A Novel Machine Learning-based Approach for Detecting Word-based DGA Botnets

Article *in* Journal of Theoretical and Applied Information Technology · December 2021

Vũ Xuân Hạnh
Hanoi Open University

Dau Hoang
Posts and Telecommunications Institute of Technology

A NOVEL MACHINE LEARNING-BASED APPROACH FOR DETECTING WORD-BASED DGA BOTNETS

[1]XUAN HANH VU, [2]XUAN DAU HOANG

[1]Faculty of Information Technology, Hanoi Open University, Hanoi, Vietnam

[2]Cybersecurity Lab, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

E-mail: [1]hanhvx@hou.edu.vn, [2]dauhx@ptit.edu.vn

ABSTRACT

Recently, Domain Generation Algorithm (DGA) has been becoming a popular technique used by many malwares in general and a large number of botnets in particular. DGA allows botnet owners to automatically generate and register domain names for their Command and Control (C&C) servers to avoid being blacklisted and blocked. Botnets and especially DGA botnets are associated with many types of dangerous attacks, such as large-scale DDoS attacks, email spamming and APT attacks. Due to the wide-spreading and serious consequences of DGA botnets, several approaches based on statistics and machine learning techniques to detect DGA botnets have been proposed. Although some machine learning-based approaches achieve high overall accuracy in detecting general or character-based DGA botnets, they fail to detect some kinds of DGA botnets, including word-based or dictionary-based botnets. These botnets usually use pre-defined English word lists to generate meaningful domain names for their C&C servers, which look almost similar to legitimate domain names. This paper proposes a novel machine learning based approach for effectively detecting word-based DGA botnets. The proposed approach introduces a new set of 16 features extracted for each domain name for training and detecting word-based DGA botnets. Extensive experiments on the word-based DGA dataset and the mixed DGA dataset confirm that our approach achieves the F1-score of 97.01% and 95.75% for the word-based and mixed DGA datasets, respectively.

Keywords: *Word-based DGA botnet, Character-based DGA botnet, DGA botnet detection, Word-based DGA botnet detection*

1. INTRODUCTION

Over last ten years, botnets have been seen one of the prime security threats to Internet-based services, Internet-connected devices and individual Internet users [1][2][3]. This is due to botnets have been linked with many kinds of Internet-based misuses and attacks, including DDoS attacks, malware transmitting, email spamming and stealing of sensitive information [4]. For example, according to a report by Symantec, about 95% of spam emails in the Internet in 2010 was created and sent by botnets [1]. In 2019, a large-scale DDoS attack overran the Telegram newspaper's network, which was said to be originated from China and related to the protests in Hong Kong in the same year [3][4]. Moreover, other dangerous kinds of attacks and misuses assisted by botnets are web injection attacks, URL spoofing and DNS spoofing [1][2].

Conventionally, a *botnet* is a network of *bots*. A bot is a special kind of malware running on an Internet-connected device that can be a computer, a smartphone or an IoT device [4][5][6]. Bots are often created and maintained by hacking groups, called *botmasters*. When a bot is infected and running on a device, it allows the botmaster to remotely control the device. A botmaster usually uses a control system, called the Command and Control (C&C or CnC) server to control and maintain a botnet [4][5][6]. On one side, the botmaster sends commands and code updates to his botnet's C&C server. On the other side, bots in his botnet are equipped with the capability of using communication channels to connect to and receive commands and code updates from the C&C server. Bots can also send their working status to their C&C server.

Bots in a botnet periodically send DNS queries containing the botnet's C&C server name to the local DNS service to find the server's IP address in order to connect to the C&C server. To avoid the C&C server from being blacklisted and then blocked if using a static name and IP address, the botmaster usually uses special techniques, such as *Fast Flux* (FF), or *Domain Generation Algorithm* (DGA) to dynamically generate and register domain names for his botnet's C&C server [4][5][6]. Bots in the botnet are also equipped with the capability to dynamically generate server's names using the same DGA technique. Therefore, bots can still find the IP address of the C&C server to connect to by sending queries of server names to the local DNS service, as illustrated in Figure 1. Since DGA technique has been very popular and then botnets that utilize DGA technique are called *DGA botnets*.

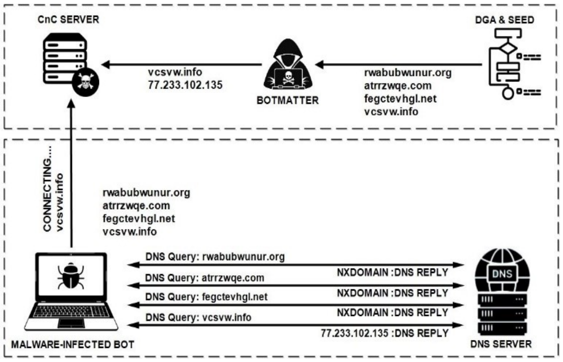

Figure 1. Example of a botnet using DGA techniques to generate, register and query C&C server names.

In general, DGA botnets can be detected by monitoring and analyzing DNS queries sent by bots and legitimate applications [4][5][6]. Domain names extracted from DNS queries are then processed and classified to find domain names generated and used by DGA botnets. Because domain names generated by DGA botnets using mathematical algorithms usually have no meaning and are generally different from legitimate domain names in the order of characters, approaches based on statistics and machine learning can be used to classify DGA and legitimate domain names. Recent machine learning-based proposals for DGA botnet detection, such as those in [4], [7] and [8] achieved relatively high detection accuracy and fairly low false alarm rate. However, these proposals only work well on the domain names generated by character-based DGA botnets. They are unable to detect new families of DGA botnets that are named word-based or dictionary-based DGA botnets [4]. The word-based DGA botnets usually use pre-defined lists of English words to generate meaningful domain names for their C&C servers, which are highly similar to legitimate domain names. The word-based DGA botnets will be discussed in details in Section II.

This paper proposes a novel model based on supervised machine learning techniques for effectively detecting word-based DGA botnets. We use the original model proposed in [4] and introduce a new set of 16 features for classification of legitimate domain names and word-based DGA botnet domain names. Traditional supervised machine learning techniques, including Naïve Bayes, decision tree, random forest, logistic regression and Support Vector Machine (SVM) are used to construct the detection model because they are fast and proven to produce good performance for text classification problems [9][10].

2. RELATED WORKS

3.1 Overview of Word-based DGA Botnets

DGA techniques can be divided into 3 categories of character-based, word-based and mixed DGA methods. Character-based DGA techniques usually use mathematical algorithms to generate domain names for C&C servers. Therefore, character-based DGA domain names are strings that have characters in random order. DGA botnets, such as *cryptolocker*, *emotet* and *feodo* are typical botnets that use character-based DGA techniques to generate domain names. On the other hand, word-based DGA techniques usually use pre-complied lists of English nouns, verb and adjectives to generate domain names. Therefore, word-based DGA domain names have some meanings and their characters are in correct order. DGA botnets, such as *bigviktor, matsnu* and *suppobox* are typical botnets that use word-based DGA techniques to generate domain names. Mixed DGA techniques are a combination of character-based and word-based DGA techniques, in which one part of a domain name is generated using the character-based technique and the other part of the domain name is generated using the word-based technique. *Banjori* botnet is a typical botnet that uses mixed DGA techniques to generate domain names. Table 1 shows some samples of DGA botnet domain names generated using different DGA techniques.

Table 1. Samples of DGA Botnet Domain Names Generated Using Different DGA Techniques

Family of Botnets	DGA Technique	Domain Name Samples
crypto-locker	character-based	ryojulmtdxljnkn.biz icfpkabnmsse.org kynkbkflfrlqcx.biz
emotet	character-based	affvqugewqpbcbic.eu amxecvgvhfequgpo.eu atqanjgnftfsnywb.eu teswpukmvttjigbj.eu
feodo	character-based	hmvmgywkvayilcwh.ru xvmzegestulhtvqz.ru

		hjpyvexsutdctjol.ru
bigviktor	word-based	knowredpermit.art winstilllandscape.club helppurpledistance.fans
matsnu	word-based	row-closed-bid.com brushpot-guide.com sort-address.com
suppobox	word-based	necessarypower.net pleasantcountry.net necessarycountry.net
banjori	mixed	ztgxrasildeafeninguvuc.com vdrgrasildeafeninguvuc.com umdhrasildeafeninguvuc.com

Mixed and especially word-based DGA botnets are more difficult to detect than character-based DGA botnets because their generated domain names are a combination of meaningful words. Therefore their generated domain names look highly similar to legitimate domain names. For example, *suppobox* botnet combines an adjective and a noun to create a domain name, such as *necessarypower*.net, *pleasantcountry*.net and *necessarycountry*.net. Therefore, proposed machine learning-based DGA botnet detection approaches, such as those in [4], [7] and [8] fail to detect word-based DGA botnets even though they are capable of detecting character-based DGA botnets effectively.

3.2 Review of DGA Botnet Detection Proposals

As mentioned in Section I, botnets can be effectively detected by monitoring and analyzing DNS queries sent from legitimate applications and bots to find the botnets' activities in the local networks. In this direction, there have been several botnet detection proposals, such as Jiang et al. [11], Stalmans et al. [12], Antonakakis et al. [13], Bilge et al. [14], Yadav et al. [15], Kheir et al. [16], Woodbridge et al. [17], Truong et al. [8], Hoang et al. [7], Qiao et al. [18], Zhao et al. [19], Yang et al. [20], Charan et al. [21], Ren et al. [22], Satos et al. [23], and Hoang et al. [4]. This section provides a deep review of recent and closely related DGA detection proposals based on statistics and machine learning techniques, including Truong et al. [8], Hoang et al. [7], Qiao et al. [18], Zhao et al. [19], Charan et al. [21], and Hoang et al. [4].

Truong et al. [8] proposes a framework to detect domain-flux botnets using DNS traffic features, as shown in Figure 2. They use DNS domain features, such as the length and expected value of domain names to distinguish between legitimate and pseudo-random domain names (PDN) generated by botnets. The domain name expected value is computed based on the character distribution of 100,000 most popular legitimate domain names ranked by Alexa [24]. The experimental dataset includes 100,000 most popular legitimate domain names ranked by Alexa [24] and about 20,000 PDN domain names generated by Zeus and Conficker botnets. Five supervised machine learning algorithms, including Naive Bayes, kNN, SVM, decision tree and random forest have been used to build and validate the proposed detection framework. Experimental results confirm that the decision tree algorithm gives the highest overall detection accuracy of 92.30% and the false positive rate of 4.80%. Although the proposed framework's overall detection accuracy is relatively high, its false alarm rates in total are also high, at about 7.70% in the best case.

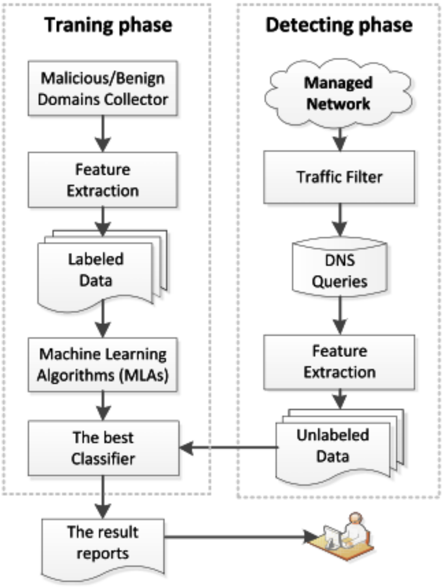

Figure 2. The botnet detection framework proposed by Truong et al. [8]

Using a similar approach, Hoang et al. [7] proposes a DGA botnet detection model based on the classification of legitimate and botnet generated domain names using supervised machine learning techniques, as illustrated in Figure 3. They propose to use 18 features of domain names, including 16 n-gram features and 2 vowel distribution features to build and validate the proposed model. Among 16 n-gram features, 8 features are computed based on each domain's 2-gram substrings and other 8 features are calculated based on the domain's 3-gram substrings. The dataset of 30,000 top legitimate domain names ranked by Alexa [24] and 30,000 malicious domain names used by DGA botnets [25] are used for experiments. Traditional supervised machine learning algorithms, including Naive Bayes, kNN, decision tree and random forest have been used to build and validate the proposed model. Various experiments have been conducted using different testing scenarios. The experimental results show that machine learning techniques can be effectively used to detect DGA botnets based on the classification of legitimate and algorithm-generated botnet domain names. The experimental results also show that the random forest algorithm produces the highest overall detection rate of over 90%. However, the main issues of the proposed model are (1) the false positive rate is pretty high at 9.30% and (2) the experimental dataset is relatively small for each testing scenario compared to other approaches. A high false positive rate will limit the proposed model's applicability in practice. On the other hand, a small dataset for experiments will reduce the reliability of results.

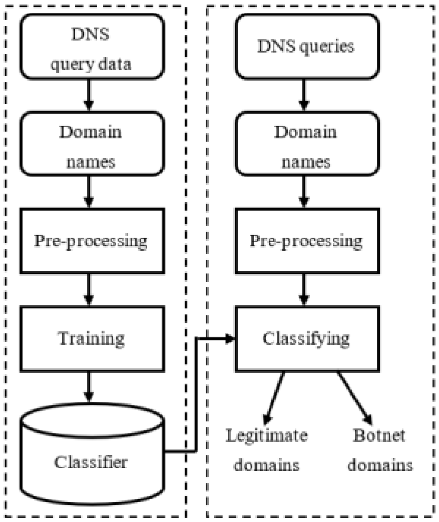

Figure 3. The botnet detection model proposed by Hoang et al. [7]

Qiao et al. [18] proposes a classification method for DGA domain names based on Long Short-Term Memory (LSTM) with attention mechanism. In the proposed method, each domain name is gone through pre-processed steps of DGA string extraction, padding and embedding. The domain name is then transformed into 54×128 matrix for training and testing. The experimental dataset of top one million legitimate domain names ranked by Alexa [24] and 1,675,404 malicious domain names generated by various DGA botnets [25]. Experimental results show that the proposed method performs better than current state-of-the-art methods with the average F1-score of 94.58%. Using the LSTM learning method, the proposed model can remove the feature extraction process. However, the paper does not provide any information about the method complexity, nor the requirement of computational resources. In addition, the false alarm rates are relatively high at about 5%, which can be computed from the precision and recall of both about 95%.

Using another approach, Zhao et al. [19] proposes a statistical method to detect malicious domain names using n-gram technique. Each domain name in the training set of legitimate domains is first divided into sequences of substrings using 3, 4, 5, 6 and 7-gram technique. Then, the statistics and weight values of substrings of all training domains are calculated to build the 'profile'. To validate an input domain name if it is legitimate or malicious, the domain name is also first divided into sequences of substrings using 3, 4, 5, 6 and 7-gram technique. Then, the statistics of domain name substrings are calculated and then it is used to compute the 'reputation value' of the domain name based on the 'profile'. A domain reputation threshold is generated for each category of malicious domain names using the 'profile'. If the domain name's reputation value is greater than the threshold, it is legitimate. Otherwise, it is malicious. Experimental results show that the proposed approach achieves the detection accuracy of 94.04%. However, the detection performance of the proposed approach heavily depends on the selection of the domain reputation threshold that is currently generated and selected manually. Furthermore, its false positive and negative rates are relatively high at 6.14% and 7.42%, respectively.

Charan et al. [21] proposes a new method to detect word-list based DGA domain names using ensemble learning algorithms. They propose to use 15 lexical and network-level domain name features to construct and validate the detection method. Several supervised machine learning techniques, including C4.5, C5.0, CART decision tree and random forest are used in ensemble models to enhance the detection performance. Experimental results confirm that the C5.0 decision tree is the best algorithm with the prediction accuracy of 95.03%. The only issue with the the proposed approach is it does not give the details on how are the ensemble models work to generate the results from individual models.

Most recently, Hoang et al. [4] proposed an enhanced model for detecting DGA botnets using random forest algorithm. The new DGA botnet detection model is an extension of their previous work [7] aiming at increasing the detection rate and lowering the false alarm rate. They propose a new set of 24 domain name features to construct and validate the proposed model. Extensive experiments on the dataset of 100,000 legitimate domain names [24] and 153,000 DGA botnet domain names [25] show that the proposed model achieves high detection accuracy of 97.03% and low false alarm rate of about 3%. In addition, the model is able to detect most of DGA botnets in the experimental dataset with the average detection rate of over 80%. However, the biggest issue with the proposed model is it fails to detect word-based and mixed DGA botnets, such as *banjori, matsnu, bigviktor* and *suppobox*.

Table 2 gives a general comparison of previous proposals in the following properties: the detection accuracy (ACC), the F1-score, the advantages and the disadvantages.

Table 2. General Comparison of Previous Proposals

Approaches	ACC	F1	Advantages	Disadvantages
Truong et al. [8], 2016	92.30		Simple and fast	- High false alarm rate (about 7.70%) - Cannot detect word-based DGA botnets.
Hoang et al. [7], 2018	90.90	90.90	Relatively simple and fast	- Small dataset - High false alarm rate (about 7.70%) - Cannot detect word-based DGA botnets.
Qiao et al. [18], 2019		94.58	High accuracy	- Requires extensive computing resources - High false alarm rate (about 5%) - Cannot detect word-based DGA botnets.
Zhao et al. [19], 2019	94.04		High accuracy	- Difficult to select detection threshold - High false negative rate (7.42%) - Cannot detect word-based DGA botnets.
Charan et al. (C5.0) [21], 2020	95.03		- High accuracy - Can detect word-based DGA botnets.	- The ensemble models are not clearly presented.
Hoang et al. [4], 2021	97.03	97.03	- High accuracy - Low false alarm rate.	- Cannot detect word-based DGA botnets.

3. THE PROPOSED WORD-BASED DGA DETECTION MODEL

3.1 The Word-based DGA Detection Model

We use the DGA botnet detection model proposed in [4] for detecting word-based DGA botnets. Figure 4 shows the word-based DGA detection model that consists of two phases: (a) the training phase and (b) the detection phase. In the training phase, the model is built from the training data. Then the built model is used to classify each test domain name if it is a legitimate or botnet domain name in the detection phase.

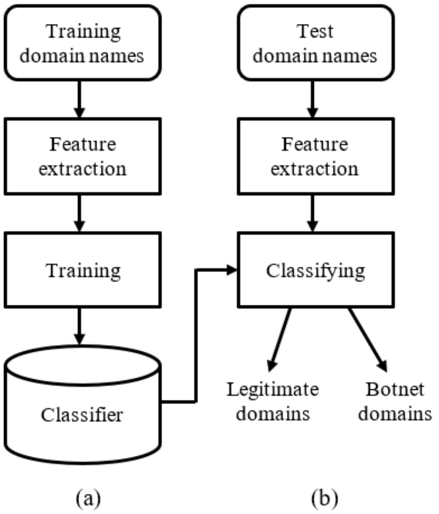

Figure 4. The word-based DGA detection model: (a) the training phase and (b) the detection phase.

The training phase as described in Figure 4(a) has two steps as the following:

- Feature extraction: The training dataset is passed through the process of feature extraction, in which 16 classification features are extracted for each domain name. Each domain name is transformed to a vector of 16 features and a class label. The class label has 2 values of 0 for legitimate and 1 for botnet. The result of the extraction of features is a training data matrix of M rows and 17 columns, in which each row of the matrix is a vector of a domain name;
- Training: In this step, the training data matrix is used to construct the detection model or the 'Classifier' using traditional supervised machine learning techniques. Several supervised learning techniques, such as Naïve Bayes, decision tree, random forest, logistic regression and Support Vector Machine (SVM) have been used to construct detection models. These learning methods are selected because they are fast and hence suitable for processing large amount of data in online mode. In addition, they have been used widely in many areas and achieved good performance [9][10]. The constructed model is also validated using the 10-fold cross-validation method to get the model's performance measurements.

The training phase as illustrated in Figure 4(b) also has two steps as the following:

- Feature extraction: Each test domain name is processed using the same procedure as done in the training phase in this step. Each test domain name is transformed to a vector of 17 features;
- Classifying: In this step, the vector of each test domain name is classified using the 'Classifier' built in the training phase. The step's result is the test domain name's predicted label of either legitimate or botnet.

3.2 Feature Extraction

As mentioned in Section 3.1, 16 classification features are extracted for each domain name in both the training and detection phases. These features are named as *f1*, *f2*, *f3*,…, *f16*. Among them, *f1*, *f3*, *f4, f5, f6* features are defined in [4][7][8]. Other features are proposed in this paper. We used and created the following English dictionaries to assist the extraction of features as follows:

- An English dictionary containing 58,000 words [26] is used for normalizing words used in domain names to the standard forms. This dictionary is named as '*english_dict*';
- Lists of common English nouns, verbs and adjectives are constructed using the frequently used words listed in [27]. The reason for creating these lists of words is some botnets, such as *bigviktor* and *matsnu* use lists of common nouns, verbs and adjectives to generate domain names for their C&C servers. The noun, verb and adjective lists are named as '*noun_dict*', '*verb_dict*' and '*adj_dict*', respectively;
- A word-based DGA dictionary called '*dga_dict*' is built. This dictionary consists of words that are used by word-based DGA botnets and their original word forms are in '*english_dict*';
- An additional word-based DGA dictionary called '*private_dict*' is built. This dictionary consists of words that are used by word-based DGA botnets and their original word forms are not in '*english_dict*'.

The explanation of each classification feature of the domain name *d* is as follows:

f1: the length of the domain name *d* in characters, represented as *len(d)* [8]*;*

f2: the total value of ASCII code of all characters in the domain name *d*, which is computed by the following formula:

$$ascii\_value(d) = \sum_{i=1}^{len(d)} ord(d[i]) \qquad (1)$$

f3: the numer of vowels of the domain name *d*, denoted as *countnv(d)* [7];

f4: the vowel distribution of the domain name *d*, which is computed by the following formula [4][7]:

$$tanv(d) = \frac{countnv(d)}{len(d)} \qquad (2)$$

f5: the numer of digits and character '-' of the domain name *d*, denoted as *countdi(d)* [4];

f6: digit and character '-' distribution of domain name *d*, which is computed by the following formula [4]:

$$tandi(d) = \frac{countdi(d)}{len(d)} \qquad (3)$$

f7: the number of words that are extracted from domain name *d* and exist in *'english_dict'* dictionary. This feature is denoted as *word_norm(d)*;

f8: the number of words that are extracted from the domain name *d* and exist in *'dga_dict'* dictionary. This feature is denoted as *word_dga(d)*;

f9: the number of words that are extracted from the domain name *d* and exist in *'noun_dict'* dictionary. This feature is denoted as *noun_count(d)*;

f10: the number of words that are extracted from the domain name *d* and exist in *'verb_dict'* dictionary. This feature is denoted as *verb_count(d)*;

f11: the number of words that are extracted from the domain name *d* and exist in *'adj_dict'* dictionary. This feature is denoted as *adj_count(d)*;

f12: the number of words that are extracted from the domain name *d* and exist in *'private_dict'* dictionary. This feature is denoted as *private_count(d)*;

f13: the ratio between *word_dga(d)* and *word_norm(d)*, which is computed by the following formula:

$$ratio\_dga(d) = \frac{word\_dga(d)}{word\_norm(d)} \qquad (4)$$

f14: the length of the longest word of the domain name *d*. This feature is denoted as *max_len_word(d)*;

f15: the length of the shortest word of the domain name *d*. This feature is denoted as *min_len_word(d)*;

f16: the ratio between the number of characters of words of the domain name *d* and the length of the domain name *d*, which is computed by the following formula:

$$ratio\_char(d) = \frac{len(words(d))}{len(d)} \quad (5)$$

3.3 Classification Measurements

Six standard measurements are used to measure the detection performance of the proposed botnet detection model. The measurements include PPV, TPR, FPR, FNR, F1 and ACC. PPV is Positive Predictive Value, or Precision; TPR is True Positive Rate, or Recall; FPR is False Positive Rate; FNR is False Negative Rate; F1 is the F1-score; and ACC is the overall accuracy. These standard measurements are computed using the following formulas [4]:

$$PPV = \frac{TP}{TP + FP} \quad (6)$$

$$TPR \frac{TP}{TP + FN} \quad (7)$$

$$FPR = \frac{FP}{FP + TN} \quad (8)$$

$$FNR = \frac{FN}{FN + TP} \quad (9)$$

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (10)$$

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \quad (11)$$

where TP, FP, FN and TN are elements of the confusion matrix given in Table 3.

Table 3: TP, FP, FN And TN in the Confusion Matrix

		Actual Class	
		Attacked	*Normal*
Predicted Class	*Attacked*	TP (True Positives)	FP (False Positives)
	Normal	FN (False Negatives)	TN (True Negatives)

Furthermore, the Detection Rate (DR) is used to measure the detection models' effectiveness for classifying domain names generated by various botnets. The DR for each botnet type is computed as follows:

$$DR = \frac{Number\ of\ detected\ DGA\ domains}{Total\ number\ of\ DGA\ test\ domains} \quad (12)$$

4. EXPERIMENTS AND RESULTS

4.1 Experimental Dataset

The dataset used for experiments consists of three subsets as the following:

- The subset of 48,000 legitimate domain names extracted from Top Alexa one million domains [24];
- The subset of 64,000 word-based DGA domain names generated using DGA scripts [28] for 4 typical word-based DGA botnets of *bigviktor, matsnu, suppobox* and *pizd*. 48,000 domain names of this subset are used for training and validating the detection models and 16,000 domain names are used for testing;
- The subset of 63,905 DGA domain names generated by 16 DGA botnets. These domain names are collected from Netlab360 [25]. 48,000 domain names of this subset are used for training and validating the detection models and 15,905 domain names are used for testing.

From the 3 data subsets, we create 2 datasets for our experiments as follows:

- Dataset-01 is used for validating the word-based DGA botnet detection capability of the proposed model. The dataset is comprised of (1) a training part of 48,000 legitimate domain names and 48,000 word-based DGA domain names, and (2) a testing part of 16,000 word-based DGA domain names. Table 4 shows the detailed components of Dataset-01;
- Dataset-02 is used for validating the DGA botnet detection capability of the proposed model. The dataset is comprised of (1) a training part of 48,000 legitimate domain names and 48,000 DGA domain names, and (2) a testing part of 15,905 DGA domain names. The DGA domain names are generated by both word-based and character-based DGA botnets. Table 5 presents the detailed components of Dataset-02.

Table 4. The Components of Dataset-01

Family of Domain Names	Domain Type	Training Part	Testing Part
Bigviktor	word-based	12,000	4,000
Matsnu	word-based	12,000	4,000
Suppobox	word-based	12,000	4,000
Pizd	word-based	12,000	4,000
Benign	legitimate	48,000	
Total		**96,000**	**16,000**

Table 5. The Components of Dataset-02

Family of Domain Names	Domain Type	Training Part	Testing Part
Bigviktor	word-based	3,000	1,000
Matsnu	word-based	3,000	905
Suppobox	word-based	3,000	1,000
Pizd	word-based	3,000	1,000
Flubot	char-based	3,000	1,000
Necurs	character-based	3,000	1,000
Ramnit	character-based	3,000	1,000
Ranbyus	character-based	3,000	1,000
Rovnix	character-based	3,000	1,000
Tinba	character-based	3,000	1,000
Cryptolocker	character-based	3,000	1,000
Dyre	character-based	3,000	1,000
Emotet	character-based	3,000	1,000
Gameover	character-based	3,000	1,000
Murofet	character-based	3,000	1,000
Shiotob	character-based	3,000	1,000
Benign	legitimate	48,000	
Total		**96,000**	**15,905**

4.2 Experimental Scenarios and Results

Our experiments are implemented using the following scenarios:

- Scenario-1: training and validating the detection model using the 'training part' of the Dataset-01. Five supervised machine learning techniques, including Naïve Bayes (NB), decision tree, random forest, logistic regression (Logistic) and SVM are used in sequence to construct the detection models, in which 80% of the 'training part' is used for training to build the models and 20% of the 'training part' is used for validating the models to get the models' performance measurements. The J48 tree is used for the decision tree and the random forest is used with 35 trees (RF-35);
- Scenario-2: testing the detection models built in Scenario-1 using the 'testing part' of the Dataset-01. The purpose of this scenario is to find the detection rate (DR) of the built models on some typical word-based DGA botnets;
- Scenario-3: training and validating the detection model using the 'training part' of the Dataset-02. NB, J48 tree, RF-35, Logistic and SVM algorithms are used in sequence to construct the detection models, in which 80% of the 'training part' is used for training to build the models and 20% of the 'training part' is used for validating the models to get the models' performance measurements;
- Scenario-4: testing the detection models built in Scenario-3 using the 'testing part' of the Dataset-02. The purpose of this scenario is to find the detection rate (DR) of the built models on typical DGA botnets of both word-based and character-based DGA botnets.

Table 6 presents the detection performance of the proposed model based on 5 learning algorithms using 'training part' of the Dataset-01. The performance measurements on this table confirm that the proposed model performs very well on Dataset-01 with word-based DGA botnets using all 5 learning algorithms. The built model from the 'training part' of the Dataset-01 also gives high detection rate for all 4 word-based DGA botnets, as shown in Table 7.

Table 8 shows the detection performance of the proposed model based on 5 learning algorithms using 'training part' of the Dataset-02. The performance measurements on this table also confirm that the proposed model performs well on Dataset-02 with both word-based and character-based DGA botnets using all 5 learning algorithms. The constructed model from the 'training part' of the Dataset-02 also produces good detection rate for most DGA botnets, as given in Table 9.

Table 6. The Models' Detection Performance Based on Various Learning Algorithms Using Dataset-01 (%)

Algorithm	PPV	TPR	FPR	FNR	ACC	F1
NB	98.47	91.16	1.64	8.84	94.48	94.67
J48	***98.25***	***95.81***	***1.78***	***4.19***	***96.99***	***97.01***
RF-35	97.27	95.95	2.74	4.05	96.60	96.61
Logistic	98.63	92.97	1.45	7.03	95.60	95.71
SVM	98.70	93.73	1.36	6.27	96.07	96.15

Table 7. The Models' Detection Rate (DR) Based on Various Learning Algorithms for Word-based DGA Botnets (%)

Algorithm Botnet	NB	J48	RF-35	Logistic	SVM
Bigviktor	96.35	96.78	95.28	96.88	97.08
Matsnu	99.13	97.98	97.55	99.10	99.03
Pizd	98.98	98.63	97.50	98.98	98.98
Suppobox	99.48	99.30	96.93	99.48	99.48
Average	**98.51**	**98.19**	**96.81**	**98.63**	**98.66**

Table 8. The Models' Detection Performance Based on Various Learning Algorithms Using Dataset-02 (%)

Algorithm	PPV	TPR	FPR	FNR	ACC	F1
NB	65.30	89.13	27.18	10.87	78.77	75.38
J48	***96.89***	***94.62***	***3.15***	***5.38***	***95.71***	***95.75***
RF-35	96.02	94.78	3.99	5.22	95.39	95.40
Logistic	88.34	90.47	11.29	9.53	89.57	89.40
SVM	88.79	90.15	10.94	9.85	89.59	89.47

Table 9. The Models' Detection Rate (DR) Based on Various Learning Algorithms for DGA Botnets (%)

Algorithm **Botnet**	**NB**	**J48**	**RF-35**	**Logistic**	**SVM**
Bigviktor	77.80	70.70	67.70	88.60	90.00
Matsnu	60.33	98.34	94.59	78.01	81.99
Suppobox	8.40	97.90	99.40	73.60	75.70
Pizd	7.30	99.10	97.70	94.10	97.30
Flubot	73.90	99.20	99.10	96.00	96.10
Necurs	53.40	91.70	90.20	83.10	83.10
Ramnit	51.30	92.10	91.20	84.50	84.50
Ranbyus	72.80	98.00	97.20	94.60	94.90
Rovnix	100.00	99.30	99.60	99.30	99.40
Tinba	27.40	98.90	97.60	61.50	91.40
Cryptolocker	48.50	96.70	95.80	91.80	92.20
Dyre	100.00	100.00	100.00	100.00	100.00
Emotet	96.90	99.40	99.10	97.40	97.70
Gameover	100.00	99.80	99.80	99.90	99.90
Murofet	84.00	99.50	99.70	99.00	99.00
Shiotob	74.60	95.20	94.80	84.00	85.00
Average	**64.82**	**95.98**	**95.32**	**91.14**	**91.92**

4.3 Discussion

From the experimental results given in Table 6, Table 7, Table 8 and Table 9, we can draw the following comments:

- The proposed detection model gives high performance on the Dataset-01 with the overall detection accuracy (ACC) and F1-score of over 95% using 5 learning algorithms. Among them, J48 decision tree performs best with highest detection rate and lowest false alarm rate, as shown in Table 6. The detection rate of 4 typical word-based DGA botnets given in Table 7. also confirms that the model is highly capable of detecting word-based DGA botnets. This means that the selected 16 domain name features are a suitable choice for the classification of word-based DGA domain names and legitimate domain names;
- The proposed detection model also produces good performance on the Dataset-02 with the overall detection accuracy (ACC) and F1-score of over 95% using the decision tree and random forest algorithms. While the models based on logistic regression and SVM achieve the overall detection accuracy (ACC) and F1-score of over 89%, the Naïve Bayes-based model only has the F1-score of about 75%, as presented on Table 8. The detection rate of 4 word-based DGA botnets and 12 character-based DGA botnets shown in Table 9 confirms that the J48 decision tree-based model performs well on most experimental botnets, except '*Bigviktor*'. The SVM-based model has higher detection rate on '*Bigviktor*' than that of J48-based model. However, J48-based model has better detection rate on most botnets than that of SVM-based model.

Table 10 shows the comparison of the detection performance between the proposed model and other DGA botnet detection proposals. Table 11 gives the detection rate comparison of 16 word-based and character-based DGA botnets between our J48 decision tree-based model and the improved DGA botnet detection model proposed in [4]. From the results presented in Table 10 and Table 11, the following comments can be drawn:

- Our model performs much better than other DGA detection proposals, including Truong et al. [8], Hoang et al. [7], Qiao et al. [18], Zhao et al. [19] and Charan et al. [20], except Hoang et al. [4];
- Our model has the almost the same detection performance of Hoang et al. [4]. Specifically, the ACC and F1 of our model and Hoang et al. [4] are 96.99% and 97.01, and 97.03% and 97.03%, respectively;
- Although Hoang et al. [4] slightly performs better than our model on character-based DGA botnets, it is not able to detect word-based DGA botnets. On the other hand, our model has the detection rate of 70.70%, 98.34%, 97.90% and 99.10% for *Bigviktor, Matsnu, Pizd* and *Suppobox* botnets, respectively.

Table 10. The Models' Detection Performance Versus Other Proposals (%)

Approaches	PPV	TPR	FPR	FNR	ACC	F1
Truong et al. [8]	94.70		4.80		92.30	

Hoang et al. [7]	90.70	91.00	9.30		90.90	90.90
Qiao et al. [18]	95.05	95.14				94.58
Zhao et al. [19]			6.14	7.42	94.04	
Charan et al. (C5.0) [21]					95.03	
Hoang et al. [4]	97.08	96.98	2.92	3.02	97.03	97.03
Our model (J48)-Dataset-01	98.25	95.81	1.78	4.19	96.99	97.01
Our model (J48)-Dataset-02	96.89	94.62	3.15	5.38	95.71	95.75

Table 11. The Models' Detection Rate for Word-based DGA Botnets Versus Other Proposals (%)

Proposals Botnet	Our model (J48)-Dataset-02	Hoang et al. [4]
Bigviktor	70.70	3.00
Matsnu	98.34	1.14
Pizd	97.90	-
Suppobox	99.10	0.95
Flubot	99.20	-
Necurs	91.70	98.67
Ramnit	92.10	97.20
Ranbyus	98.00	99.82
Rovnix	99.30	100
Tinba	98.90	98.77
Cryptolocker	96.70	99.00
Dyre	100.00	98.00
Emotet	99.40	99.85
Gameover	99.80	100
Murofet	99.50	99.85
Shiotob	95.20	99.55

5. CONCLUSION

This paper proposes a novel model based on supervised machine learning techniques for detecting word-based DGA botnets. The proposed model improves the detection performance for word-based DGA botnets by using a new set of 16 features for distinguishing word-based DGA and legitimate domain names. Experimental results confirm that the proposed model achieves the F1-score of 97.01% for the word-based DGA dataset (Dataset-01). Moreover, our J48 decision tree-based model also performs well on the combination dataset (Dataset-02) of word-based and character-based DGA domain names with the F1-score of 95.75%. In addition, our model outperforms previous approach [4] in detecting word-based DGA botnets and it has the comparable detection rate to that of [4] for character-based DGA botnets.

For future work, we will continue to enhance our detection model so that it can detect more word-based DGA botnets as well as has higher detection rate for other types of DGA botnets.

ACKNOWLEDGMENT

Authors sincerely thank the Cyber Security Lab, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam for the facility support to complete the research project.

REFRENCES:

[1] Spamhaus Botnet Threat Update: Q1-2021. Available online: https://www.spamhaus.org/news/article/809/spamhaus-botnet-threat-update-q1-2021 (last accessed July 2021).

[2] AO Kaspersky Lab - Bots and botnets in 2018: Statistics on botnet attacks on clients of organizations. Available online: https://securelist.com/bots-and-botnets-in-2018/90091/ (last accessed July 2021).

[3] Radware Blog - More Destructive Botnets and Attack Vectors Are on Their Way. Available online: https://blog.radware.com/security/botnets/2019/10/scan-exploit-control/ (last accessed July 2021).

[4] Hoang, X.D.; Vu, X.H., An improved model for detecting DGA botnets using random forest algorithm, Information Security Journal: A Global Perspective, July 2021, DOI: 10.1080/19393555.2021.1934198.

[5] Alieyan, K.; Almomani, A., Manasrah, A.; Kadhum, M.M., A survey of botnet de-tection based on DNS. Nat. Comput. Appl. Forum 2017, 28, 1541–1558.

[6] Li, X.; Wang, J.; Zhang, X., Botnet Detection Technology Based on DNS. J. Future Internet 2017, 9, 55.

[7] Hoang, X.D.; Nguyen, Q.C., Botnet Detection Based on Machine Learning Techniques Using DNS Query Data. J. Future Internet 2018, 10, 43; doi:10.3390/fi10050043.

[8] Truong, D.T; Cheng, G., Detecting domain-flux botnet based on DNS traffic features in managed network. Security Comm. Networks 2016; 9: 2338–2347; John Wiley & Sons.

[9] Sangani, N.K., Zarger, H., "Machine Learning in Application Security," Book chapter in "Advances in Security in Computing and Communications", IntechOpen, 2017.

[10] Daniel Gibert, Carles Mateu, Jordi Planes. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges, Journal of Network and Computer Applications, 2020.

[11] Jiang, N., Cao, J., Jin, Y., Li, L., Zhang, Z.L., Identifying suspicious activities through DNS failure graph analysis. In 18th IEEE international conference on network protocols (ICNP), pp 144–153, 2010.

[12] Stalmans, E., Irwin, B., A framework for DNS based detection and mitigation of malware infections on a network. In IEEE Information security South Africa (ISSA), pp 1–8, 2011.

[13] Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, N., Dagon, D., Detecting malware domains at the upper DNS hierarchy. In SEC'11: Proceedings of the 20th USENIX conference on Security, 2011.

[14] Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M., EXPOSURE: finding malicious domains using passive DNS analysis. In: NDSS, 2011.

[15] Yadav, S., Reddy, A.K.K., Reddy, A.L.N., Ranjan, S., Detecting Algorithmically Generated Domain-Flux Attacks With DNS Traffic Analysis. IEEE/ACM Trans. Netw. 2012, 20, 1663–1677. doi:10.1109/TNET.2012.2184552.

[16] Kheir, N., Tran, F., Caron, P., Deschamps, N., Mentor: positive DNS reputation to skim-off benign domains in botnet C&C blacklists. In ICT systems security and privacy protection. Springer, Berlin, Heidelberg, pp 1–14, 2014.

[17] Woodbridge, J., Anderson, H.S., Ahuja, A., Grant, D., Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. arXiv 2016, arXiv:1611.00791.

[18] Qiao, Y., Zhang, B., Zhang, W., Sangaiah, A.K., and Wu, H., DGA Domain Name Classifi-cation Method Based on Long Short-Term Memory with Attention Mechanism. Appl. Sci. 2019, 9, 4205; doi:10.3390/app9204205.

[19] Zhao, H.; Chang, Z.; Bao, G., Zeng, X., Malicious Domain Names Detection Algorithm Based on N-Gram. Journal of Computer Networks and Communications 2019. Vol. 2019; doi: 10.1155/2019/4612474; Hindawi.

[20] Yang, L.; Zhai, J.; Liu, W.; Ji, X.; Bai, H.; Liu, G.; Dai, Y., Detecting Word-Based Algorithmically Generated Domains Using Semantic Analysis. Symmetry 2019, 11, 176. https://doi.org/10.3390/sym11020176.

[21] Charan, P.V.S.; Shukla, S.K.; Anand, P. M., Detecting Word Based DGA Domains Using Ensemble Models, In: Krenn S., Shulman H., Vaudenay S. (eds) Cryptology and Network Security. CANS 2020. Lecture Notes in Computer Science, vol 12579. Springer, Cham. https://doi.org/10.1007/978-3-030-65411-5_7.

[22] Ren, F.; Jiang, Z.; Wang, X.; Liu, J., A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. Cybersecurity 3, 4 (2020). https://doi.org/10.1186/s42400-020-00046-6.

[23] Satoh, A.; Fukuda, Y.; Kitagata, G.; Nakamura, Y., A Word-Level Analytical Approach for Identifying Malicious Domain Names Caused by Dictionary-Based DGA Malware. Electronics 2021, 10, 1039. https://doi.org/10.3390/electronics10091039.

[24] DN Pedia – Top Alexa one million domains. Available online: https://dnpedia.com /tlds/topm.php (last accessed July 2021).

[25] Netlab 360 – DGA Families. Available online: https://data.netlab.360.com/dga/ (last accessed July 2021).

[26] English dictionary - 58 000 English words. Available online: http://www.mieliestronk.com/wordlist.html (last accessed July 2021).

[27] Top 1500 English Nouns. Available online: https://www.talkenglish.com/vocabulary/top-1500-nouns.aspx (last accessed July 2021).

[28] DGA algorithms, Available online: https://github.com/baderj/domain_generation_algorithms (last accessed July 2021).