symmetry

MDPI

Article

Detecting Word-Based Algorithmically Generated Domains Using Semantic Analysis

Luhui Yang [1], Jiangtao Zhai [2], Weiwei Liu [1], Xiaopeng Ji [1], Huiwen Bai [1], Guangjie Liu [1,*] and Yuewei Dai [2]

[1] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China; yangluhui005@foxmail.com (L.Y.); lwwnjust5817@gmail.com (W.L.); jixiaopeng_nj@163.com (X.J.); Baihw2035@163.com (H.B.)

[2] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210094, China; jiangtaozhai@gmail.com (J.Z.); dywjust@163.com (Y.D.)

* Correspondence: gjieliu@njust.edu.cn; Tel.: +86-025-8431-5467

Received: 13 December 2018; Accepted: 29 January 2019; Published: 2 February 2019

Abstract: In highly sophisticated network attacks, command-and-control (C&C) servers always use domain generation algorithms (DGAs) to dynamically produce several candidate domains instead of static hard-coded lists of IP addresses or domain names. Distinguishing the domains generated by DGAs from the legitimate ones is critical for finding out the existence of malware or further locating the hidden attackers. The word-based DGAs disclosed in recent network attack events have shown significantly stronger stealthiness when compared with traditional character-based DGAs. In word-based DGAs, two or more words are randomly chosen from one or more specific dictionaries to form a dynamic domain, these regularly generated domains aim to mimic the characteristics of a legitimate domain. Existing DGA detection schemes, including the state-of-the-art one based on deep learning, still cannot find out these domains accurately while maintaining an acceptable false alarm rate. In this study, we exploit the inter-word and inter-domain correlations using semantic analysis approaches, word embedding and the part-of-speech are taken into consideration. Next, we propose a detection framework for word-based DGAs by incorporating the frequency distribution of the words and that of part-of-speech into the design of the feature set. Using an ensemble classifier constructed from Naive Bayes, Extra-Trees, and Logistic Regression, we benchmark the proposed scheme with malicious and legitimate domain samples extracted from public datasets. The experimental results show that the proposed scheme can achieve significantly higher detection accuracy for word-based DGAs when compared with three state-of-the-art DGA detection schemes.

Keywords: network attack; domain generation algorithm; DGA detection; semantic analysis; ensemble classifier

1. Introduction

In most network security events, attackers often use command and control (C&C) servers to maintain full control over the victim hosts for long periods of time after injecting malicious softwares. To establish the connection between the C&C server and the victim hosts, the earlier attackers always embeded static hard-coded lists of IP addresses or domain names in the source code of the malicious softwares. However, these C&C servers with fixed IP address or domains can be easily located and blocked by the network defenders. In recent years, an increasing number of network attackers have begun to use the domain-fluxing technology [1], which applies the domain generation algorithm (DGA) to dynamically produce several candidate domains, and registers these domains at the network service provider. By this way, the malicious malware injected into the victim host only needs a random seed and a built-in DGA algorithm to generate some existing C&C server domain names. This technique was first adopted in the field of botnet [1]. In the past few years, some reports (FireEye, Apt34, https://www.fireeye.com/blog/threat-research/2017/12/targeted-attack-in-middle-east-by-apt34.html) on advanced persistent threat (APT) have revealed that many attackers have applied DGA in more common scenarios. According to the basic constituent elements of the generated domains, DGA can be categorized into three types: word-based DGA, character-based DGA, and hybrid DGA.

Among them, the character-based DGA is the most primitive one, which uses a random seed to select the alphabet or numbers to generate domain names [2]. These algorithmically generated domains (AGDs) maintain much stronger randomness in character level when compared to legitimate domains, which leads to the vulnerability to the entropy-based detection method. Recently proposed character-based DGAs using hidden Markov models (HMMs) and probabilistic context-free grammars (PCFGs) have been able to resist the entropy-based detection method [3]. Nevertheless, they can still be detected by deep neural networks with character sequences as inputs.

The word-based DGAs, e.g., *Suppobox* [4] and *Matsnu* [5], have been disclosed in recent network security events. These DGAs choose some specific English words to form a candidate dictionary for generating domains. After analyzing the top one million domains (T1MD) (Cisco Umbrella, Top1million domains, http://s3-us-west-1.amazonaws.com/umbrella-static/index.html) on the Internet, we have found out that over 67% domains contain at least one English word, and nearly 30% domains are entirely composed of English words. This fact suggests that the word-based DGAs may have stronger stealthiness when its components are legitimate English words. Some prior works have proven the undetectability of word-based DGAs [6]. Thus, the detection of word-based DGAs still remains a challenging work in the field of C&C detection.

In this study, we propose a novel detection method for word-based DGAs by analyzing semantic features including word-wise distribution, character-wise distribution, and their correlations. We first analyze the frequency distributions of words and part-of-speech. Next, we exploit the inter-word and inter-domain correlations using semantic analysis approaches, the word embedding and the part-of-speech are also taken into consideration. Then, we propose a detection framework for word-based DGAs by incorporating the feature set into an ensemble classifier, which is constructed from Naive Bayes, Extra-Trees, and Logistic Regression. The proposed scheme is benchmarked on malicious and legitimate domain samples extracted from public datasets. A series of experiments are carried on to compare the proposed scheme with three state-of-the-art methods. The experimental results show that the proposed scheme can achieve significantly higher detection accuracy for word-based DGAs when compared with the three state-of-the-art detection schemes.

The followings are the key contributions of this paper:

(1) We propose two types of features for DGA detection, inter-word correlation and inter-domain correlation, which have proved to be effective for DGA detection. In addition, these features can also be adapted for detecting multi-word-based DGAs besides the common two-word-based DGAs.

(2) We propose a novel detection framework for word-based DGAs based on semantic analysis and ensemble classifier. A comparative analysis with the state-of-the-art DGA detection schemes is given in detail based on a serial of public datasets.

The rest of the paper is organized as follows. In the next section, we summarize the related works on the existing DGA detection methods and semantic analysis. In Section 3, we present the semantic analysis of word-based domains, including the legitimate ones and AGDs. The word frequency, part-of-speech frequency, inter-word correlation, and inter-domain correlation are analyzed, respectively. In Section 4, we describe the proposed detection framework including word segmentation, the design of the feature set, and the ensemble classifier. Experimental results are presented in Section 5. Finally, in Section 6, we give a conclusion for this paper and discuss the future work.

2. Related Work

As DGAs have been developed as common tools for network attackers in recent years, DGA detection has significant importance for defending network security. In [2], DGAs are classified into two categories according to their functions, they are binary-based DGAs that are always used for mapping into the C&C server, and script-based DGAs that are embedded in JavaScript code loaded in the browser. As the most widely studied branch of DGAs, binary-based DGAs have developed six main categories, including *Dictionary of specific words* that generates domains by selected words, *Use of dynamic DNS* that is often used by CDN providers, *Alphabetic layout* that generates domains using letters of alphabet, *Numeric layout* that only uses numbers to form a domain, *Alphanumeric layout* that uses both letters and numbers to generate a domain, and *Hybrid layout* which may apply letters, numbers, and words to randomly generate a domain. Except for the *Dictionary of specific words*, other types of binary-based DGAs can be concluded as the character-based DGAs. Most of the existing works on DGA detection focus on the character-based DGAs.

Domain-based and behavior-based detections are the two main types of DGA detection schemes. Earlier researchers began the DGA detection with domain-based methods since it is easy to implement in the real world, the main advantage of these methods is that long periods of observations and additional information are not required. In [1], the KL distance, Jaccard index coefficients, and edit distance of AGDs are analyzed to form identification features. More features and linear regression classifier are used for classification in [7]. This method can effectively detect some AGDs, whereas the detection efficiency is limited by the effectiveness of features, and the classifier is not well-designed. Then, two basic linguistic features named meaningful character ratio and n-gram normality score are introduced in [8,9], respectively. The meaningful character ratio calculates the ratio of characters in a domain that can form a meaningful word, and the n-gram normality score is obtained by finding n-grams within a domain and calculating their counts in the English language. The mean and covariance of these features are calculated from a set of legitimate texts. Next, the EXPOSURE system was proposed in [10], 15-dimensional features of domain name were extracted, and J48 decision tree was used for classification.

In recent years, deep learning has made significant progress in many fields, some researchers have introduced deep learning into DGA detection. In [6], a long short-term memory (LSTM) model was used to detect DGA and the detection accuracy on most character-based DGA family can achieve about 100%. However, the detection performance on word-based DGA is still poor. Inspired by [6], a CNN-based DGA detection algorithm was proposed in [11], and the authors made a comparative analysis of the proposed scheme, LSTM-based method, and Random Forest methods. All the above schemes have achieved significantly better detection performance on character-based DGAs than the traditional schemes. However, they still cannot effectively distinguish between the word-based AGDs and word-based legitimate domains.

Some researchers introduce behavior analysis on network flows into DGA detection as they belive that the domain name is not sufficient for accurate detection. In [12], a detection scheme based on the length distribution of DNS request domain name was proposed, which can be used for detecting unknown DGAs. In [13], a detection model on normal DNS domain names for recognizing abnormal domain names was established, it uses natural language processing (NLP) to analyze the character features. In [14], the method based on network flow information over DNS traffic rather than domain names was proposed, but it is limited by the difficulty of collecting the flow information in large-scale networks. In [15], offline analysis to detect DGA botnets through whitelist filtering and clustering was given. In [16], BotMeter, a tool that charts the DGA-bot population landscapes in large-scale networks was proposed, which relies on a long period of analysis.

In actual fact, the methods in [12–16] are all limited by the status of the network environment and data integrity. In real networks, especially in large-scale networks, these traffic features are very difficult to collect. Thus, we concentrate on the DGA detection based on domain names in this study, predominantly on the word-based DGA detection, the proposed scheme can be combined with other traffic analysis schemes to further improve the detection performance when the required network traffic is available.

As semantic analysis has been used in many tasks, e.g., analysis of the syntactic and semantic facets of the web information [17], source code plagiarism detection and investigation [18], and automated essay evaluation [19]. Partly inspired by these works, the semantic analysis methods are introduced into the design of the word-based DGA detection scheme.

Semantic similarity is a subset of the notion of semantic relatedness only considering taxonomic relationships in the evaluation of the semantic interaction between two elements [20]. Measuring the semantic similarity between words is an important component in various tasks. There have been a variety of methods for measuring the semantic similarity. In [21], an empirical method to estimate semantic similarity using page counts and text snippets retrieved from a web search engine for two words was proposed, it has achieved state-of-the-art of web-based semantic similarity measures. In [22], a document representation method based on fuzzy-rough hybrid approach was proposed, it is an effective approach that can be extended to process document in different languages.

The most popular achievement on semantic similarity is word representation. Distributed representations of words in a vector space help learning algorithms to achieve better performance in NLP tasks. In [23], a vector space model (VSM) for automatic indexing was proposed, it translates a document into a vector in terms of some statistics such as Term Frequency (TF) and Inversed Document Frequency (IDF). VSM only considered the frequency of word but did not consider the semantic relatedness of words. Improved from VSM, a better word representation method named latent semantic analysis(LSA) was proposed [24], it assumed that words that are close in meaning will occur in similar pieces of text, and used global matrix factorization to generate word vector. Some other researchers also considered the word co-occurrence for better word representation [25–27]. The further approaches are to learn word representations within local context windows and neural network, e.g., a word representation method based on context windows and deep learning architectures was proposed in [28]. In 2013, Mikolov et al.[29] proposed a state-of-the-art word representation method based on Continuous Bag-of-Words (CBOW) and skip-gram model with a simple single-layer architecture [29–31], which is known as *word2vec* (Google, word2vec, https://code.google.com/archive/p/word2vec/), after that in [32], *GloVe* (GloVe, https://nlp.stanford.edu/projects/glove/) was proposed, which is another effective word representation approach. Both *word2vec* and *GloVe* have been regarded as the most widely used word representation tools.

Word2vec can be used for learning high-quality word vectors from huge data sets with billions of words and none of the previously proposed architectures can be successfully trained on more than a few hundreds of millions of words. It has been a wildly used model for word representation on many kinds of NLP tasks, this kind of word representation goes beyond simple syntactic regularities, but also other multiple degrees of similarity. As the skip-gram methods used in *word2vec* are trained on separate local context windows instead of on global co-occurrence counts, *GloVe* is developed to use the statistics of the corpus better, which has been proven to be more effective than *word2vec* for some NLP tasks. In this study, we employ *word2vec* as the word embedding scheme as it is always regarded as a more general word embedding tool.

3. Semantic Analysis of Word-Based AGDs

As the word-based domains are composed of several words, we analyze their intrinsic features using natural language processing methods in this section. Firstly, we analyze the word frequency distribution of word-based AGDs using the ranking result of the word frequency counted from top one million domains. Then, we analyze the part-of-speech frequency and 2-gram part-of-speech frequency. Besides these distribution-based features, we develop inter-word correlation and inter-domain correlation as correlation-based features. The inter-word correlation measures the correlation among all the words in a single domain. The inter-domain correlation measures the correlation between a domain and a set of legitimate domains. The semantic analysis of word-based AGDs is given in detail. We analyze the differences between the word-based AGDs and the legitimate domains from four aspects: word frequency analysis, part-of-speech analysis, inter-word correlation analysis, and inter-domain correlation analysis.

3.1. Word Frequency Analysis

Word frequency analysis is one of the most fundamental analytic methods in semantic analysis. The frequency distribution of words is quite different between the word-based AGDs and the legitimate domains. To clearly illustrate the difference, we count the frequency of each word in the legitimate domains and the word-based AGDs, respectively. Here, we use T1MD as the legitimate domain set, use *Matsnu* and *Suppobox* as the AGD set. For each domain set, we rearrange the frequency of the words from large to small. Let $\mathbf{\Lambda}= \{c_1, ..., c_N\}$ be the word set which contains all different words in a given domain set, which satisfies $p(c_i) \geq p(c_j)$ for arbitrary $i < j$. $p(c_i)$ denotes the probability of the word c_i, and $\sum\limits_{i=1}^{N} p(c_i) = 1$. As the word sets for different domain sets are varying, we merge the probabilities of adjacent r words in the word set to form L-dimensional frequency coefficients $\boldsymbol{q} = [q_1, ..., q_L]$, where $r = \lceil N/L \rceil$.

$$\begin{cases} q_i = \sum\limits_{j=(i-1)*r}^{i*r} p\left(c_j\right), & i = 1, ..., L-1 \\ q_i = \sum\limits_{j=(L-1)*r}^{N} p\left(c_j\right), & i = L \end{cases} \tag{1}$$

The frequency coefficients $\boldsymbol{q}$ for the legitimate domains and word-based AGDs are shown in Figure 1. The horizontal axis denotes the index of each frequency coefficient, the vertical axis denotes the corresponding frequency coefficient.

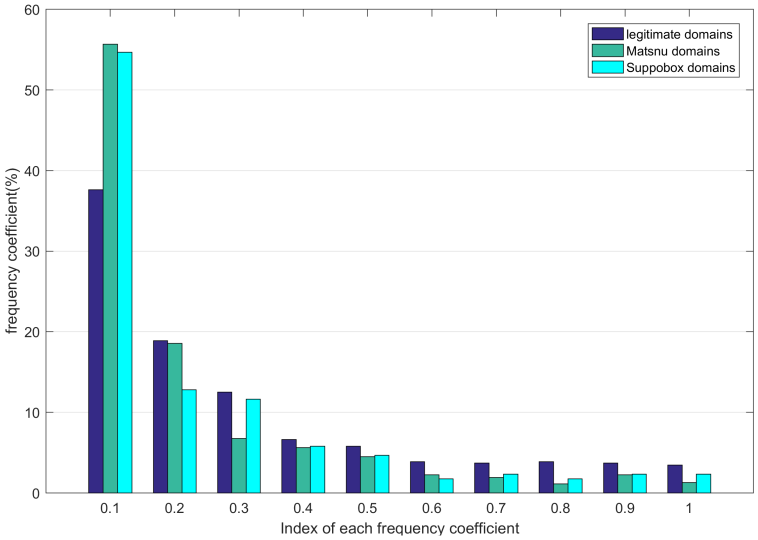
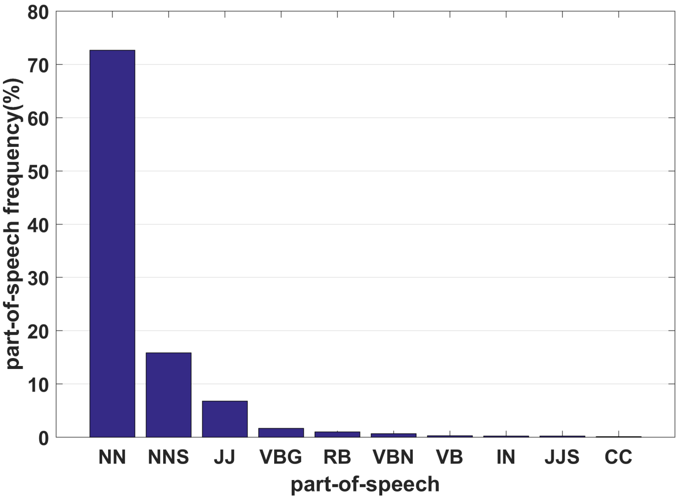


Figure 1. Word frequency results of *Matsnu*, *Suppobox*, and legitimate domains.

As shown in Figure 1, we analyze the frequency coefficients of 1000 random selected legitimate, *Matsnu*, and *Suppobox* domains, there exist a significant difference between the frequency coefficients of the legitimate domains and the word-based AGDs. The largest frequency coefficients for legitimate, *Matsnu* and *Suppobox* domains all belong to 0.1; however, the frequency coefficients of *Matsnu* and *Suppobox* domains are much higher than that of legitimate domains, the frequency coefficients of *Matsnu* and *Suppobox* are about 55% and 54%, while the frequency coefficients of legitimate domains is only 38%. When the index of frequency coefficient is larger than 0.1, the frequency coefficient of legitimate domains is higher than that of *Matsnu* and *Suppobox* domains, especially when the index is larger than 0.5, this result is more obvious. The statistical result denotes that the words of legitimate domains appear more frequently than words in AGDs. We can use the frequency coefficients as one type of features to help distinguish between the legitimate domains and AGDs.

3.2. Part-of-Speech Analysis

Part-of-Speech is a category of words which have similar grammatical properties. The statistical characteristics of the part-of-speech of the legitimate domains are significantly different from that of the word-based AGDs. Here, we extract the part-of-speech of words using NLTK [33]. The probability distributions of the part-of-speech of the words in T1MD are shown in Figure 2. We only give the statistical results for the part-of-speech with the largest ten probabilities. The horizontal axis denotes the type of the part-of-speech, the vertical axis denotes the corresponding probability.



* *NN means noun, NNS means plural noun, JJ means adjective, VBG means gerund or present participle of verb, RB means adverb, VBN means past participle of verb, VB means base form of verb, IN means preposition or subordinating conjunction, JJS means superlative adjective, CC means coordinating conjunction*

Figure 2. Histogram of Part-of-Speech frequency.

As shown in Figure 2, for legitimate domains, more than 70% of the part-of-speech is noun, and the total probability of the largest three types of part-of-speech is more than 95%, namely, other types of part-of-speech rarely appear in the legitimate domains. Therefore, the part-of-speech can be considered as a distinctive feature to detect the word-based DGAs.

We define the vector composed of the part-of-speech of each word in a domain as its part-of-speech vector. Here, we analyze the distribution probabilities of the part-of-speech vector of the legitimate domains and word-based AGDs. Hereinafter, we concentrate on the domains that composed of two words for simplicity, unless otherwise stated. We consider the bigram model in the part-of-speech analysis. Figure 3 shows the probability distribution of the part-of-speech vectors for the legitimate domain set and word-based AGDs. The horizontal axis denotes the part-of-speech vector, the vertical axis denotes the corresponding probability.

From Figure 3 we can find that the probability distribution of the part-of-speech vector of the legitimate domain set is significantly different from that of the word-based AGDs. NN-NN is the part-of-speech vector with the largest probability for all the three domain sets. The probabilities of NN-NN in the legitimate domains and *Suppobox* are both between 50% and 60% while that in *Matsnu* is about 85%. NN-NNS is the second frequently appeared part-of-speech vector in the legitimate domains. Nevertheless, this part-of-speech vector nearly never appears in *Suppobox* and *Matsnu*. The probability of NNS-NN in the legitimate domains is also much larger than that in the word-based AGDs. The results in Figure 3 show that a given domain can be regarded as the legitimate one with a high probability when its corresponding part-of-speech vector is NN-NNS or NNS-NN. We can use the probability distribution of part-of-speech vector as one type of features to help distinguish between the legitimate domains and AGDs.

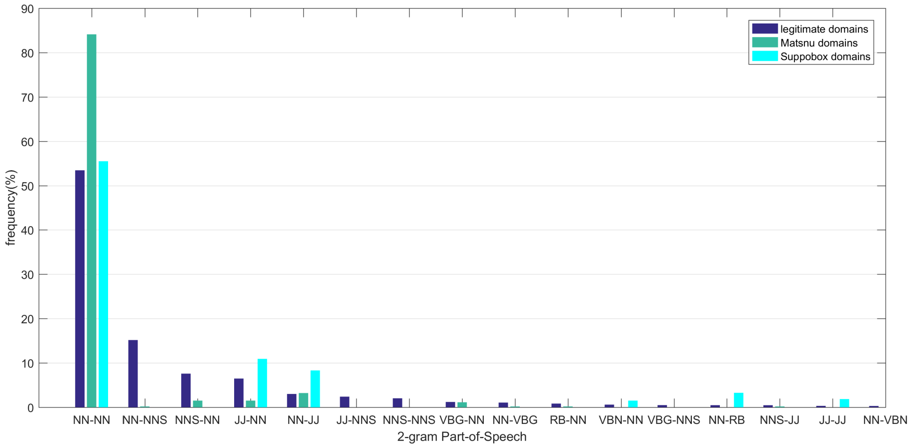


Figure 3. Distribution of 2-gram Part-of-Speech frequency.

3.3. Inter-Word Correlation Analysis

The words used in a legitimate domain name always have a specific meaning, e.g., for websites in the field of movies or sports, the words in the domain are always related to movies or sports. Thus, the similarity between these words and the topic word will be higher than the similarity between randomly selected words. We use the similarity between the two words in a domain to measure the inter-word correlation. The *word2vec* is used for word embedding, so we can calculate the similarity between two words in domains. The distribution of the inter-word correlation for the legitimate domain set, *Suppobox*, and *Matsnu* are shown in Figure 4. The horizontal axis denotes the similarity range. Here we use 10 bins, namely, the i-th similarity range is $((i-1)/10, i/10]$. The vertical axis denotes the probability of each similarity range.

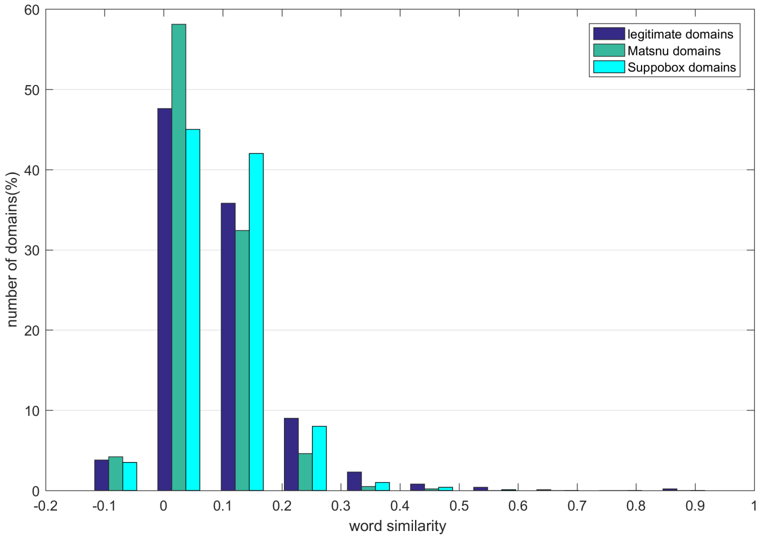


Figure 4. The probability distribution of the inter-word correlation for legitimate domains and AGDs.

From Figure 4 we can find out that the word similarity of the word vector of the legitimate domains is significantly different from that of the word-based AGDs. Most word similarity of legitimate domains, *Matsnu* domains, and *Suppobox* domains belongs to the bin of 0–0.1,in which the ratio of *Matsnu* is about 58% and much more than the legitimate and *Suppobox* domains which are only 47%

and 45%, respectively. As for the bin of 0.1–0.2, the ratio of *Suppobox* domains are about 42% which is much more than the legitimate and *Matsnu* domains whose ratios are 36% and 32%, respectively. When the word similarity is larger than 0.2, the number of legitimate domains is more than *Matsnu* and *Suppobox* domains. We can use the word similarity of word vector of domains as one type of features to help distinguish between the legitimate domains and AGDs.

3.4. Inter-Domain Correlation Analysis

Different legitimate domains of the same type of websites always have a certain similarity in the real world, e.g., there may exist the same topic word in the domains. Using a sufficient number of legitimate domains as the reference samples, the AGDs can be identified by calculating the similarity between them and the legitimate domain set. Let $d = (x_0, y_0)$ be the domain composed of two words, x_0 and y_0 denote the first and second words, respectively. For a legitimate domain set, we construct two domain subsets which are related to the domain d, $D_F = \{(x_1, y_0), (x_2, y_0), \ldots, (x_n, y_0)\}$ denotes the domain set in which all the domains contain the second word y_0, $D_L = \{(x_0, y_1), (x_0, y_2), \ldots, (x_0, y_n)\}$ denotes the domain set in which all the domains contain the first word x_0. We define the former-word-correlation (FWC) and latter-word-correlation (LWC) to measure inter-domain correlation. FWC measures the similarity between x_0 and $\{x_1, x_2, \ldots, x_n\}$. LWC measures the similarity between y_0 and $\{y_1, y_2, \ldots, y_n\}$.

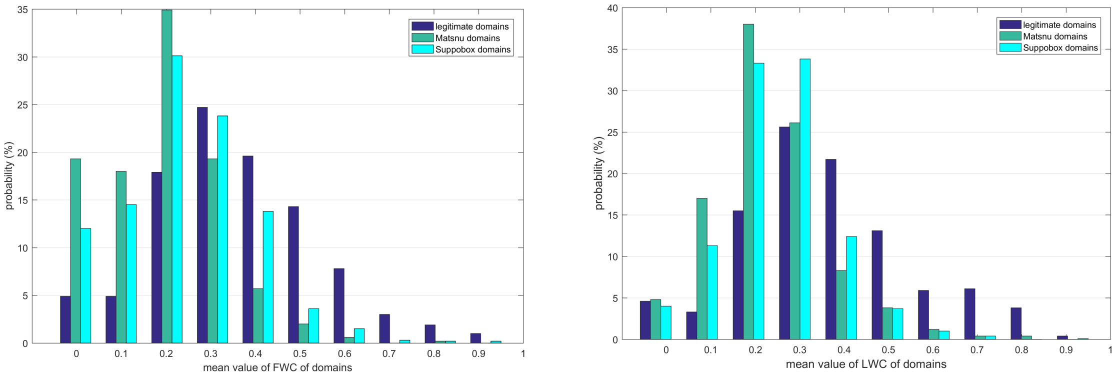
Using *word2vec*, a domain can be translated to be a vector of float values. In this way, when calculating FWC, x_0 and $\{x_1, x_2, \ldots, x_n\}$ can be translated to vector $\boldsymbol{v}_0$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n\}$. $\boldsymbol{v}_0$ is a m-dimensional vector which can be present as $\boldsymbol{v}_0 = [v_{0,1}, v_{0,2}, \ldots, v_{0,m}]$. Cosine distance of two vectors can be used to represent the similarity between two words. The LWC can be calculated. y_0 and $\{y_1, y_2, \ldots, y_n\}$ can be translated to vector $\boldsymbol{u}_0$ and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n\}$.

$$\begin{aligned} FWC &= \{sim(\boldsymbol{v}_0, \boldsymbol{v}_1), sim(\boldsymbol{v}_0, \boldsymbol{v}_2), ..., sim(\boldsymbol{v}_0, \boldsymbol{v}_n)\} \\ LWC &= \{sim(\boldsymbol{u}_0, \boldsymbol{u}_1), sim(\boldsymbol{u}_0, \boldsymbol{u}_2), ..., sim(\boldsymbol{u}_0, \boldsymbol{u}_n)\} \end{aligned} \tag{2}$$

The cosine distance is calculated in Equation (3).

$$\mathrm{sim}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \boldsymbol{v}_2^T}{\|\boldsymbol{v}_1\|_2 \cdot \|\boldsymbol{v}_2\|_2} \tag{3}$$

Figure 5 shows the probability distribution of the mean value of FWC and LWC. The horizontal axis denotes the mean value of FWC and LWC, we divide 1 into 10 bins of equal size and calculate the probability of the values in each bin. The vertical axis denotes the probability corresponding to each bin.



(**a**) FWC (**b**) LWC

Figure 5. FWC and LWC of AGDs and legitimate domains.

It can be seen from Figure 5 that the FWC and LWC of the legitimate domains are significantly different from that of the word-based AGDs. As Figure 5a shows, when the mean value of FWC is smaller than 0.3, the probability of *Matsnu* and *Suppobox* domains is obviously much higher than legitimate domains, and when the mean value of FWC is larger than 0.4, the probability of legitimate domains is obviously higher than *Matsnu* and *Suppobox* domains. As Figure 5b shows, when the mean value of LWC is between 0.1 and 0.2, the probability of *Matsnu* and *Suppobox* domains is much higher than legitimate domains, when the mean value of FWC is larger than 0.4, the probability of legitimate domains is significantly higher than *Matsnu* and *Suppobox* domains. It denotes that the inter-domain correlation of word-based DGA is much lower than the legitimate domains. We can use FWC and LWC as one type of features to help distinguish between the legitimate domains and word-based AGDs.

4. Word-Based AGDs Detection Framework

The framework of the proposed DGA detection scheme is illustrated in Figure 6. We use some word-based legitimate domains and word-based AGDs as the training samples. For each domain d in the training dataset, we firstly segment it into a word list $[w_1, w_2, \ldots, w_n]$. Next, we construct 24-dimensional features from the word list in terms of word frequency, part-of-speech frequency, inter-word correlation, and inter-domain correlation. The 24-dimensional features are used for training an ensemble classifier composed of Naive Bayes, Extra-Trees, and Logistic Regression. With the trained model, we can identify whether a word-based domain is legitimate or algorithmically generated by extracting its corresponding features.

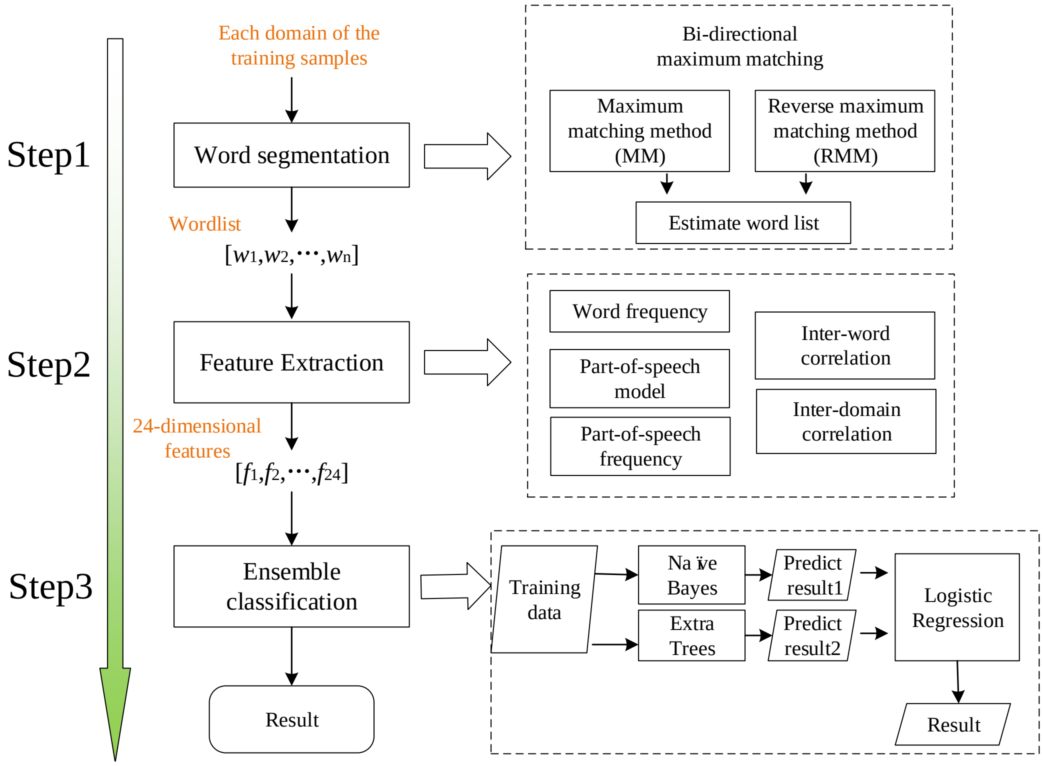


Figure 6. The framework of Word-Based AGDs Detection.

4.1. Word Segmentation Based on Bi-directional Maximum Matching

As there exists no separator between adjacent words in a word-based domain, we need to extract a word list from each domain. Here, we propose word segmentation based on bi-directional maximum matching. The maximum matching method (MM) and reverse maximum matching method (RMM) in Chinese word segmentation methods [34] are introduced into the English word segmentation. In the MM method, We repeatedly remove the rightmost character until the remaining substring is a word. After removing the extracted word, the process continues to find the next word in the rest of the domain string. The RMM method is similar to the MM method, the only difference is that the word identification starts from the rightmost and the character removal starts from the leftmost.

Two word lists $\boldsymbol{f}$ and $\boldsymbol{r}$ can be obtained using MM and RMM segmentation, respectively. If these two word lists are identical, the segmentation result is $\boldsymbol{f}$. In some special cases, the two world lists may be different. We can choose the word list by comparing the occurrence probability of the words. For a word list $\boldsymbol{s} = \{s_1, ..., s_l\}$ composed of l words, the average occurrence probability of the words in this word list can be given by

$$P_{\boldsymbol{s}} = \sum_{i=1}^{l} p\left(s_i\right)/l \tag{4}$$

where $p\left(s_i\right)$ denotes the probability of the word s_i in the corpus in NLTK.

Let $\boldsymbol{f} = \{f_1, ..., f_m\}$ and $\boldsymbol{r} = \{r_1, ..., r_n\}$ be the word lists extracted by MM method and RMM method, respectively. m and n denote the number of words in $\boldsymbol{f}$ and $\boldsymbol{r}$, respectively. We can choose the resulting word list $\boldsymbol{h}$ as follows:

$$\boldsymbol{h} = \begin{cases} \boldsymbol{f} & P_{\boldsymbol{f}} > P_{\boldsymbol{r}} \\ \boldsymbol{r} & P_{\boldsymbol{f}} < P_{\boldsymbol{r}} \\ m < n ? \boldsymbol{f} : \boldsymbol{r} & P_{\boldsymbol{f}} = P_{\boldsymbol{r}} \end{cases} \tag{5}$$

4.2. The Design of Feature Set

For a domain, we define four types of features according to the analysis results in Section 3. They are the features in terms of word frequency, part-of-speech, inter-word correlation, and inter-domain correlation. We describe each type of features in detail as follows:

(1) Features of Word Frequency

The analysis results in Section 3.1 have shown that the frequency of words in AGDs is significantly lower than that in legitimate domains. For a domain word sequence $\boldsymbol{s} = \{s_1, s_2, \ldots, s_n\}$, the corresponding frequency sequence $P(\boldsymbol{s}) = \{p(s_1), p(s_2), \ldots, p(s_n)\}$ can be obtained by

$$P\left(s_i\right) = \frac{C\left(s_i\right)}{C_{total}} \tag{6}$$

where C_{total} is the total number of words in T1MD, and $C(s_i)$ is the number of s_i in T1MD. We use the maximum, minimum, mean, and variance of $P(\boldsymbol{s})$ as features.

(2) Features of Part-of-Speech

The analysis results in Section 3.2 show that the probability distribution of part-of-speech for the legitimate domains is different from that for the word-based AGDs, some types of part-of-speech sequences for the word sequence used in the word-based AGDs may rarely present in the legitimate domains. Thus, we consider 1-gram and 2-gram frequency of part-of-speech. For a given part-of-speech sequence $\boldsymbol{t} = \{t_1, t_2, \ldots, t_n\}$, the 1-gram frequency $P_{t\_1} = \{P(t_1), P(t_2), \ldots, P(t_n)\}$ can be obtained using similar method in (3). We use the frequency of occurrence for t_i in T1MD and the number of words in T1MD to calculate the corresponding 1-gram frequency $P(t_i)$.

The 2-gram frequency $P_{t\_2} = \{P(t_1|t_2), \ldots, P(t_{n-1}|t_n)\}$, where

$$P(t_{i-1}|t_i) = \frac{C(t_{i-1}, t_i)}{C(t_i)} \tag{7}$$

Here, $C(t_{i-1}, t_i)$ is the frequency of the part-of-speech sequence (t_{i-1}, t_i), and denotes the frequency of the part-of-speech t_i. The statistical results are also obtained from T1MD. We use the maximum, minimum, mean, and variance of P_{t_1} and P_{t_2} as features.

(3) Features of Inter-word Correlation

We denote $\boldsymbol{w} = \{w_1, w_2, \ldots, w_n\}$ as the word sequence for a given domain d, the corresponding word embedding set is $\boldsymbol{W} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$, we calculate the similarity between each two adjacent words.

It is apparent that the two adjacent words are more similar when the similarity $\mathrm{sim}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ is closer to 1. The inter-word correlation for the domain d can be obtained by

$$sim_{\boldsymbol{w}} = \{sim(\boldsymbol{v}_1, \boldsymbol{v}_2), ..., sim(\boldsymbol{v}_{n-1}, \boldsymbol{v}_n)\} \tag{8}$$

$sim_{\boldsymbol{w}}$ is an $(n-1)$-dimensional vector. We take the maximum, minimum, mean, and variance of $sim_{\boldsymbol{w}}$ as features.

(4) Features of Inter-domain Correlation

As described in Section 3.4, Let $d = (x, y)$ be the domain composed of the word x and y, and $listf = \{(x_1, y), (x_2, y), \ldots, (x_n, y)\}$ denotes the domain list in which the latter word is y, similarly, $listl = \{(x, y_1), (x, y_2), \ldots, (x, y_n)\}$ denotes the domain list in which the former word is x. The FWC and LWC of d can be calculated respectively, they represent the correlation between the domain d and legitimate domains. *word2vec* was used to calculate the similarity between two words. We take the maximum, minimum, mean, and variance of FWC and LWC as features.

With the above four types of features, the proposed feature set can be depicted in Figure 7.

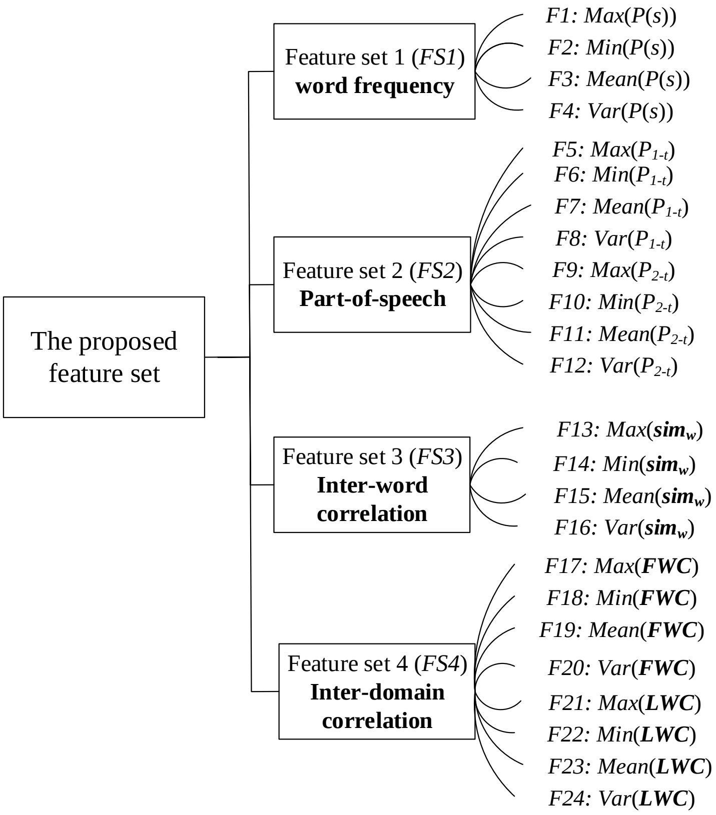

The proposed feature set
Feature set 1 (FS1) word frequency
F1: Max(P(s))
F2: Min(P(s))
F3: Mean(P(s))
F4: Var(P(s))
Feature set 2 (FS2) Part-of-speech
F5: Max($P_{1\text{-}t}$)
F6: Min($P_{1\text{-}t}$)
F7: Mean($P_{1\text{-}t}$)
F8: Var($P_{1\text{-}t}$)
F9: Max($P_{2\text{-}t}$)
F10: Min($P_{2\text{-}t}$)
F11: Mean($P_{2\text{-}t}$)
F12: Var($P_{2\text{-}t}$)
Feature set 3 (FS3) Inter-word correlation
F13: Max($\boldsymbol{sim}_w$)
F14: Min($\boldsymbol{sim}_w$)
F15: Mean($\boldsymbol{sim}_w$)
F16: Var($\boldsymbol{sim}_w$)
Feature set 4 (FS4) Inter-domain correlation
F17: Max(**FWC**)
F18: Min(**FWC**)
F19: Mean(**FWC**)
F20: Var(**FWC**)
F21: Max(**LWC**)
F22: Min(**LWC**)
F23: Mean(**LWC**)
F24: Var(**LWC**)

Figure 7. The proposed feature set for word-based DGA detection.

As shown in Figure 7, 24-dimensional features can be extracted from a domain when we consider word frequency, part-of-speech, inter-word correlation, and inter-domain correlation. For word frequency, as analyzed in Section 3.1, the distribution of word frequency of legitimate domains is significantly different from word-based AGDs, the word frequency of domains can be used to distinguish legitimate domains and AGDs, in this way, four statistical features are extracted considering the word frequency. As for the part-of-speech aspect, the analysis result in Section 3.2 denotes that the distribution of 2-gram part-of-speech frequency can be used to distinguish legitimate domains and word-based AGDs, so eight statistical features are extracted including four features of 1-gram part-of-speech frequency and four statistical features of 2-gram part-of-speech frequency. As for the inter-word correlation, the analysis result in Section 3.3 denotes that the word similarity of words in legitimate domains is different from that of word-based AGDs, in this way, four statistical features of inter-word correlation are extracted. As for the inter-domain correlation, it is obvious that the FWC and LWC of legitimate domains are different from word-based AGDs, in this way, eight statistical features are extracted including four features of FWC and four features of LWC.

4.3. Ensemble Classifier

As different machine learning classifiers have different tendencies for the classification of positive and negative samples, we can get more balanced classification results using multiple different machine learning models. Since Naive Bayes is a classical classifier, it is simple, and a Naive Bayes classifier will converge quicker than discriminative models, such as logistic regression, so it needs less training data. Furthermore, a Naive Bayes classifier still often does a great job in practice even if the Naive Bayes assumption does not hold. Extremely Randomized Trees (Extra-Trees) is similar to but performs better than Random Forest; it builds totally randomized trees whose structures are independent of the output values of the learning sample. Extra-Trees always holds a significant effective classification result in most situations. In this study, we design an ensemble classification method. Using the legitimate and DGA samples to train the Naive Bayes and Extra-Trees classifiers respectively, we can first obtain two classification models. Next, all the training samples are predicted using the trained Naive Bayes model and Extra-Trees model. A Logistic Regression classifier is then trained using the results predicted from the two models. To distinguish between the legitimate domain and AGD for a domain name, the three models will be used together to obtain a classification result as shown in Figure 8.

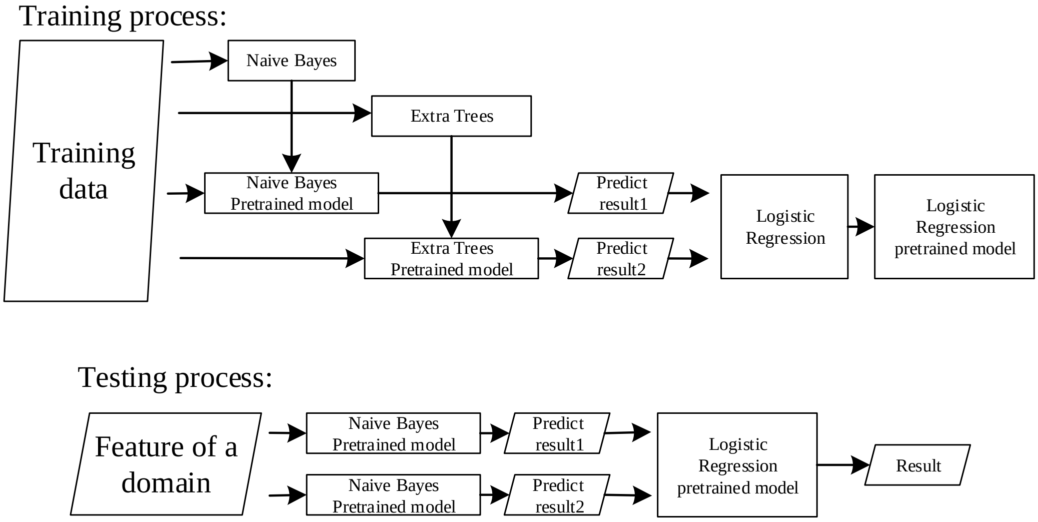

Training process:
Training data
Naive Bayes
Extra Trees
Naive Bayes Pretrained model
Extra Trees Pretrained model
Predict result1
Predict result2
Logistic Regression
Logistic Regression pretrained model
Testing process:
Feature of a domain
Naive Bayes Pretrained model
Naive Bayes Pretrained model
Predict result1
Predict result2
Logistic Regression pretrained model
Result

Figure 8. Structure of proposed ensemble classifier.

5. Experimental Results and Analysis

In this section, we have carried on some experiments and analyzed the results. Firstly, we collect some data from public datasets and generate some data using a designed DGA. Using these data, we design four types of experimental datasets. After that, four common classifiers and the proposed ensemble classifier are used to classify the four types of datasets. The results show that the proposed ensemble classifier performs best on the datasets. Then, we do some experiment on the four feature sets described in Section 4.2, and the results show that every feature set is valid. Finally, a comparative experiment is done to compare the effect of three state-of-the-art methods and the proposed method in this paper. The comparative results show that the proposed method has better classification effect and stronger generalization ability.

5.1. Data Set

In this section, we benchmark the proposed DGA detection scheme with public and self-built domain datasets. The legitimate domain dataset was constructed from the world's top one million DNS request domains collected by Cisco Umbrella (http://s3-us-west-1.amazonaws.com/umbrella-static/index.html), 500,000 domains composed of two words and 50,000 domains composed of three words were chosen as training samples. The *Matsnu* and *Suppobox* domains exposed by Network Security Research Lab (https://data.netlab.360.com/dga/) were chosen as the real AGD samples. The numbers of them are 6877 and 1191, respectively. Besides these public DGA samples, we also use Oxford 3000 words to randomly generate 500,000 two-word DGA samples, 50,000 three-word DGA samples, and 10,000 four-word DGA samples using word-based DGA (WB-DGA). Hereinafter, n-word domain sample denotes the domain composed of n words. *Matsnu* is a DGA that randomly select one word from a list of verbs and another word from a list of nouns, then concatenate these two words and a top-level domain (TLD) to be a generated domain. *Suppobox* is a DGA that select two words randomly from a word list and concatenates these two words and a TLD to be a generated domain. The methodology and samples of three types of DGAs are given in Table 1.

Table 1. The description of DGA used in the experiment.

DGA	Methodology	Samples
Matsnu	Step1: select x from a list of verbs Step2: select y from a list of nouns Step3: concatenate x,y then get domain xy.tld	*analystfinance.com* *landscapeborn.com* *moneylimited.com*
Suppobox	Step1: select two indexes of words randomly Step2: pick up two words x and y using the indexes Step3: concatenate x,y then get a domain xy.tld	*partyprobable.net* *fightprobable.net* *freshwelcome.net*
WB-DGA	Step1: select two or more words randomly from Oxford 3000 words Step2: concatenate these words and a TLD to get a domain	*woundedmoney.com* *eightycabinet.com* *movementborder.com*

As shown in Table 2, we categorize the above legitimate and DGA domain samples into four datasets. DS-MS is composed of the legitimate samples and disclosed two-word DGA samples, it aims to test the detection performance of the proposed method with real samples in the training and testing stages. In DS-LS-2, we use the legitimate and WB-DGA domains as the training set, take the collected *Matsnu* and *Suppobox* domains as the test set, all these real DGA domains are composed of two words. The DS-LS-3 is used for verifying the detection performance on domains composed of more than two words, we chose 50,000 three-word legitimate domains and 50,000 WB-DGA domains as the training set, the collected *Matsnu* and *Suppobox* domains are used as the test set. DS-LSM is used for testing the generalization ability of the proposed scheme when training with the two-word domains and testing with the three-word and four-word domains.

Table 2. Datasets used in the experiments.

Dataset Name	Training Set	Test Set
DS-MS	10,000 legitimate samples(two-word) 6000 Matsnu samples 1000 *Suppobox* samples	1000 legitimate samples(two-word) 877 Matsnu samples 191 *Suppobox* samples 1000 WB-DGA samples(two-word)
DS-LS-2	500,000 normal samples(two words) 500,000 WB-DGA samples(two-word)	10,000 legitimate samples(two-word) 6877 *Matsnu* samples 1191 *Suppobox* samples 10,000 WB-DGA samples(two-word)
DS-LS-3	50,000 legitimate samples(three-word) 50,000 WB-DGA samples(three-word)	10,000 legitimate samples(three-word) 6877 *Matsnu* samples 1191 *Suppobox* samples 10,000 WB-DGA samples(three-word)
DS-LSM	500,000 legitimate samples(two-word) 500,000 WB-DGA samples(two-word)	10,000 legitimate samples(three-word) 10,000 legitimate samples(four-word) 10,000 WB-DGA samples(three-word) 10,000 WB-DGA samples(four-word)

5.2. Experimental Results Using Different Classifiers

The design of the classifier has a significant influence on the detection performance of the proposed scheme. To validate the effectiveness of the ensemble classifier. We compared the detection results of the proposed scheme using different classifiers. The detection results on DS-MS, DS-LS-2, DS-LS-3 are shown in Table 3. DS-MS and DS-LS-2 are both datasets with two-word domains, DS-LS-3 is the dataset with the domains composed of more than two words. We compare the proposed ensemble classifier with four typical classifiers, including Naive Bayes, SVM, Decision Tree, Extra-Trees.

Table 3. Detection results using different classifiers.

Dataset	Classifier	Legitimate Domains (%)	WB-DGA Domains (%)	*Matsnu* Domains (%)	*Suppobox* Domains (%)
DS-MS	Naive Bayes	69.10	82.46	97.38	84.17
	SVM	91.30	46.40	96.67	66.67
	Decision tree	95.80	51.33	100	73.33
	Extra Trees	98.20	43.80	100	75.83
	Ensemble classifier	**96.80**	**44.04**	**100**	**84.17**
DS-LS-2	Naive Bayes	56.95	91.66	89.04	83.38
	Decision tree	95.95	84.92	79.77	74.64
	Extra Trees	96.74	88.47	83.93	77.50
	Ensemble classifier	**96.32**	**92.21**	**88.09**	**83.63**
DS-LS-3	Naive Bayes	73.26	94.26	96.60	92.02
	Decision tree	88.28	87.34	62.76	57.35
	Extra Trees	90.04	88.29	83.93	74.64
	Ensemble classifier	**89.62**	**91.06**	**93.01**	**86.23**

The detection accuracy measures the proportion of actual legitimate or algorithmically generated domains that are correctly identified. Let P_{legit} and P_{AGD} be the detection accuracy for the legitimate domains and the AGDs, respectively. They can be obtained using Equations (9) and (10).

$$P_{legit} = N^{o}_{legit}/N_{legit} \tag{9}$$

$$P_{AGD} = N^{o}_{AGD}/N_{AGD} \tag{10}$$

where N_{legit} and N_{AGD} denote the number of the legitimate domains and the number of the AGDs, respectively, N^{o}_{legit} denotes the number of the correctly identified legitimate domains, N^{o}_{AGD} denotes the number of the correctly identified AGDs.

In the dataset DS-MS, the Extra-Trees method has the best classification accuracy for legitimate domains, it can achieve the detection accuracy of 98.20%. For *Matsnu*, the detection accuracy using the Decision Tree and Extra-Trees can both achieve 100%. For *Suppobox*, Naive Bayes can achieve the best performance. Extra-Trees has achieved the best detection performance among SVM, Decision Tree, and Extra-Trees. The corresponding detection accuracy for legitimate domains, *Matsnu*, and *Suppobox* are 98.20%, 100%, and 75.83%, respectively. The ensemble classifier performs best among all classifiers, the detection accuracy for legitimate domains, *Matsnu*, and *Suppobox* can achieve 96.80%, 100%, 84.17%, respectively. Nevertheless, the ensemble classifier detect WB-DGA. It reveals that even the classifiers perform well for trained DGA families, they still cannot work for unknown DGA families.

A larger dataset DS-LS-2 is used to benchmark the generalization ability of the proposed scheme, the experimental results in Table 2 show that Naive Bayes has the best detection performance for AGDs among all single classifiers. The corresponding detection accuracy can achieve 91.66%, 89.04%, and 83.38% for the WB-DGA domains, *Matsnu*, and *Suppobox*, respectively, but the detection accuracy for the legitimate domains is only 56.95%. For the legitimate domains, Extra-Trees can achieve the best detection accuracy. For the ensemble classifier , the detection accuracy can achieve 95.76% for legitimate domains, 88.09% for *Matsnu*, and 83.63% for *Suppobox*, which performs better than the Naive Bayes and Extra-Trees on average. The enhancement of detection performance has proven the effectiveness of the ensemble classifier.

To validate the effectiveness of the proposed scheme for domains composed of more than two words, we benchmark the proposed scheme with DS-LS-3. The results are similar to the results of DS-LS-2. The Extra-Trees has the best detection accuracy for the legitimate domains. Naive Bayes can achieve the best detection accuracy for WB-DGA. While the ensemble classifier performs best on average with the detection accuracy of 89.62% for the legitimate domains and 89.62% for WB-DGA domains. The two-word *Matsnu* and *Suppobox* samples are also tested on this model and have obtained detection accuracy of 93.01% and 86.23%, respectively. The results show that the model trained on three-word domains can also be used for detecting two-word DGA. It demonstrates that the proposed scheme can adapt to the varying number of words in domains.

To benchmark the effectiveness for multi-word domains, we test the proposed scheme with DS-LSM. The detection performance for multi-word domains is shown in Table 4.

Table 4. Detection performance on DS-LSM.

Classifier	Three-Word Legitimate Domains (%)	Four-Word Legitimate Domains (%)	Three-Word WB-DGA Domains (%)	Four-Word WB-DGA Domains (%)
Naive Bayes	86.27	92.30	83.62	74.91
Decision tree	79.14	80.00	82.34	79.54
Extra Trees	92.98	95.38	74.06	66.77
Ensemble classifier	**90.49**	**93.25**	**83.96**	**77.19**

From the experimental results, it can be seen that the model trained with two-word domains can also be used for detecting three-word and four-word AGDs, but the detection performance decreases with the increasing number of words. The main reason is that we deliberately use the samples of two-word to train the model, and test the model with three-word and four-word samples. The mismatch of the training set and the testing set will cause the performance reduction. Secondary, when a domain contains more words, the relatedness of two adjacent words may be weaker, which will have a negative impact on the detection result. However, in the actual network environment, there are very few domains containing more than four words. For AGDs with less than four words, the proposed model can maintain more than 77% detection accuracy. For the legitimate domains, the detection accuracies for three-word and four-word domains are greater than 90%. The above results have shown that the proposed scheme has high detection accuracy and good generalization ability.

5.3. Experiment Results Using Different Feature Sets

There are four types of features in the proposed feature set. As shown in Figure 7, *FS1* and *FS2* are features on word and part-of-speech frequency, respectively. *FS3* are the features on inter-word correlation, and *FS4* are features on inter-domain correlation. To compare the performance of different types of features, we examine the detection performance using combinations of different features. Table 4 shows the detection accuracy with *FS1*, *FS1*+*FS2*, *FS1*+*FS2*+*FS3*, and *FS1*+*FS2*+*FS3*+*FS4*, respectively. The adopted dataset is DS-LS-2, and the adopted classifier is the ensemble classifier.

It can be seen from Table 5, the classification results grow better and better both on legitimate domains and AGDs with the increase of features. The experimental results prove that the four feature sets proposed in this paper are all valid features for detection word-based DGA.

Table 5. Detection performance using different feature sets.

Feature Set	Legitimate Domains (%)	WB-DGA Domains (%)	*Matsnu* Domains (%)	*Suppobox* Domains (%)
$FS1$	78.19	78.11	79.21	76.49
$FS1 + FS2$	86.14	82.63	81.85	81.27
$FS1 + FS2 + FS3$	94.66	89.60	86.68	82.48
$FS1 + FS2 + FS3 + FS4$	96.32	92.21	88.09	83.63

5.4. Comparative Experiments with the State-of-the-Art Methods

We also compare the proposed scheme with three state-of-the-art DGA detection schemes, including a feature-based scheme [10], a CNN-based detection scheme [11], a LSTM-based detection scheme [6]. The comparative experimental results are shown in Table 6.

Table 6. Detection results using different DGA detection schemes.

Dataset	Algorithm	Detection Accuracy(%)				Average	Average
		Legitimate Domains	WB-DGA Domains	*Matsnu* Domains	*Suppobox* Domains	Detection Rate(%)	False Alarm(%)
DS-MS	CNN-based [11]	84.39	33.44	100	84.54	62.14	12.21
	LSTM-based [6]	76.98	47.75	95.18	83.18	63.20	15.75
	feature-based [10]	73.23	42.32	63.40	60.12	51.52	22.34
	the proposed	**96.80**	**44.04**	**100**	**84.17**	**67.98**	**0.98**
DS-LS-2	CNN-based [11]	92.82	90.10	73.36	72.21	82.54	4.59
	LSTM-based [6]	90.41	92.85	85.45	80.82	89.24	5.61
	feature-based [10]	71.63	66.42	62.58	58.76	64.46	19.59
	the proposed	**96.32**	**92.21**	**88.09**	**83.63**	**89.91**	**2.22**
DS-LS-3	CNN-based [11]	91.96	88.12	32.89	52.43	64.74	6.43
	LSTM-based [6]	88.98	90.45	29.51	19.86	62.60	8.88
	feature-based [10]	68.42	56.23	45.87	48.56	51.78	25.24
	the proposed	**89.62**	**91.06**	**93.01**	**86.23**	**91.48**	**5.91**

From Table 6 we can see that the proposed scheme has the best performance on the average detection rate and the average false alarm rate among all the four schemes. For the dataset DS-MS which is used for benchmarking the detection performance on known DGA families, the average detection rates of CNN-based and LSTM-based detection schemes are both about 60%, and their average false alarm rates are both higher than 10%. The proposed scheme has significantly better performance on detection accuracy when compared with them. The average detection rate and false alarm rate of the proposed scheme are about 68% and 1%, respectively. Among the four schemes, the traditional feature-based detection scheme has the worst detection performance. Its detection rate is about 12% and 16% lower than the deep-learning-based scheme and the proposed scheme, respectively.

For the dataset DS-LS-2 which is used for benchmarking the detection performance on unknown DGA families, the proposed scheme can still achieve significantly better performance when compared with other three state-of-the-art detection schemes. Among existing detection schemes for word-based AGDs, LSTM-based scheme has obtained the best average detection accuracy, its detection rate is 89.24%, the CNN-based scheme has obtained the lowest false alarm rate which is 4.59%. In contrast, the average detection rate and false alarm rate of the proposed scheme are 89.91% and 2.22%, respectively. For the Matsnu and Suppobox domains that are not involved in the training of the detection model, the proposed scheme has achieved the best detection accuracy which is 88.09% and 83.63%, respectively. The comparative results on DS-LS-2 demonstrate that the proposed scheme has better detection performance for unknown DGAs when compared with the existing schemes.

The dataset DS-LS-3 is used for benchmarking the performance when the word number of the domains in the training set is different from that in the testing set. The training set of it is composed of three-word domains, the testing set is composed of two-word domains, including the legitimate domains, WB-DGA, Matsnu domains, and Suppobox domains. As shown in Table 6, the proposed scheme is still significantly better than the other three schemes, the average detection rate and false alarm rate of the proposed scheme are 91.48% and 5.91%. Although the CNN-based method can achieve better performance on the legitimate domains, the detection accuracy of the proposed scheme for the WB-DGA domains, Matsnu domains, and Suppobox domains are all better than it. The results demonstrate that the proposed scheme has stronger generalization capability than the CNN-based and LSTM-based detection schemes.

For further comparison of the generalization capability, we evaluated the four schemes with the dataset DS-LSM. The training set is composed of the two-word legitimate samples and WB-DGA samples, the testing set is composed of three-word and four-word legitimate samples and WB-DGA samples. The detection results are shown in Table 7.

Table 7. Detection performance on DS-LSM.

Algorithm	**Detection Accuracy (%)**				**Average Detection Rate (%)**	**Average False Alarm (%)**
	Three-Word Legitimate Domains	**Four-Word Legitimate Domains**	**Three-Word WB-DGA Domains**	**Four-Word WB-DGA Domains**		
CNN-based [11]	13.95	35.82	89.25	95.05	92.15	44.91
LSTM-based [6]	6.61	5.50	89.62	76.63	83.20	53.06
feature-based [10]	23.45	18.12	34.54	33.67	34.11	69.90
the proposed	**90.49**	**93.25**	**83.96**	**77.19**	**80.58**	**9.17**

Table 7 shows the detection results on the DS-LSM dataset, the training set of which is composed of two-word domains, and the testing set is composed of three-word and four-word domains. The proposed scheme has a similar detection rate with the deep-learning-based schemes, whereas its false alarm rate is significantly lower than the two schemes. Among existing detection schemes for word-based AGDs, The CNN-based detection can achieve better detection rate than LSTM-based and feature-based scheme, while the average alarm rate is unacceptably high, which is higher than 44%. The average detection rate and false alarm rate of LSTM-based and feature-based scheme are even worse than the CNN-based scheme. For the proposed scheme, the average detection rate of it is higher than 80% and the false alarm rate is lower than 10%. The results demonstrate that the generalization capability of the proposed scheme is significantly stronger than the existing schemes.

6. Conclusions and Future Work

In this study, we concentrate on the detection of word-based AGDs, the features in terms of word frequency, part-of-speech, inter-word correlation, and inter-domain correlations are analyzed. Based on the analysis results, we design a novel framework to detect the word-based DGAs. It uses 24-dimensional features and an ensemble classifier based on Naive Bayes, Extra-Trees, and Logistic Regression to distinguish between the legitimate domains and AGDs. Four types of data sets are used to benchmark the proposed scheme with different classifiers. The datasets are composed of one million legitimate domains, public AGDs, and generated AGDs using WB-DGA. The comparative results with three state-of-the-art DGA detection schemes show that the proposed scheme can achieve significantly better detection performance.

Although the experimental results have proven the effectiveness of the proposed scheme, more features need to be studied to further improve the detection performance. In addition, as we all know that the design of neural networks employed in many deep-learning-based pattern recognition schemes always refers to the analysis approaches in feature engineering. The further analysis of the neural network representation of the employed features is promising to develop a more effective DGA detection scheme, which needs more studies in the future.

Author Contributions: Conceptualization, L.Y. and W.L.; methodology, L.Y. and W.L.; software, L.Y. and H.B.; validation, L.Y., J.Z. and W.L.; formal analysis, L.Y.; data curation, L.Y.; writing—original draft preparation, L.Y.; writing—review and editing, W.L., J.Z., X.J. and G.L.; supervision, G.L. and Y.D.; project administration, G.L. and Y.D.

Funding: This work was supported by The National Natural Science Foundation of China(Grant No. 61702235, U1836104, 61602247, U1636117), Natural Science Foundation of Jiangsu Province (Grant No. BK20160840).

Acknowledgments: Note that the original idea of this study was presented orally at the 2018 International Conference on Cloud Computing and Security(ICCCS 2018), 8–10 June 2018, Haikou, China, The body of this research has been continuously irrigated across a time span, so the authors sincerely thank these conferences for the platform for idea exchanging and thank the anonymous colleagues for the valuable recommendations.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yadav, S.; Reddy, A.K.K.; Reddy, A.N.; Ranjan, S. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. *IEEE Acm Trans. Netw.* **2012**, *20*, 1663–1677. [CrossRef]
2. Sood, A.K.; Zeadally, S. A taxonomy of domain-generation algorithms. *IEEE Secur. Priv.* **2016**, *14*, 46–53. [CrossRef]
3. Fu, Y.; Yu, L.; Hambolu, O.; Ozcelik, I.; Husain, B.; Sun, J.; Sapra, K.; Du, D.; Beasley, C.T.; Brooks, R.R. Stealthy domain generation algorithms. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1430–1443. [CrossRef]
4. Geffner, J. End-to-end analysis of a domain generating algorithm malware family. In Proceedings of the Black Hat USA, Las Vegas, NV, USA, 27 July–1 August 2013.
5. Stanislav, S. Matsnu Malware ID. Check Point Blog Post. 2015. Available online: https://blog.checkpoint.com/wp-content/uploads/2015/07/Matsnu-malwareid-technical-brief.pdf (accessed on 5 May 2016).
6. Woodbridge, J.; Anderson, H.S.; Ahuja, A.; Grant, D. Predicting domain generation algorithms with long short-term memory networks. *arXiv* **2016**, arXiv:1611.00791.
7. Yadav, S.; Reddy, A.K.K.; Reddy, A.L.; Ranjan, S. Detecting algorithmically generated malicious domain names. In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, Melbourne, Australia, 1–3 November 2010; pp. 48–61. [CrossRef]
8. Schiavoni, S.; Maggi, F.; Cavallaro, L.; Zanero, S. Tracking and characterizing botnets using automatically generated domains. *arXiv* **2016**, arXiv:1311.5612.
9. Schiavoni, S.; Maggi, F.; Cavallaro, L.; Zanero, S. Phoenix DGA-based botnet tracking and intelligence. *Detect. Intrusions Malware Vulnerability Assess.* **2014**, *8850*, 192–211._11. [CrossRef]
10. Bilge, L.; Sen, S.; Balzarotti, D.; Kirda, E.; Kruegel, C. Exposure: A passive dns analysis service to detect and report malicious domains. *ACM Trans. Inf. Syst. Secur.* **2014**, *16*, 1–28. [CrossRef]

11. Yu, B.; Gray, D.L.; Pan, J.; De Cock, M.; Nascimento, A.C. Inline dga detection with deep networks. In Proceedings of the IEEE International Conference on Data Mining Work, New Orleans, LA, USA, 18–21 November 2016; pp. 683–692. [CrossRef]
12. Mowbray, M.; Hagen, J. Finding domain-generation algorithms by looking at length distribution. In Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, 3–6 November 2014; pp. 395–400. [CrossRef]
13. Raghuram, J.; Miller, D.J.; Kesidis, G. Unsupervised, low latency anomaly detection of algorithmically generated domain names by generative probabilistic modeling. *J. Adv. Res.* **2014**, *5*, 423–433. [CrossRef] [PubMed]
14. Grill, M.; Nikolaev, I.; Valeros, V.; Rehak, M. Detecting DGA malware using NetFlow. In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 1304–1309. [CrossRef]
15. Nguyen, T.D.; Cao, T.D.; Nguyen, L.G. DGA botnet detection using collaborative filtering and density-based clustering. In Proceedings of the International Symposium on Information and Communication Technology, Hue City, Vietnam, 3–4 December 2015; pp. 203–209. [CrossRef]
16. Wang, T.; Hu, X.; Jang, J.; Ji, S.; Stoecklin, M.; Taylor, T. BotMeter: Charting DGA-botnet landscapes in large networks. In Proceedings of the IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016; pp. 334–343. [CrossRef]
17. Loia, V.; Pedrycz, W.; Senatore, S. Semantic web content analysis: A study in proximity-based collaborative clustering. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 1294–1312. [CrossRef]
18. Cosma, G.; Joy, M. An approach to source-code plagiarism detection and investigation using latent semantic analysis. *IEEE Trans. Comput.* **2012**, *61*, 379–394. [CrossRef]
19. Zupanc, K.; Bosnić, Z. Automated essay evaluation with semantic analysis. *Knowl.-Based Syst.* **2017**, *120*, 118–132. [CrossRef]
20. Harispe, S.; Ranwez, S.; Janaqi, S.; Montmain, J. Semantic similarity from natural language and ontology analysis. *Synth. Lect. Hum. Lang. Technol.* **2015**, *8*. [CrossRef]
21. Bollegala, D.; Matsuo, Y.; Ishizuka, M. A web search engine-based approach to measure semantic similarity between words. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 977–990. [CrossRef]
22. Huang, H.H.; Kuo, Y.H. Cross-lingual document representation and semantic similarity measure: A fuzzy set and rough set based approach. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 1098–1111. [CrossRef]
23. Salton, G.; Wong, A.; Yang, C.S. A vector space model for automatic indexing. *Commun. ACM* **1975**, *18*, 613–620. [CrossRef]
24. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407.:6<391::AID-ASI1>3.0.CO;2-9. [CrossRef]
25. Lund, K.; Burgess, C. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behav. Res. Methods Instrum. Comput.* **1996**, *28*, 203–208. [CrossRef]
26. Dagan, I.; Lee, L.; Pereira, F.C.N. Similarity-based models of word cooccurrence probabilities. *Mach. Learn.* **1999**, *34*, 43–69.:1007537716579. [CrossRef]
27. Bullinaria, J.A.; Levy, J.P. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behav. Res. Methods* **2007**, *39*, 510–526. [CrossRef] [PubMed]
28. Bengio, Y. Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
29. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. In Proceedings of the Workshop at International Conference on Learning Representations, Scottsdale, AZ, USA, 2–4 May 2013.
30. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Neural Information Processing Systems Conference (NIPS), Lake Tahoe, NV, USA, 5–10 December 2013.
31. Mikolov, T.; Yih, W.; Zweig, G. Linguistic regularities in continuous space word representations. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), Atlana, GA, USA, 9–14 June 2013.
32. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

33. Bird, S.; Loper, E. NLTK: The natural language toolkit. In Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions, Barcelona, Spain, 21–26 July 2004. [CrossRef]
34. Chunyu, J.; Yuan, L.; Nanyuan, L. On methods of Chinese automatic segmentation. *J. Chin. Inf. Process.* **1989**, *3*, 3–11.