through the LA_Bin07 binary classifier to detect benign and malicious domains. Malicious domains are last reviewed with the most popular benign domains before concluding. Information about a malicious query is checked against the database to find the following fields: the IP address of the machine sending the malicious query and the IP address of the C&C server. From there, find malicious IP addresses on the internal network and on the Internet.

The LA_Bin07 model plays a crucial role in finding malicious IPs. This deep learning model was developed based on LSTM and Attention network architectures. The design of the LA_Bin07 model is shown in Figure 2 [11].

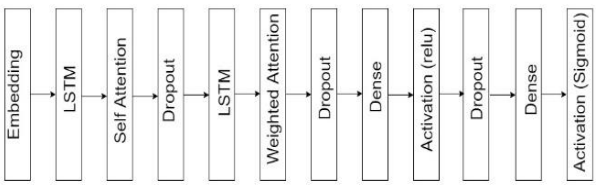

Figure 2. The architecture of the LA_Bin07 model

Model LA_Bin07 is designed in the form of seq-2-seq, including 12 layers, of which the main layers are as follows:

- LSTM layer: This layer is used to train the model. Some previous research results have shown the effectiveness of LSTM in detecting malicious domains;

- Attention layer: Added to the LSTM layer to help the model determine the essential parameters to learn, helping to improve the overall learning efficiency of the LA_Bin07 model.

- Dropout layer: Reduces the model's training parameters, reducing the computational load while maintaining a certain level of accuracy.

- Activation functions: ReLU and Sigmoid are used to produce the label to be classified.

Details of the parameters of the LA_Bin07 model are given in Table 1.

TABLE 1. PARAMETERS OF MODEL LA_BIN07

No	Class	Out size	Number of Paragrams
1	Embedding	(None, 100, 128)	640,000
2	Bidirectional LSTM	(None, 100, 256)	263.168
3	Seq Self Attention	(None, None, 256)	16,449
4	Dropout	(None, None, 256)	0
5	Bidirectional LSTM	(None, None, 256)	394.240
6	Seq Weighted Attention	(None, 256)	257
7	Dropout	(None, 256)	0
8	Dense	(None, 64)	16,448
9	Activation (ReLU)	(None, 64)	0
10	Dropout	(None, 64)	0
11	Dense	(None, 1)	65
12	Activation (Sigmoid)	(None, 1)	0
Total Paragrams			**1.330,627**

Specialized DGA botnet datasets previously trained the LA_Bin07 model. After training, the model will detect malicious domains extracted from real network datasets.

The next stage is to identify the DGA botnet families present in the network. The domains identified as malicious will be used for inclusion in the multiclass classifier LA_Mul07. The structure of the model LA_Mul07 (Figure 3) was proposed by us in the previous study [11].

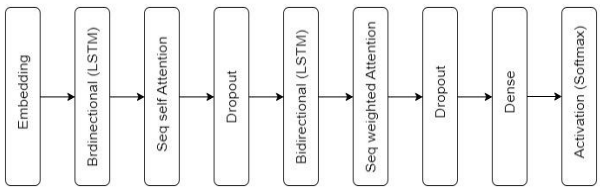

Figure 3. The architecture of the LA_Mul07 model

In it, the model includes classes similar to LA_Bin07 but simpler. This is suitable to optimize the training time of the model in case the number of labels to classify is large.

Details of the parameters of the LA_Mul07 model are given in Table 2 below.

TABLE 2. PARAMETERS OF MODEL LA_MUL07

No	Class	Out size	Number of Paragrams
1	Embedding	(None, 100, 128)	640,000 won
2	Bidirectional LSTM	(None, 100, 256)	263.168
3	Seq Self Attention	(None, None, 256)	16,449
4	Dropout	(None, None, 256)	0
5	Bidirectional LSTM	(None, None, 256)	394.240
6	Seq Weighted Attention	(None, 256)	257
7	Dropout	(None, 256)	0
8	Dense	(None, n)	P_n
9	Activation (Softmax)	(None, n)	0
Total Paragrams			**1,314,114+P_n**

Where, P_n is the number of classes to be classified, this value increases as the number of classes increases. This solution relies on the domain name to determine the family of the infected botnet in the network. This helps give suggestions for network managers to find and remove malicious code.

The number of parameters affects the speed and accuracy of the deep learning model. We attemp to design LA_Bin07 and LA_Mul07 models that balance training time and accuracy achieved. The evaluation results in the previous study showed that both models achieved high accuracy with sufficient training time.

IV. EXPERIMENTAL EVALUATION

In this part, we present the evaluations of the LA_Bin07 and LA_Mul07 models in two cases. In the first case, we evaluate a specialized DGA botnet dataset, UTL_DGA22, to assess the performance of the proposed model. In the second case, we evaluate an ISXC-Bot-2014 reality dataset. We evaluate the datasets with appropriate parameters depending on their features.

A. Evaluation on specialized datasets

1. UTL_DGA22 dataset

We built and published the UTL_DGA22 dataset in 2022 at [12]. Today, this can be considered the newest and most complete specialized dataset on DGA botnet, with 76 DGA botnet families synthesized and presented. To our knowledge, no research results on the DGA botnet have been evaluated using this dataset.

We used 1,000,000 benign domains and 76,000 DGA botnet domains, corresponding to 76 DGA bonet families, each family has 10,000 domains. Two deep learning models are run on Google Colab using GPU for a performance boost, RAM Premium, epochs = 10.

2. The evaluation result of the LA_Bin07 model

The evaluation result of the ability to detect the DGA botnet of the LA_Bin07 model on the UTL_DGA22 dataset is shown in Table 3.

TABLE 3. EVALUATION RESULT OF LA_BIN07 MODEL ON UTL_DGA22 DATASET

Label	Precision	Recall	F_1-Score
Benign	0.98	0.98	0.98
DGA Botnet	0.98	0.97	0.97
Accuracy	**0.98**		

It can be seen from the above table that the LA_Bin07 model has very high accuracy in detecting the domain name of the DGA botnet, with the accuracy reaching 0.98. The ability to identify labels is also uniform without deviating from either side, as shown by the precision and recall parameters that are very similar between the labels.

When evaluating the LA_Bin07 model, the confusion matrix and ROC curve are also shown in Figure 4 and 5, respectively, providing more information about the classification ability of the model.

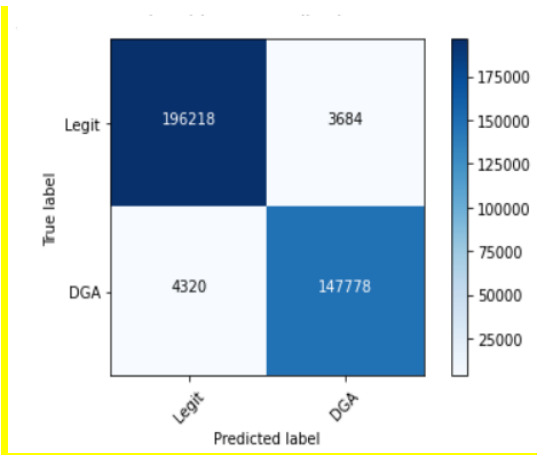

Figure 4. Confusion matrix of LA_Bin07 model when evaluated on UTL_DGA22 dataset

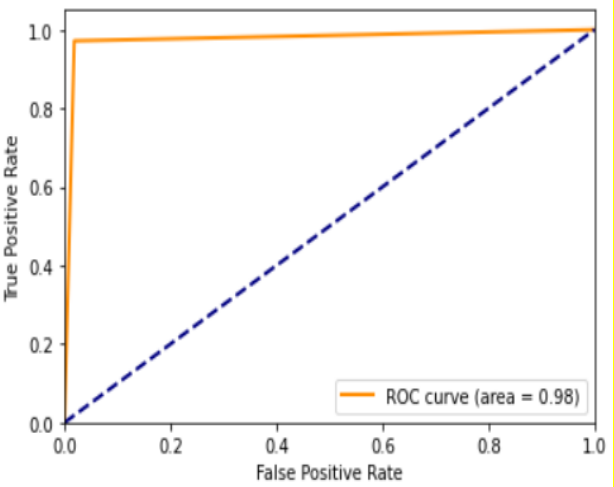

Figure 5. ROC Curve of LA_Bin07 model when evaluated on UTL_DGA22 dataset

Evaluating the LA_Bin07 model with basic machine learning algorithms when solving the problem of detecting DGA botnet, we have the results given in Table 4.

TABLE 4. COMPARISON WITH OTHER MACHINE LEARNING IN DGA BOTNET DETECTION

Model	**Pre.**	**Re.**	**F_1.**
Logistic Regression	0.97	0.95	0.96
Naive Bayes	0.91	0.86	0.86
Decision Tree	0.90	0.88	0.89
Neural Network	0.97	0.96	0.96
Support Vector Machine	0.97	0.94	0.96
Random Forrest	0.99	0.07	0.14
K-Nearest Neighbor	0.97	0.56	0.71
AdaBoost	0.84	0.79	0.81
LA_Bin07	*0.98*	*0.98*	*0.98*

Table 4 shows that the LA_Bin07 model is most effective in detecting the DGA botnet with an F1-score of 0.98. The Random Forest model was the least effective, with an F_1-score of 0.14. This model also has a huge difference in precision and recall. Other models with good results are logistic regression, neural networks, and support vector machines which both have an F_1-score of 0.96.

3. Evaluation results of the LA_Mul07 model

The evaluation result of the ability to detect the DGA botnet families of the LA_Mul07 model on the UTL_DGA22 dataset is shown in Table 5.

TABLE 5. EVALUATION RESULT OF LA_MUL07 MODEL ON UTL_DGA22 DATASET

No.	**DGA Botnet**	**Pre.**	**Re.**	**F_1.**
1	bamital	1.00	1.00	1.00
2	banjori	1.00	1.00	1.00
3	bazarbackdoor	1.00	1.00	1.00
4	bazarbackdoor_v2	1.00	1.00	1.00
5	bazarbackdoor_v3	1.00	1.00	1.00
6	bedep	0.71	0.67	0.69
7	bigviktor	0.97	0.96	0.97
8	ccleaner	1.00	1.00	1.00
9	chinad	1.00	1.00	1.00
10	corebot	1.00	1.00	1.00
11	cryptolocker	0.69	0.65	0.67
12	dircrypt	0.50	0.13	0.20
13	dnschanger	0.41	0.84	0.55
14	dyre	1.00	1.00	1.00
15	emotet	1.00	1.00	1.00
16	enviserv	1.00	1.00	1.00
17	fobber_v1	0.88	1.00	0.94
18	fobber_v2	0.44	0.17	0.25
19	gozi_gpl	0.97	0.99	0.98
20	gozi_luther	0.98	0.97	0.97
21	gozi_nasa	0.94	0.97	0.95
22	gozi_rfc4343	0.93	0.95	0.94
23	infy	1.00	1.00	1.00
24	kraken_v1	0.90	0.44	0.59
25	kraken_v2	0.58	0.66	0.62
26	locky	0.85	0.63	0.72
27	matsnu	0.96	0.98	0.97
28	monerodownloader	1.00	1.00	1.00
29	murofetweekly	0.50	0.70	0.59
30	murofet_v1	0.95	0.96	0.95
31	murofet_v2	0.74	0.84	0.79
32	murofet_v3	0.48	0.29	0.36
33	mydoom	1.00	1.00	1.00
34	necurs	0.99	0.80	0.89
35	newgoz	1.00	1.00	1.00
36	nymaim	0.50	0.84	0.63
37	nymaim2	0.99	0.94	0.96
38	oderoor	0.43	0.17	0.24
39	padcrypt	1.00	1.00	1.00
40	pandabanker	1.00	1.00	1.00
41	pitou	1.00	1.00	1.00

42	pizd	0.92	0.97	0.95
43	proslikefan	0.79	0.62	0.70
44	pushdo	1.00	0.99	1.00
45	pykspa_ improved_noise	0.44	0.19	0.26
46	pykspa_ improved_useful	0.36	0.37	0.37
47	pykspa_precursor	0.99	0.99	0.99
48	qadars	1.00	0.99	1.00
49	qakbot	0.83	0.48	0.61
50	qsnatch	1.00	1.00	1.00
51	ramdo	1.00	1.00	1.00
52	ramnit	0.37	0.56	0.44
53	ranbyus_v1	0.75	0.98	0.85
54	ranbyus_v2	0.83	0.87	0.85
55	reconyc	1.00	1.00	1.00
56	rovnix	0.98	0.95	0.96
57	shiotob	1.00	0.91	0.95
58	simda	1.00	1.00	1.00
59	sisron	1.00	1.00	1.00
60	sphinx	0.82	0.96	0.89
61	suppobox_1	0.97	0.92	0.94
62	suppobox_2	0.99	1.00	0.99
63	suppobox_3	1.00	1.00	1.00
64	symmi	1.00	1.00	1.00
65	szribi	0.99	1.00	0.99
66	tempedreve	0.59	0.75	0.66
67	tinba	0.73	0.98	0.84
68	tinynuke	1.00	1.00	1.00
69	torpig	0.98	1.00	0.99
70	vawtrak_v1	1.00	1.00	1.00
71	vawtrak_v2	1.00	1.00	1.00
72	vawtrak_v3	1.00	1.00	1.00
73	vidro	0.33	0.48	0.39
74	virut	0.90	1.00	0.95
75	wd	1.00	1.00	1.00
76	zloader	0.95	1.00	0.97
Accuracy		**0.86**		

The results show that the model LA_Mul07 has a high accuracy of 0.86 when evaluated on the new UTL_DGA22 dataset. There are 32 families of DGA botnets identified with almost absolutely correct labels with an F_1-score of 1.00. Some DGA botnet families have poor detection ability with a low F_1-score, such as dircrypt (0.20), fobber_v2 (0.25), murofet_v3 (0.36), oderoor (0.24), pykspa_improved_noise (0.26), pykspa_improved_useful (0.37), ramnit (0.44)), vidro (0.39). The remaining DGA botnet families generally give good classification results with high precision and recall and have a high degree of similarity.

B. Extension for malicious IP detection problem of DGA botnet

1. Description of the data

In this part, we extend the evaluation results to a non-specialized dataset of the DGA botnet. It is to be observed that, because of the non-specialty of the dataset (no corresponding label for the DGA botnet and the small number of samples, the huge data imbalance between malicious IPs and benign IPs), we consider the problem at the level of trying to find malicious IPs based on the indication of the DGA botnet.

We conduct model training on a combined dataset of 360NetLab and UMUDGA datasets. Then, we evaluate the ISXC-Bot-2014 dataset. Details of the datasets are described below:

- 360NetLab dataset [13]: A dataset of malicious domains collected from the real world, built by the 360NetLab research team, Qihoo 360 Technology Co., Ltd. The search engine will continuously find and detect the latest DGA botnet models to update the database.

- UMUDGA dataset [14]: Compiled and built by Zago and his colleagues from the University of Murcia. This is the most complete dataset available today, when 50 families of DGA botnets have been synthesized. Each family has between 10,000 and 500,000 domain name samples. The dataset is publicly available on the Mendeley Data website.

- ISCX-Bot-2014 dataset: A network dataset built by the Canadian Cybersecurity Institute, including 16 different types of botnets, including Neris, Rbot, Menti, Sogou, Murlo, Virus, NSIS, Zeus, SMTP Spam, UDP Storm, Tbot, Zero Access, Weasel, Smoke Bot, Zeus Control and ISCX IRC Bot [15]. The network traffic dataset is divided into two groups: TrainSet with a volume of 5.3 GB and TestSet with 8.5 GB. Each set includes malicious and typical network traffic.

The ISCX-Bot-2014 dataset is built based on the Overlay Method, which is the synthesis of 3 component datasets, including the ISOT Dataset, the ISCX 2012 IDS Dataset, and the Malware Capture Facility Project. Honeypot collects the botnet traffic. The IP addresses of the botnets are mapped to the external server addresses using the BitTwist packet generator. Finally, the above traffic is captured as pcap using TCPRelay and TCPdump.

Information about malicious IPs operating on the IRC channel includes [15]:

```
192.168.2.112 -> 131.202.243.84
192.168.5.122 -> 198.164.30.2
192.168.2.110 -> 192.168.5.122
192.168.4.118 -> 192.168.5.122
192.168.2.113 -> 192.168.5.122
192.168.1.103 -> 192.168.5.122
192.168.4.120 -> 192.168.5.122
192.168.2.112 -> 192.168.2.110
192.168.2.112 -> 192.168.4.120
192.168.2.112 -> 192.168.1.103
192.168.2.112 -> 192.168.2.113
192.168.2.112 -> 192.168.4.118
192.168.2.112 -> 192.168.2.109
192.168.2.112 -> 192.168.2.105
192.168.1.105 -> 192.168.5.122
```

The malicious IP addresses in the network are shown in Table 6 [15].

TABLE 6. LIST OF MALICIOUS IP ADDRESSES INFECTED WITH MALWARE

Botnet	IP	Botnet	IP
Neris	147.32.84.180	RBot	147.32.84.170
Menti	147.32.84.150	Sogou	147.32.84.140
Murlo	147.32.84.130	Virus	147.32.84.160
IRCbot and Black Hole 1	10.0.2.15	Black hole 2	192.168.106.141
Zeus	192.168.3.35, 192.168.3.25, 192.168.3.65, 172.29.0.116	TBot	172.16.253.130, 172.16.253.131, 172.16.253.129, 172.16.253.240
Zero access	172.16.253.132, 192.168.248.165	Weasel	74,78,117,238; 158.65.110.24
Black hole 3	192.168.106.131	Osx_trojan	172.29.0.109
Smoke bot	10.37.130.4		

To process large data sets like ISXC-Bot-2014, we use the SplitCap tool to filter DNS packets through port 53. The packets are then aggregated using the MergerCap tool. We extract information about the source and destination IP addresses, ports, query addresses, and response results using Python and supporting libraries. The data will be included after the noise removal steps to evaluate the model.

We used the Google Colab tool for the tests, configuring Intel Xeon Processors and 12 GB of RAM using the Tensorflow library in a Linux environment.

Usually, a dataset will be divided into two parts, called TrainSet and TestSet. That is, train and evaluate on the same data set. Our approach trains the model on a separate dataset and evaluates it on the ISCX-Bot-2014 dataset. This approach helps clarify the trained model's usability when applied in practice.

2. Detect malicious IP addresses and botnet malware

The evaluations for the binary classification problem on each dataset show that the LA_Bin07 model achieves an accuracy of 98.32% on the UMUDGA dataset, 99.39% on the 360NetLab dataset [11] and 97.73% on the UTL_DGA22 dataset. These results were obtained when training and evaluating the same data set. In our approach, we will use the above-trained model to evaluate the ISCX-Bot-2014 dataset. This represents a distinct difference between training and evaluation data.

Conduct experiments with two tasks, including: (1) Detecting malicious IPs operating on the IRC channel and (2) Detecting malicious IPs with signs of malicious code infection, recording the following results.

For the task of detecting malicious IPs operating on the IRC channel, fully detecting malicious IPs of 12/15 communications of malicious code via IRC protocol, the accuracy reaches 80%. IRC communications are not fully detectable, including:

```
192.168.2.112 -> 131.202.243.84
192.168.5.122 -> 198.164.30.2
192.168.2.112 -> 192.168.2.105
```

For the task of detecting potentially infected malicious IPs, shown in their behavior of sending malicious DNS queries to C&C servers, malicious IP addresses of 7/15 types of code were detected, as listed in Table 7.

TABLE 7. LIST OF DETECTED MALICIOUS IPS

Malicious code	IP	Malicious code	IP
Neris	147.32.84.180	Weasel	158.65.110.24
Black hole 3	192.168.106.131	Osx_trojan	172.29.0.109
Zero access	192.168.248,165	Virus	147.32.84.160
Smoke bot	10.37.130.4		

Note that the LA_Bin07 model can detect malicious IP addresses with 80% accuracy in detecting IRC communications and 47% accuracy in detecting infected IP addresses, respectively. The evaluation results on the UMUDGA dataset show that, with 1,500,000 DNS records, the training and evaluation times are 4424s and 100s, respectively. The 360NetLab dataset contains more than 2,500,000 records. The time is 6883s and 204s, respectively. Evaluation on UTL_DGA22 dataset with 1,760,000 records has a training time of 6861s and an evaluation time of 120s. The LA_Bin07 solution can also operate using conventional CPUs without needing high-performance packet processors.

The above results illustrate that the DGA botnet detection solution can be applied to the malicious IP detection problem, providing good suggestions for network administrators in searching IP addresses with dedications to the botnet. Because of the limitations of data labels, the number of data samples, and imbalances, we do not go into detailed parameters.

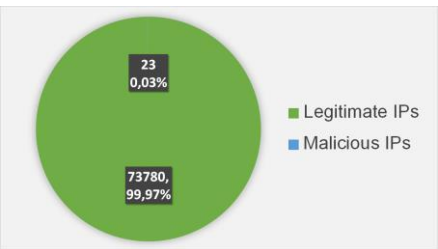

Figure 6. Correlation between normal and malicious IP numbers in the dataset

Figure 6 shows the percentage of malicious IPs and benign IP addresses in the dataset, representing the imbalance presented.

V. CONCLUSION AND DEVELOPMENT DIRECTION

In this study, we presented new evaluations of using LA_Bin07 and LA_Mul07 deep learning models on the new UTL_DGA22 dataset to solve the DGA botnet problem. The evaluation results show that both proposed models have high accuracy, especially in the DGA botnet detection problem. A comparison with machine learning algorithms shows that the new model improves accuracy significantly while having a much faster execution time. We also conducted an extended application of the above model to detect malicious IP addresses, and the results were positive.

Regarding the research direction, we continue to improve the DGA botnet detection and classification models based on deep learning, focusing mainly on the DGA botnet families' classification. We also make improvements and conduct new experiments on similar reality datasets to collect more complete and comprehensive data. Finally, the proposed solution can be applied as a malicious IP address and domain name detection module, installed on security solutions such as firewalls, intrusion detection systems, and unified threat management - UTM.

ACKNOWLEDGMENT

Tong Anh Tuan was funded by Vingroup JSC and supported by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, code VINIF.2021.TS.050.

This research is funded by the National Science and Technology Major Project of the Ministry of Science and Technology of Vietnam under grant number ĐTĐLCN.105/21-C.

REFERENCE

[1] R. Vishwakarma and A. K. Jain, "A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks," *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, pp. 1019–1024, 2019, doi: 10.1109/ICOEI.2019.8862720.

[2] D. T. Son, N. T. K. Tram, and T. T. Thu, "Machine learning approach detects DDoS attacks," *J. Sci. Technol. Inf. Secur.*, vol. 1, no. 15, pp. 102–108, 2022, doi: 10.54654/isj.v1i15.850.

[3] P. Wang, L. Wu, R. Cunningham, and C. C. Zou, "Honeypot detection in advanced botnet attacks," *Int. J. Inf. Comput. Secur.*, vol. 4, no. 1, pp. 30–51, 2010, doi: 10.1504/IJICS.2010.031858.

[4] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing botnet membership using DNSBL counter-intelligence," *2nd Work. Steps to Reducing Unwanted Traffic Internet, SRUTI 2006*, pp. 49–54, 2006.

[5] N. Kheir, F. Tran, P. Caron, and N. Deschamps, "Mentor: Positive DNS reputation to skim-off benign domains in botnet C&C blacklists," *IFIP Adv. Inf. Commun. Technol.*, vol. 428, 2014, doi: 10.1007/978-3-642-55415-5_1.

[6] J. Wang and I. C. Paschalidis, "Botnet Detection Based on Anomaly and Community Detection," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 392–404, 2017, doi: 10.1109/TCNS.2016.2532804.

[7] S. Arshad, M. Abbaspour, M. Kharrazi, and H. Sanatkar, "An anomaly-based botnet detection approach for identifying stealthy botnets," *ICCAIE 2011 - 2011 IEEE Conf. Comput. Appl. Ind. Electron.*, pp. 564–569, 2011, doi: 10.1109/ICCAIE.2011.6162198.

[8] K. Wang, C. Y. Huang, S. J. Lin, and Y. D. Lin, "A fuzzy pattern-based filtering algorithm for botnet detection," *Comput. Networks*, vol. 55, no. 15, pp. 3275–3286, 2011, doi: 10.1016/j.comnet.2011.05.026.

[9] Z. Ahmed, S. M. Danish, H. K. Qureshi, and M. Lestas, "Protecting IoTs from mirai botnet attacks using blockchains," *IEEE Int. Work. Comput. Aided Model. Des. Commun. Links Networks, CAMAD*, vol. 2019-Septe, 2019, doi: 10.1109/CAMAD.2019.8858484.

[10] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "EXPOSURE: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, 2014, doi: 10.1145/2584679.

[11] T. A. Tuan, H. V. Long, and D. Taniar, "On Detecting and Classifying DGA Botnets and their Families," *Comput. Secur.*, vol. 113, 2022, doi: 10.1016/j.cose.2021.102549.

[12] T. A. Tuan, N. V. Anh, T. T. Luong, and H. V. Long, "UTL_DGA22 - a dataset for DGA botnet detection and classification," *Comput. Networks*, vol. 221, 2023, doi: 10.1016/j.comnet.2022.109508.

[13] 360NetLab, "DGA - Netlab OpenData Project," *Qihoo 360 Technology*, 2022. https://data.netlab.360.com/dga/ (accessed Mar. 09, 2021).

[14] M. Zago, M. Gil Pérez, and G. Martínez Pérez, "UMUDGA: A dataset for profiling DGA-based botnet," *Comput. Secur.*, vol. 92, 2020, doi: 10.1016/j.cose.2020.101719.

[15] D. Gonzalez-Cuautle *et al.*, "Synthetic minority oversampling technique for optimizing classification tasks in botnet and intrusion-detection-system datasets," *Appl. Sci.*, vol. 10, no. 3, 2020, doi: 10.3390/app10030794.

ABOUT THE AUTHORS

Tong Anh Tuan

Workplace: Faculty of Information Technology, University of Technology - Logistics of Public Security.

Email: tuanqb92@gmail.com

Education: PhD student at Graduate University of Science and Technology, Vietnam Academy of Science and Technology.

Recent research interests: Deep learning; Network attack and Defense.

Nguyen Ngoc Cuong

Workplace: University of Technology - Logistics of Public Security.

Email: cuongnnhvan@gmail.com

Education: Obtained a PhD degree from University of Science - Vietnam National University.

Recent research interests: Information security; Artificial Intelligence.

Nguyen Viet Anh

Workplace: Institute of Information Technology, Vietnam Academy of Science and Technology.

Email: nvanguyen@gmail.com

Education: Obtained a PhD degree in Computer Science from Kyoto University, Japan.

Recent research interests: Machine learning; Big data and Social network analysis.

Hoang Viet Long

Workplace: Faculty of Information Technology, University of Technology-Logistics of Public Security.

Email: longhv08@gmail.com

Education: Received PhD diploma in Computer Science at Hanoi University of Science and Technology in 2011; He has been promoted to Associate Professor since 2018.

Recent research interests: Machine learning with applications to Cybersecurity.