A Feature Ensemble-based Approach to Malicious Domain Name Identification from Valid DNS Responses

Chen Zhao*†, Yongzheng Zhang*†, Yipeng Wang*†
*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
†Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
Email: {zhaochen, zhangyongzheng, wangyipeng}@iie.ac.cn

***Abstract*—Identifying malicious domain names in Internet activities has become an effective method to protect Internet users. Previous works have achieved great identification results, but they highly rely on historical Domain Name System (DNS) responses and external intelligence sources. Thus, they may fail to identify unknown domain name without any prior knowledge. In this paper, we propose Glacier, a feature ensemble-based approach to identifying malicious domain names from valid DNS responses. Glacier addresses the aforementioned problem by utilizing two types of features in domain name strings: the linguistical features and the statistical features. (1) Linguistical features are vector representations generated from the character sequences of domain names by a bidirectional long short-term memory (BiLSTM) neural network. It is worthy to notice that we modify the last BiLSTM layer to enhance the expressiveness of the linguistical features. (2) Statistical features are six manually designed statistics that represent the structural information of a domain name. Structural information can hardly be learnt by a BiLSTM neural network directly. Thus, combining statistical features with linguistical features can improve the effectiveness of malicious domain name identification. We evaluate the identification ability of Glacier on a real-world domain name data set. The best metrics of Glacier are an average accuracy of 90.86% and an average F1-score of 84.37%. Our experimental results show that Glacier can accurately identify resolvable malicious domain names without any DNS traffic data or prior knowledge about unknown domain names.**

***Index Terms*—Domain Name System, Malicious Domain Name, Neural Language Model, Deep Learning, Cyber Security**

I. Introduction

Malicious domain names are domain names owned by attacker and used in illegal activities, such as phishing, spamming, malwares [1], and botnet-based attacks [2]. Identifying malicious domain names from Domain Name System (DNS) traces is an effective method of protecting Internet users from cyber-attacks. Many previous works have been focused on the identification of malicious domain names, such as Notos [3], Exposure [4], Seguigo [5] and FANCI [6]. However, these works rely on massive historical DNS traffic storage or external intelligence sources, which are not applicable in scenarios without enough resources. Meanwhile, they fail to identify domain names without any historical DNS query or relating intelligence information [7]. To address the aforementioned limitations, we utilize the character compositions of domain names, which have significant differences between malicious domain names and benign domain names.

Our insight is based on the following observation regarding to real-world malicious domain names. In practice, malicious domain names are always at risk of being blocked, so attackers who abuse domain names often need to register domain names in bulk. To keep the domain name registration fee as low as possible and avoid conflicts with existing benign domain names, attackers prefer domain names that are less popular or less expensive. As a result, malicious domain names that are valid and resolvable, are usually unreadable or unnecessarily long.

In this paper, we propose Glacier, a malicious domain name identification method based on the differences in character compositions between malicious domain names and benign domain names. Glacier passively monitors the resolvable DNS traces to classify newly-appeared domain names into malicious or benign. Glacier contains two key features: linguistical features and statistical features. (1) Linguistical features are the vector representations of domain names with a fixed length, they are generated by a character-wised bidirectional long short-term memory (BiLSTM) neural network. It is worthy to notice that we also modify the last BiLSTM layer to enhance the expressiveness of the linguistical features. (2) Statistical features are six manually designed statistics that represent the structural information of a domain name. The structural information can hardly be learnt by a BiLSTM neural network but it is of vital importance. In addition, we combine the statistical features with the linguistical features, Experimental results show that combining statistical features with linguistical features can improve the effectiveness of malicious domain names identification.

Our contributions of this paper are listed as follows:

- We propose Glacier, a novel and light-weighted approach to identifying malicious domain names based on the character composition differences between malicious domain name and benign ones. Glacier requires no massive DNS traffic data or external intelligence sources, and can alert users for possible malicious domain names with a rather low cost.

This work is supported by the National Key Research and Development Program of China (No. Y950121201).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

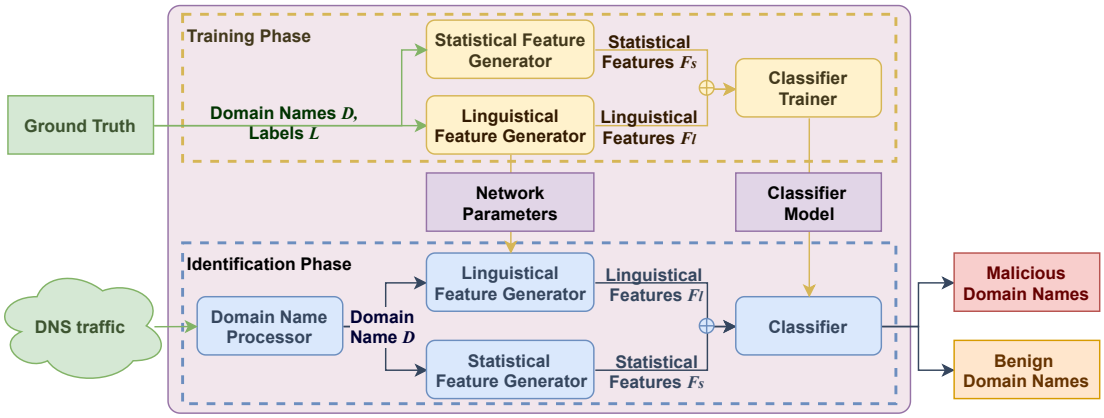

Fig. 1. The architecture of Glacier.

- We propose a feature ensemble method to enhance the identification effectiveness of the neural network. Combing the feature representations generated by the neural network with manual-devised statistical features, Glacier is able to know more information that a neural network can hardly learn directly. In this way, we improve the final identification performance without adding sophisticated network structures or massive learnable parameters.
- We evaluate Glacier on a real-world domain name data set. Glacier achieves an average accuracy of 90.86% and an average F1-score of 84.37%, and exceeds the pure BiLSTM network with relative error reductions of 3.18% in accuracy and 2.74% in F1-score. Experimental results show that Glacier is accurate in identifying resolvable malicious domain names, and the feature ensemble method outperforms the pure neural network method with the same amount of parameters.

The remaining parts of this paper are organized as follows: Section II introduces the composition of Glacier, and describe the key techniques in detail. Section III evaluates the identification ability of Glacier on a real-world domain name data set, and discusses the influences of parameter settings and classifying algorithms on the identifying effectiveness. Section IV briefly reviews the related works on malicious domain name identification. Section V summarizes this paper in the end.

II. Glacier

In this section, we introduce the design of Glacier in detail. First, we describe the components and functional procedure of Glacier. Next, we define the two types of features we use to represent a domain name.

A. The Framework of Glacier

Glacier aims at identifying malicious domain names from valid DNS traffic based on the characters in resolvable domain names themselves. The overall procedure of Glacier is demonstrated in the Fig. 1. Glacier involves two working phases: the training phase and the identification phase. And Glacier employs five key components: the domain name processor, the statistical feature generator, the linguistical feature generator, the classifier, and the classifier trainer.

(1) **The training phase**: Glacier needs training to prepare the parameters and the classifying model before identification online. The input to the training phase is the ground truth that consists of known domain names D and their corresponding labels L. Domain names D are fed to the linguistical feature generator and the statistical generator. Labels L are fed to the linguistical feature generator and the classifier trainer. The linguistical feature generator treats domain names D as sequences of characters, and generates the linguistical features F_l. Meanwhile, the training part of the linguistical feature generator transforms the linguistical features F_l into predictions P, and adjusts the parameters in the neural network. The statistical feature generator takes the domain names D and computes the statistical features F_s. The classifier trainer concatenates the linguistical features F_l and the statistical features F_s, and train the classification model based on labels L. The final outputs of the training phase are the parameters of the linguistical feature generator and the classifying model of the classifier.

(2) **The identification phase**: Glacier starts to work in the identification phase after the parameters and the classifying model are trained. The input to the identification phase is the real-time DNS traffic. The domain name processor extracts unknown domain names D from DNS traces and sends them to the linguistical feature generator and the statistical feature generator. The linguistical feature generator treats domain names D as sequences of characters, and generates the linguistical features F_l based on the parameters trained in the training phase. The statistical feature generator takes the domain names D and computes the statistical features F_s. The classifier trainer concatenates the linguistical features F_l and the statistical features F_s, and classifies unknown domain names as malicious and benign based on the model trained in the training phase.

B. Feature Representation

As mentioned above, Glacier utilizes two types of features to identify malicious domain names: (1) Linguistical features are vector representation of domain names that generated by

the BiLSTM based neural network. (2) Statistical features are manually designed to obtain the structural information of domain names that the neural network cannot learn directly. In this part, we will describe these features in detail.

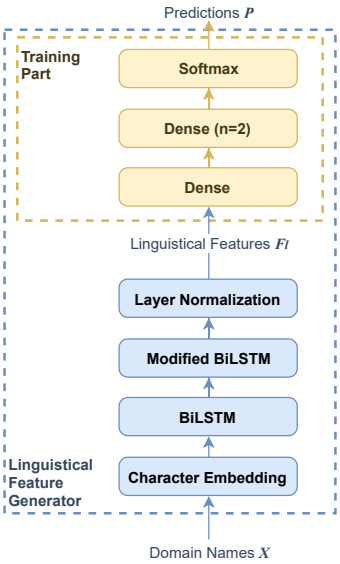

Fig. 2. The structure of the linguistical feature generator.

1) Linguistical Features: Linguistical features are vector representation of domain names that generated by the linguistical features generator. The structure of the linguistical feature generator is shown in Fig 2. The linguistical feature generator contains a character embedding layer, a BiLSTM layer, a modified BiLSTM layer, a layer normalization, two dense layers, and a softmax layer. The two dense layers and the softmax layer compose the training part, and they only take effect in training phase.

Character Embedding: The number of semantic components in a valid domain name are various. For example, domain name *researchgate.net* contains two words, *research* and *gate*; domain name *54569.com* contains no significant semantical component in the left-most label. Thus, it is costly and inefficient to represent domain names in the level of labels or semantical components using methods such as n-gram [8] and word2vec [9]. To represent domain names, we turn to character-wised methods. Note that we have tried one-hot encoding for characters in domain names, but its performance is far worse than that of the character embedding.

Modified BiLSTM layer: It is worth mentioning that we modified the structure of the last BiLSTM layer to enhance the expressiveness of linguistical features. Generally, a BiLSTM layer generates representation with a fixed length for sequences with variable length through discarding the results of time steps except for the last [10]. This method loses the information of previous time steps, For the memorable steps of an LSTM cell is short. On account of this, we design a *multi-to-fixed* operation to replace the discarding. The *multi-to-fixed* operation computes the average value and maximum value of the results generated by an LSTM cell, and retains the result of the last time step. The specific algorithm of the *multi-to-fixed* operation is shown in Algorithm 1.

TABLE I
STATISTICAL FEATURES

#	Definition	db66.cc	hdchina.org
1	length of domain name	7	11
2	number of unique characters	5	11
3	changes between vowels and consonants	0	4
4	information entropy	0.4261	0.6246
5	ratio of numerical characters	28.57%	0
6	ratio of consonant characters	100%	70%

Algorithm 1 Multi-to-fixed Operation

Input: The result of last LSTM layer $\boldsymbol{H}$, $\boldsymbol{H} \subset \mathbb{R}^{n \times s}$, n is the length of sequence from upper layer, s is the number of cells in last LSTM layer.

$i = 1$;
for $i <= n$ **do**
  $a_i = H_{i,s}$;
  $\hat{h}_i = \text{Maximum}(H_{i,1}, H_{i,2}, \cdots, H_{i,s})$;
  $\bar{h}_i = \text{Average}(H_{i,1}, H_{i,2}, \cdots, H_{i,s})$;
  $\boldsymbol{f}_i = \{a_i, \hat{h}_i, \bar{h}_i\}$;
  $i = i + 1$;
end for
$\boldsymbol{f} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n\}$
$\boldsymbol{f'} = \text{LayerNormalize}(\boldsymbol{f})$
return $\boldsymbol{f'}$

2) Statistical Features: Our neural network for linguistical features processes data sequentially, and it can hardly learn the structural information in a sequence. However, in the identification of malicious domain names, the structural information, such as the length, and the ratio of digital, has a significant importance. To address this deficiency, we manually design six statistical features to summarize the structural information of domain names. The definitions of statistical features are listed in Table I with two examples. The *db66.cc* is a malicious domain name, and its statistical features are {7, 5, 0, 0.4261, 0.2857, 1.00}. The *hdchina.org* is a benign domain name, and its statistical features are {11, 11, 4, 0.6246, 0.00, 0.70}.

III. EXPERIMENTS

In this section, we will evaluate the effectiveness of Glacier by on real-world domain names. For better understanding, we first introduce the data set and experiment settings, define the metrics we use for evaluation. Then we describe the key parameters of Glacier, and demonstrate the experimental results of different parameter settings. Finally, we present the experimental results of different classification algorithms.

A. Experiment Setup

In this part, we will give the basic knowledge of our experiments. We first introduce how we build our real-world data set, then state the experimental settings and finally defines the evaluation metrics used in following experiments.

1) Data Set:

*a) **Data Collection**:* In this paper, we collect DNS responses on a recursive DNS server of a major ISP (Internet Service Provider) from March 2018 to May 2018, and finally we get $2,724,393$ records for valid domain names. We concern about the identification of unknown malicious domain names in the valid DNS traffic, and thus we need to remove two kinds of domain names that are out of the scope of this paper, (1) domain names that don't have any valid resolution IP address; and (2) domain names that are very popular and whose reputation information can be easily verified. In addition, it is also worthy to notice that for the popular domain names we refer to the Alexa Top Global Site list to remove these domain names.

*b) **Ground Truth**:* To evaluate the effectiveness of our proposed approach, we first need to give the DNS data we captured a right label, *i.e.,* benign or malicious. The specific labeling process is as follows. First, we collect some public black lists from the *malwaredomainlist.com* and *malwaredomains.com*, and label the domain names that appear in these black list as malicious domain names. Then, we query the inspection API of *VirusTotal*, a public security intelligence service, for the risk of the rest domain names. We label domain names that *VirusTotal* reports serving legal services as benign domain names; we label domain names reports to have involved in network attacks as malicious domain names. In this way, we get $7,179$ malicious domain names and $15,076$ benign domain names, $22,255$ in total.

2) Experiment Settings: We evaluate the Glacier Net on a GeForce RTX 2080 GPU with Python 3.7.0 and PyTorch 1.2.0. To eliminate the influence of random initialization of the embedding layer, we pre-train several groups of embedding weights for every embedding size to initialize the embedding layers in the following experiments. In the experiments of every parameter setting and classifier setting of Glacier, we apply 5-fold cross validation and train 512 epoches for each fold. And the experimental result of a parameter setting is consisted of the average value and the standard deviation value of the 5 group metrics generated by 5-fold cross validation.

3) Metrics: We define four data sets from identification results for evaluation:

- True positives (TP): the set of domain names that are parsed as malicious by Glacier and are indeed malicious domain names.
- False positives (FP): the set of domain name that are parsed as malicious by Glacier but are actually benign domain names.
- True negatives (TN): the set of domain names that are parsed as benign by Glacier and are indeed benign domain names.
- False negatives (FN): the set of domain name that are parsed as benign by Glacier but are actually malicious domain names.

Based on these four groups, we define metrics to evaluate the performance of Glacier:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Relative Error Reduction} = 1 - \frac{1 - \text{Metric}_{\text{new}}}{1 - \text{Metric}_{\text{old}}}$$

Note that we use the relative error reduction to assess the improvement of metric after imposing statistical features. The $\text{Metric}_{\text{old}}$ denotes the metric value before applying the machine-learning classifier with statistical features; and the $\text{Metric}_{\text{new}}$ denotes the metric value after applying the classifier with statistical features.

B. Parameter Selection

In this part, we will evaluate the identification ability of Glacier on the real-world domain name data set described in Section III-A1. We first introduce the key parameters of Glacier and their influences of identification results. Then, we demonstrate the experimental results of different parameter settings and discuss the influence of parameter settings on the identification ability.

1) Parameter Definition: The linguistical feature generator in Glacier involves following five parameters: 1) parameter E: the output size of the character embedding layer; 2) parameter H: the number of LSTM cells in one direction of each BiLSTM layer; 3) parameter N: the number of BiLSTM layers (including the modified BiLSTM layer); 4) parameter D: the number of neurons in the first dense layer.

- The output size of the character embedding layer (E): The embedding layer of Glacier transforms each character in a domain name into a vector of length E. The latent amount of information in embedding vectors will increase as E increase. However, setting E too large will increase the computational cost of network while brings little improvement in the expressiveness of embedding vectors. In the following evaluations, we vary the range $E \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$.
- The number of LSTM cells in one direction of each BiLSTM layer (H): An LSTM layer uses multiple cells to process a sequence at the same time. The weights and biases of cells are different, so that different cells can learn different information. Larger H means more cells and more perspectives of information. But it also increases the computational cost and causes over-fitting when H is excessive. In the following evaluations, we vary the range $H \in \{8, 16, 24, 32, 40, 48, 56, 64\}$.
- The number of BiLSTM layers (N): LSTM structure can stack multiple layers together. In a multi-layer LSTM

structure, latter layers take the output sequences of formal layers as inputs. Adding LSTM layers helps network learn more complex reliance in sequences. But it also increases the computational cost and causes the gradient disappearance when N is excessive. In the following evaluations, we vary the range $N \in \{1, 2, 3, 4, 5, 6, 7, 8\}$.

- The number of neurons in the dense layer before the predicting layer (D): Glacier trains the network relying on the prediction of two Dense layers. The output size of last layer is the number of categories, and the output size of the intermediate dense layer (D) influences the prediction result. The intermediate layer integrates the content from the previous layer and prepares for prediction, as a result, the output size D should be smaller than the input size of the intermediate layer. In Glacier, we empirically set $D = (H \times 2 \times 3)/4$, which means D varies implicitly with the parameter H.

2) Experimental Results: We evaluate the quality of Glacier using the domain names from the real-world DNS traffic in terms of accuracy and F1-score. The experimental results of different parameter compositions are demonstrated in Table II–Table IV. Each Table contains three parts: (1) The evaluation metrics of predictions generated by the network using linguistical features only. (2) The evaluation metrics of predictions generated by the Random Forest classifier using both linguistical features and statistical features. (3) The relative error reduction (RER) values of accuracy and F1-score after applying the Random Forest classifier with statistical features. Each metric in table consists of the average value and the standard variation of the results generated in the 5-fold cross validation. The best metrics of each part are marked in bold. Note that the classification result of statistical features using the random forest classifier is noted under each table. The accuracy value of statistical features is 81.55%, and the F1-score value of statistical features is 68.24%.

Table II shows the experimental results of the embedding size E. When varying the embedding size E, we fix the other two parameters to $H = 32$, $N = 2$. The average accuracy values vary in the range of 89.84% – 90.86% for different parameter settings. The average F1-score values vary in the range of 82.58% – 84.37% for different parameter settings. The best accuracy value is 90.86%, achieved by the combination of statistical features and linguistical features with $E = 70$. The best F1-score value is 84.37%, achieved by the combination of statistical features and linguistical features with $E = 70$. The highest RER value of accuracy is 3.56% where $E = 20$, and the highest RER value of F1-score is 3.27% where $E = 10$.

Table III shows the experimental results of the hidden unit sizes H. When varying the number of hidden units H, we fix the other two parameters to $E = 40$, $N = 2$. The average accuracy values vary in the range of 90.06% – 90.83% for different parameter settings. The average F1-score values vary in the range of 83.10% – 84.36% for different parameter settings. The best accuracy value is 90.83% that achieved by the combination of statistical features and linguistical features with $H = 44$ and $H = 48$. The best F1-score value is 84.36% that achieved by the combination of statistical features and linguistical features with $H = 48$. The highest RER value of accuracy is 3.37% where $H = 48$, and the highest RER value of F1-score is 3.02% where $H = 24$.

Table IV shows the experimental results of BiLSTM layer numbers N. When varying the embedding size N, the other two parameters are fixed to $E = 40$, $H = 32$. The average accuracy values vary in the range of 69.35% – 90.76% for different parameter settings. The average F1-score values vary in the range of 45.74% – 84.25% for different parameter settings. The best accuracy value is 90.76% that achieved by the combination of statistical features and linguistical features with $N = 2$. The best F1-score value is 84.25% that achieved by the combination of statistical features and linguistical features with $N = 2$. The highest RER value of accuracy is 35.92% where $N = 8$, and the highest RER value of F1-score is 29.01% where $N = 8$.

3) Discussion: From the experimental results, we have following observations:

- The accuracy values and F1-score values increase as the embedding size E increase when $E \in \{5, 10, 20\}$. And the metric values no longer vary orderly as the embedding size E keeps increasing after $E > 16$. Which denotes that increasing the embedding size does not have significant benefits on the detection performance.
- The accuracy values and F1-score values increase as the hidden unit size H increase when $H \in \{8, 16\}$. And the metric values no longer vary orderly as the hidden unit size H keeps increasing after $H > 16$.
- The accuracy values and F1-score values increase as the layer number N increase when $N \in \{1, 2\}$. And the accuracy values and F1-score values increase as the layer number N keeps increasing when $N > 2$. Note that the accuracy values and F1-score values are rather low when $N \in \{7, 8\}$, which implicit that layer number over 6 is too high and causes the gradient disappearance.
- RER values that measure the improvement of feature ensemble method to linguistical features are always positive, and the best metrics of all parameters are achieved by the combination of statistical features and linguistical features. The experimental result show that this feature ensemble-based method can surly improve the effectiveness of malicious domain name identification.

C. Classifier Comparison

In this part, we assess the algorithm used in the classifier using the combination of linguistical features and statistical features. Note that the parameters of the linguistical feature generator are fixed to $E = 40$, $H = 32$, and $n = 2$ when generating the linguistical features used in classification.

1) Classifier Introduction: In this parts, we test five following classifiers: the Decision Tree classifier, the AdaBoost classifier, the Bagging classifier, the Extra Trees classifier, and the Random Forest classifier. The decision tree classifier is a basic machine-learning algorithm that fits data through

TABLE II
EXPERIMENTAL RESULTS OF DIFFERENT EMBEDDING SIZES (E)

Embed	Linguistical Features		Statistical + Linguistical Features		Relative Error Reduction	
	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)
5	89.84± 0.62	82.58± 1.12	90.12± 0.49	82.87± 0.86	2.76	1.66
10	90.33± 0.35	83.50± 0.75	90.67± 0.46	84.04± 0.81	3.52	**3.27**
20	90.45± 0.23	83.81± 0.35	90.79± 0.13	84.25± 0.26	**3.56**	2.72
30	90.54± 0.20	83.87± 0.26	90.63± 0.12	84.00± 0.27	0.95	0.81
40	90.51± 0.30	83.79± 0.59	90.76± 0.48	84.25± 0.83	2.63	2.84
50	90.46± 0.10	83.72± 0.27	90.69± 0.26	84.12± 0.42	2.41	2.46
60	90.55± 0.27	83.95± 0.46	90.74± 0.29	84.20± 0.45	2.01	1.56
70	90.56± 0.15	83.93± 0.27	**90.86**± 0.18	**84.37**± 0.25	3.18	2.74
80	**90.74**± 0.76	**84.21**± 1.38	90.81± 0.62	84.32± 1.04	0.76	0.70
90	90.62± 0.12	83.99± 0.27	90.79± 0.15	84.33± 0.25	1.81	2.12
100	90.51± 0.30	83.84± 0.47	90.79± 0.26	84.32± 0.38	2.95	2.97

*Results of statistical features: Accuracy 81.55 ± 0.19, F1-score 68.24 ± 0.25.

TABLE III
EXPERIMENTAL RESULTS OF DIFFERENT HIDDEN UNIT SIZES (H)

Hidden	Linguistical Features		Statistical + Linguistical Features		Relative Error Reduction	
	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)
8	90.06± 0.17	83.10± 0.33	90.33± 0.30	83.47± 0.56	2.72	2.19
16	**90.62**± 0.23	**83.95**± 0.36	90.77± 0.34	84.20± 0.59	1.60	1.56
24	90.32± 0.26	83.46± 0.39	90.62± 0.30	83.96± 0.51	3.10	**3.02**
32	90.51± 0.30	83.79± 0.59	90.76± 0.48	84.25± 0.83	2.63	2.84
40	90.29± 0.20	83.46± 0.38	90.57± 0.19	83.94± 0.34	2.88	2.90
48	90.51± 0.10	83.92± 0.24	**90.83**± 0.20	**84.36**± 0.33	**3.37**	2.74
56	90.44± 0.30	83.74± 0.62	90.50± 0.37	83.80± 0.59	0.63	0.37
64	90.39± 0.18	83.63± 0.43	90.59± 0.44	83.91± 0.81	2.08	1.71

*Results of statistical features: Accuracy 81.55 ± 0.19, F1-score 68.24 ± 0.25.

TABLE IV
EXPERIMENTAL RESULTS OF DIFFERENT LAYER NUMBERS (N)

Layer	Linguistical Features		Statistical + Linguistical Features		Relative Error Reduction	
	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)
1	90.40± 0.35	83.69± 0.63	90.70± 0.24	83.85± 0.45	3.12	0.98
2	**90.51**± 0.30	**83.79**± 0.59	**90.76**± 0.48	**84.25**± 0.83	2.63	2.84
3	90.38± 0.28	83.66± 0.55	90.61± 0.39	84.03± 0.67	2.39	2.26
4	90.37± 0.27	83.64± 0.44	90.46± 0.15	83.79± 0.23	0.93	0.92
5	89.82± 0.23	82.58± 0.78	90.09± 0.35	82.96± 0.89	2.65	2.18
6	89.97± 0.45	82.82± 0.92	90.09± 0.64	83.01± 1.17	1.20	1.11
7	76.86± 9.19	59.88±15.67	83.53± 4.00	68.51± 8.97	28.82	21.51
8	69.35± 0.60	45.74± 4.94	80.36± 0.40	61.48± 0.88	**35.92**	**29.01**

*Results of statistical features: Accuracy 81.55% ± 0.19, F1-score 68.24% ± 0.25.

building tree-like structures. The other four classifiers are ensemble algorithms that integrate multiple base classifiers use different ensemble strategies, In the experiments, we use the decision tree as the their base classifiers of all the four classifiers. Meanwhile, we also show the predicting result of the linguistical feature generator with the training part for comparison.

2) Experimental Results: The average accuracy values vary in the range of 84.94% – 90.80% for different classifying algorithms. The average F1-score values vary in the range of 76.82% – 84.34% for different classifying algorithms. The best accuracy value is 90.76% that achieved by the random forest classifier. The best F1-score value is 84.25% that achieved by the random forest classifier. The highest RER value of accuracy is 3.06%, and the highest RER value of F1-score is 3.39%, both achieved by the random forest classifier.

3) Discussion: The decision tree classifier and the AdaBoost classier perform worse than the linguistical feature generator in both accuracy and F1-score. The bagging classifier performs worse than the linguistical feature generator in accuracy and outperforms the linguistical feature generator in F1-score. The random forest classifier and the extra tree classier outperform the linguistical feature generator in both accuracy and F1-score. Experimental results show that the ensemble strategies of the bagging classifier, the random forest classifier, and the extra tree classifier are capable to improve the classification abilities.

IV. RELATED WORKS

Researches on malicious domain names identification mainly fall into two categories: feature-based methods and relation-based methods.

TABLE V
EXPERIMENTAL RESULTS OF DIFFERENT CLASSIFIERS

Algorithm	Classifier Results		Relative Error Reduction	
	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)
Linguistical Feature Generator	90.51± 0.30	83.79± 0.59	–	–
Decision Tree Classifier	84.94± 0.48	76.82± 0.64	-58.69	-43.00
AdaBoost Classifier	89.92± 0.26	83.20± 0.48	-6.22	-3.64
Bagging Classifier	90.49± 0.51	83.97± 0.86	-0.21	1.11
Random Forest Classifier	**90.80**± 0.45	**84.34**± 0.76	**3.06**	**3.39**
Extra Trees Classifier	90.76± 0.48	84.25± 0.83	2.63	2.84

Most relation-based methods compute reputation scores for unknown domain names basing on graphs of domain names and other revelent identities. B. Rahbarinia et al. [5] present Segugio, a malicious domain names identification system based on a bipartite graph between clients and queried domain names. Segugio attributes nodes of domain names with domain activity features and IP abuse features basing on the bipartite graph and classifies. I. Khalil et al. [11] develop a malicious domain names inferences system based on a weighted graph of domain names to spread reputation scores from known domain names. The nodes in the graph represent domain names, and the weight between two nodes is determined by how many resolved IP addresses they share. H. Tran et al. [12] propose a detection approach based on the graph of domain names and IP addresses. They use a belief propagation algorithm to compute the malicious scores of domain names.

Most feature-based methods extract features that characterize the differences between malicious domain names and benign domain names from DNS traffic. M. Antonakakis et al. [3] propose Notos, a system monitors DNS traffic between clients and recursive resolvers. It assigns reputation scores to domain names according to features about the IP addresses, related domain names, and the appearances of the target domain name and its IP addresses in malicious samples and black lists. L. Bilge et al. [13] propose Exposure, a behavior-based malicious domain name identification system which using time-based features, DNS answer-based features, TTL value-based features, and domain name-based features to profile malicious samples. M. Antonakakis et al. [14] present another system, Kopis which works on the level above recursive resolvers, it detects malicious domain names using three types of statistical features: requester diversity, requester profile, and resolved-IPs reputation. M. Weber et al. [15] design a method to expand existing public black lists using clustering methods based on features relating to resolved IP addresses, registration information and the appearances in DNS traces of domain names.

V. CONCLUSION

In this paper, we propose Glacier, a novel approach to malicious domain names identification based on feature ensemble. Glacier utilizes two features extract from domain names strings: linguistical features that represent character compositions in domain names, and statistical features that represent the structural information of domain names. Glacier modifies the BiLSTM neural networks to enhance the expressiveness of linguistical features. And Glacier uses the statistical features to compensate the structural information that the linguistical features can not learn. Our experimental results show that Glacier can accurately identify malicious domain names in the real-world DNS responses.

REFERENCES

[1] C. Peng, X. Yun, Y. Zhang, and S. Li, "Malshoot: Shooting malicious domains through graph embedding on passive DNS data," in *Collaborative Computing: Networking, Applications and Worksharing*. Cham: Springer International Publishing, 2019, pp. 488–503.

[2] X. Yun, J. Huang, Y. Wang, T. Zang, Y. Zhou, and Y. Zhang, "Khaos: An adversarial neural network DGA with high anti-detection ability," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019.

[3] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS." in *USENIX security symposium*, 2010, pp. 273–290.

[4] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Transactions on Information and System Security (TISSEC)*, vol. 16, p. 14, 2014.

[5] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient Behavior-Based Tracking of Malware-Control Domains in Large IS&P Networks," in *Proceedings of the International Conference on Dependable Systems and Networks*, vol. 2015-Septe, 2015, pp. 403–414.

[6] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "Fanci: Feature-based automated NXDomain classification and intelligence," in *27th USENIX Security Symposium*, 2018, pp. 1165–1181.

[7] C. Lever, P. Kotzias, D. Balzarotti, J. Caballero, and M. Antonakakis, "A lustrum of malware network communication: Evolution and insights," in *2017 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2017, pp. 788–804.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *CoRR*, vol. abs/1607.04606, 2016.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.

[10] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[11] I. Khalil, T. Yu, and G. Bei, "Discovering malicious domains through passive DNS data graph analysis," in *ACM on Asia Conference on Computer& Communications Security*, 2016.

[12] H. Tran, A. Nguyen, P. Vo, and T. Vu, "DNS graph mining for malicious domain detection," *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. 2018-Janua, pp. 4680–4685, 2018.

[13] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive DNS analysis." in *NDSS*, 2011, pp. 1–17.

[14] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon, "Detecting malware domains at the upper DNS hierarchy." in *USENIX security symposium*, vol. 11, 2011, pp. 1–16.

[15] M. Weber, J. Wang, and Y. Zhou, "Unsupervised Clustering for Identification of Malicious Domain Campaigns," in *Proceedings of the First Workshop on Radical and Experiential Security - RESEC '18*. ACM Press, 2018, pp. 33–39.