DGA-Based Botnet Detection Toward Imbalanced Multiclass Learning

Yijing Chen, Bo Pang, Guolin Shao*, Guozhu Wen, and Xingshu Chen

Abstract: Botnets based on the Domain Generation Algorithm (DGA) mechanism pose great challenges to the main current detection methods because of their strong concealment and robustness. However, the complexity of the DGA family and the imbalance of samples continue to impede research on DGA detection. In the existing work, the sample size of each DGA family is regarded as the most important determinant of the resampling proportion; thus, differences in the characteristics of various samples are ignored, and the optimal resampling effect is not achieved. In this paper, a Long Short-Term Memory-based Property and Quantity Dependent Optimization (LSTM.PQDO) method is proposed. This method takes advantage of LSTM to automatically mine the comprehensive features of DGA domain names. It iterates the resampling proportion with the optimal solution based on a comprehensive consideration of the original number and characteristics of the samples to heuristically search for a better solution around the initial solution in the right direction; thus, dynamic optimization of the resampling proportion is realized. The experimental results show that the LSTM.PQDO method can achieve better performance compared with existing models to overcome the difficulties of unbalanced datasets; moreover, it can function as a reference for sample resampling tasks in similar scenarios.

Key words: botnet; Domain Generation Algorithm (DGA); multiclass imbalance; resampling

1 Introduction

The Internet has been integrated into nearly all fields of economy, society, and life. The rapid development of the Internet has brought about much convenience to daily life, but also poses a threat to network security. Various malicious network attacks can emerge in an endless stream; among these attacks, botnet-based attacks represent a main threat[1]. A botnet consists of a series of hosts infected by malware. The master machine controls the infected host remotely via a Command and Control (C&C) server and executes a series of malicious activities. Based on the Domain Generation Algorithm (DGA) mechanism, botnets can make use of the characteristics of the algorithm of generating the pseudo-random string and the way of re-selecting the connected host by using botnet, which greatly improves the concealment and robustness of botnet and brings great challenges to botnet detection. Given this perspective, studying a high-accuracy and low-cost detection scheme for DGA domain names is of great significance for network security.

Traditional solutions for DGA domain name detection are mainly based on artificial feature extraction of Domain Name Server (DNS) traffic or domain name language statistical characteristics. Machine learning is then used to analyze the extracted features and complete the identification and classification of DGA domain names. However, determining the appropriate DGA type is a complex process. A DGA family usually corresponds to a group of similar DGA algorithms,
and different types of DGA have different DNS traffic and domain name language statistical characteristics. Therefore, detection schemes based on artificial feature extraction entail high cost, and the extracted features are not flexible; thus, these schemes cannot cope with the complexity of DGA types. Thus, the construction of a DGA detection model based on deep learning has attracted research interest as an improved detection method compared with previous approaches. Existing studies seldom pay attention to the imbalance of datasets. Unfortunately, the imbalance of sample size in different classes of datasets will lead to great reductions in the performance of classifiers. At present, processing methods for imbalanced datasets are mainly based on the data and algorithm levels. At the data level, the imbalanced samples are usually augmented or removed by resampling. At the algorithm level, the optimal classification algorithm is usually used to overcome the impact of sample number on the classification results. However, while data- and algorithm-based optimization take the original number of samples as the most important determinant, our research shows that different classes of samples have different internal properties. If the original sample size is regarded as the only determinant, the characteristics of different classes are ignored, and the optimal sampling proportion cannot be determined. Therefore, improved methods based on the sample size alone have some limitations. We propose a Long Short-Term Memory-based Property and Quantity Dependent Optimization (LSTM.PQDO) detection model based on the deep learning algorithm. Considering the high detection cost of traditional detection schemes based on artificial feature extraction, we make full use of the advantages of the deep learning LSTM algorithm in the context of understanding domain name detection and realize the automatic extraction of DGA domain name features. In view of the imbalance of DGA classes, we comprehensively consider the original number and characteristics of all types of original DGA samples. By iterating the sampling proportion, the optimal solution is searched heuristically in the right direction and around the optimal solution to realize the dynamic optimization of sampling proportion. We not only take advantage of the traditional method of optimizing sampling proportion based on the number of original samples but also take into full account the impact of different classes of samples to minimize the impact of imbalanced datasets on classification results and further optimize the detection effect of DGA domain names. The main contributions of this paper are as follows:

1. A DGA detection model based on LSTM is proposed and implemented. This model can automatically mine the comprehensive characteristics of DGA domain names, maximize the coverage of known and potential linguistic statistical characteristics, and reduce the cost of DGA detection.

2. An LSTM.PQDO method is proposed to address the imbalance of sample size in DGA detection. The method comprehensively considers the original number of samples and the characteristics of various types of samples. Moreover, based on the mature and correct principles provided in the existing research, the characteristics of different classes of samples are fully considered to achieve maximum optimization of the sampling of imbalanced data and improve the DGA classification effect. The experimental results show that the LSTM.PQDO optimization method has better performance for DGA detection than the current methods. Compared with those of unoptimized LSTM algorithms, the Macro-AVG F1 score obtained from the proposed algorithm is improved by 13.94%.

2 Related Work

The traditional DGA domain name detection scheme is mainly based on the artificial feature extraction of DNS query behavior or domain name language statistical characteristics, and the machine learning method is used to analyze the extracted characteristics to complete the classification or clustering of DGA domain names. Zhou et al.\(^2\) firstly extracted 18 feature sets for domain name diversity, timeliness, growth, and relevance based on the characteristic analysis of DNS query behavior by collecting domain name access records in real-world environments and then constructed a model based on the random forest algorithm to detect the DGA fast-changing domain name. Chang and Lin\(^3\) proposed a dynamic differential botnet detection method based on DNS traffic monitoring; here, DNS was firstly filtered to delete known normal and malicious domain names, after which the Chinese-Whispers algorithm was used to cluster the remaining domains according to the similarity of the query behavior. Kwon et al.\(^4\) used signal processing and spectral density testing technologies to find the frequency of botnet periodic DNS queries. Yadav et al.\(^5\) studied the distribution of alphanumeric characters and bigrams mapped to all domains of the same set of IP addresses to analyze domain name
language statistical characteristics and compared the performance of several distance measures. Schiavoni et al.\cite{6} introduced the Phoenix mechanism to detect DGA domain names based on the combination of domain name language statistical and IP characteristics. Truong and Cheng\cite{7} analyzed a large number of legal domains and DGA domain names and found obvious deviations in domain name construction rules. The length of domain names and character information entropy were used as classification features to detect DGA domain names. Tong and Nguyen\cite{8} detected DGA domain names by using semantic indicators, such as entropy, domain level, frequency of N-gram, and the Mahalanobis distance of domain classification. The representative solution at the data level is resampling, and three types of resampling are identified, namely, oversampling, undersampling, and hybrid sampling based on over- and undersampling. Mathew et al.\cite{9} proposed a kernel-based KSMOTE algorithm to generate a few data points directly in the feature space of support vector machine classifiers. Lin et al.\cite{10} proposed an undersampling CBUS algorithm based on clustering of most classes to equate the number of clusters in most classes with the number of clusters in a few classes. Ha and Lee\cite{11} proposed the GAUS algorithm to maximize the performance of a prototype classifier such that the prototypes minimized the loss between distributions of original and undersampled majority objects. Gazzah et al.\cite{12} proposed a hybrid sampling method that used an SMOTE star topology to oversample a few classes and undersample most classes by eliminating irrelevant examples. The experimental results showed that the performance of the classifier optimized by hybrid sampling was better than that of the single sampling method.

Tran et al.\cite{13} proposed an LSTM.MI algorithm based on binary and multiclassification models to improve the cost-effectiveness of optimization at the algorithm level. Cost items were introduced to the back propagation learning process to consider the importance of classification recognition. Chen et al.\cite{14} proposed a Quantity Dependent Back Propagation (QDBP) algorithm that takes different weights for each class to calculate the cost function according to the number difference of each class and combines differences between groups to overcome the difficulty of data imbalance to a certain extent.

In the field of DGA detection, current detection methods based on artificial feature extraction of DNS query behavior or domain name language statistical characteristics are challenged by high detection cost and inflexibility; thus, they cannot cope with the complexity of DGA. Most of the current methods based on data- or algorithm-level optimization to classify imbalanced dataset regard the original number of samples as the most important determinant of sampling proportion, which leads to neglect of other similarly important factors, such as the characteristics of various samples. Therefore, the existing optimization methods have some limitations. We propose an LSTM.PQDO optimization method to solve these problems and make full use of the advantages of an LSTM deep learning model to automatically extract characteristics. This method reduces the cost of DGA detection and realizes dynamic optimization of sampling proportions based on a comprehensive consideration of the number of the original samples and the characteristics of various samples.

3 DGA Detection Based on LSTM

3.1 Problem background

DGA is often used for communication between botnets and their C&C servers. Botnets based on DGA have strong concealment and robustness. Because the types of DGA are complex, significant technical challenges to the effective detection of botnets are often encountered. The core concept of traditional DGA detection schemes is extraction of artificial features from DNS traffic or DGA domain name language statistical characteristics. Machine learning is then used to classify or cluster the extracted features. Due to the complexity of the DGA, the DNS traffic or domain name language statistical characteristics that can be effectively learned by the detection model vary greatly according to different DGA categories, especially when the DGA produces variants or the attacker deliberately circumvents the extracted features. Given its ability to effectively detect DGA, artificial feature extraction requires high detection costs, and the extracted features are not flexible. A detection model based on deep learning can effectively overcome the defects of the above traditional artificial feature extraction detection scheme, and deep learning algorithm can realize the automatic extraction of features, which not only avoids the high detection cost of feature engineering but also has higher classification accuracy. Therefore, construction of DGA detection models based on deep learning has received extensive attention. For example, studies of the detection model were conducted based on deep learning in Refs. [15–
We propose a method for DGA detection based on the LSTM algorithm. LSTM performs well with text in long-term learning mode and has been successfully applied in text and language fields, such as handwriting recognition, sequence generation, machine translation, and video analysis\[20\]. Here, the advantages of LSTM in understanding the context of domain names and analyzing the spelling and random characteristics of domain names are utilized to detect DGA domain names.

Detection methods based on deep learning require the support of balanced datasets. The existing research rarely pays attention to optimization of imbalanced datasets. However, the existing DGA datasets have extremely imbalanced samples, which could seriously reduce the performance of the classifier. At present, optimization of imbalanced datasets is mainly conducted at the data and algorithm levels. At the data level, the resampling algorithm is usually used to adjust the number of imbalanced samples. The numbers of classes with more and less samples are decreased and increased, respectively. At the algorithm level, an optimization classification algorithm is used to overcome the impact of sample size on the classification result. According to our research, different classes of samples exhibit different complexities, intra-class distances, \( N \)-gram frequencies, domain name character lengths, transliteration abilities, and other properties; thus, the characteristics of each sample are also the main factors determining the sampling ratio. For example, because samples of certain classes are highly complex, these classes also require more samples for learning, even if the sample size is large. Greatly reducing the sampling ratio is not ideal. Some classes do not have the ability to transliterate, and the combination of vowels and numbers is chaotic, which means the sampling scale must be appropriately increased for these classes. However, most methods based on data- or algorithm-level optimization take the original number of samples as the most important determinant of the sampling proportion, which leads to neglect of the characteristics of various samples. Therefore, the method of determining sampling proportions entirely based on the original number of samples or improvement of the classification algorithm presents some limitations and optimization space.

In view of the shortcomings of the traditional optimization methods described above, a dynamic optimization method of sampling proportion based on comprehensive consideration of the original number and characteristics of all types of samples in DGA is proposed in this paper. In this method, the optimal solution based on the traditional optimization method is used as the initial iteration value, and the sampling ratio is iterated in the right direction and around the optimal solution so that the optimal solution is heuristically searched by the optimization algorithm. This method not only gives full play to the advantages of the traditional optimization method based on the original number of samples but also takes into account the impact of the characteristics of different classes of samples to further reduce the impact of imbalanced datasets.

3.2 DGA detection approach based on LSTM

Because DGA domain names are generated randomly, some linguistic statistical characteristics, such as entropy, vowel and consonant letter distribution, and \( N \)-gram frequency, can be recognized. Compared with legal domain names, most DGA domain names have higher entropy, fewer vowel letters, and lower \( N \)-gram frequency. However, one or a few features are inadequate to support the final decision-making of DGA domain names. LSTM neural networks have the ability to automatically mine and store DGA domain name characters and maximize the coverage of known and potential linguistic statistical features. Therefore, this type of network has attracted wide attention as an effective and low-cost detection method.

We implement a DGA domain name detection model based on a deep neural LSTM network. The actual process of detection is shown in Fig. 1. The model consists of an embedding layer, an LSTM layer, and a softmax layer. The embedding layer transforms DGA domain name vectors encoded by One-Hot into the \( 30 \times 128 \) dimension input vectors. The LSTM layer, as the core of the training model, learns the spelling and random characteristics of domain names from the DGA domain name samples. In the softmax layer, the output value of the DGA domain name classification is transformed into a relative probability, and the maximum bit of probability is selected as the output classification result to complete the multi-classification of DGA domain name classes. One-Hot is used to pre-process DGA domain names and is suitable for dealing with discontinuous values, such as DGA domain names. To some extent, it extends the features of DGA domain names to reasonably improve the distance calculation between DGA domain, thus solving the problem that classifiers are inadequate for dealing with attribute data. The first 30 bits of a DGA domain name are intercepted.
by One-Hot coding. According to statistics, the number of characters in a DGA domain name character set is 38. Therefore, 38-bit state registers are used to encode 38 states of domain name characters (a, . . . , z, 0, . . . , 9, -) to form a 30×38-dimension input vector $V_1$. Each state has its own register bit, and only one valid bit is always used.

The cell dimension of LSTM can be adjusted by adding a full connection layer after One-Hot coding layer; this step increases the dimensions of LSTM and enhances the complexity of its structure, thereby maximizing the features of DGA domain names that it can capture. After adjustment, the input vector $V_2$ of LSTM with dimensions of 30×128 can be obtained.

The extracted feature vector matrix of DGA domain names is input into the LSTM structure with a time sequence of 30 to capture the spelling and random features of these names. In each time sequence, the output of the DGA domain name is controlled by forgetting, input, and output gates to achieve selective deletion and increase the information of the DGA domain name. Forgetting gate $f_t$ is responsible for selectively forgetting information; it receives the previous input $h_{t-1}$ and current input $x_t$ according to Eq. (1) and outputs a value between 0 and 1 for each value in the previous state (0 for complete forgetting and 1 for complete reservation).

$$f_t = \sigma(f_t \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

where $\sigma$ is the activation function of logistics and $b_f$ is the offset vector of the input gate. The input gate $i_t$ selectively records the new information into the cell state according to Eqs. (2) and (3). Firstly, a sigmoid layer is used to decide the value to be updated, after which the layer is used to create the candidate vector $c_t$ to be added. Finally, the updated value is created by combining the two vectors:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

where $W_i$ is the weight matrix of the input gate, $b_i$ is the offset vector of the input gate, $W_c$ is the weight matrix of the candidate vector $c_t$, and $b_c$ is the offset vector of $c_t$. The output gate $o_t$ determines the final output information according to Eqs. (4) and (5). Firstly, the value is normalized through a tanh layer and then multiplied by the output of the sigmoid layer to determine the output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$h = o_t \cdot \tanh(c_t) \quad (5)$$

where $W_o$ is the weight matrix of the output gate and $b_o$ is the offset vector of the output gate. The selective elimination, addition, and output of expired information, new information, and final results of capturing DGA domain name features are determined through the collaborative control of these three gates. LSTM can store and acquire the status of long sequences, overcome the problem of feature invalidation, and maximize the comprehensive features of DGA domain name character sequences to effectively distinguish differences in DGA domain names. Finally, the output of the LSTM layer is passed through the softmax layer, and the maximum bit of relative probability is selected as the final output of the DGA classification results.

### 3.3 Imbalanced multiclass learning based on the long short-term memory-based property and quantity dependent optimization method

The imbalance of DGA samples in datasets will directly lead to unsatisfactory classification results of the model. Number of samples of DGA domain names is shown in Fig. 2. The Cryptolocker class only includes 92 samples, whereas the Banjori class features 452 426 samples. Thus, the number of samples of DGA domain names between these classes is highly imbalanced.
asymmetric information provided by different classes of samples for training algorithms will ultimately affect the classification results of LSTM for DGA domain names; as such, using effective algorithms is necessary to balance the training samples.

Most existing studies on imbalanced datasets are based on data- or algorithm-level optimization. This approach takes the original number of samples as the most important determinant to adjust the number of samples or optimize the classification algorithm. However, we find that different classes have different sample characteristics, such as sample complexity, intra-class distance, N-gram frequency, domain name character length, and transliteration ability, and the characteristics of each sample represent a determinant of equal importance as the original number of samples during sampling proportion optimization. For example, the average domain name length of Murofet is 21; this class cannot transliterate, and its combination of letters and numbers is complex. By comparison, the average domain name length of Suppobox is 17; this class can transliterate, and its domain name composition is relatively simple. Therefore, Murofet has more complex characteristics than Suppobox. From the perspective of sample characteristics, the Murofet class is unsuitable for large-scale calculations even if it has a large number of samples. Meanwhile, for the suppobox class, we can consider reducing the sampling ratio appropriately.

Optimization methods based on the original number of samples to determine the sampling ratio or improve the classification algorithm can reduce the degree of data imbalance to a certain extent because of their mature and logical principle. However, these methods ignore the characteristic of different classes of samples, which results in optimization efforts, not yielding the optimal sampling ratio.

Given this background, we propose an LSTM.PQDO method based on dynamic optimization of sampling proportion by simultaneously considering the original number and characteristics of the samples. In this method, the optimal solution obtained by optimizing the sampling proportion based on the original number of samples is taken as the initial iteration value, and the sampling proportion is iterated in the right direction and around the optimal solution. The optimal solution is heuristically searched by the optimization algorithm to determine the final optimal sampling proportion. Moreover, on the basis of the mature and logical principles of the existing research, the characteristics of different classes of samples are fully considered to maximize the sampling proportion of imbalanced datasets, address the challenges of data imbalance, and improve the classification effect of DGA.

Assuming that the number of DGA classes is $N$, the maximum number of samples in the sample set is $\text{max}_c$, and the number of samples in each category is $\text{cnt}_i, i \in N$. $\alpha$ is used to control the initial values of the relative sampling coefficients of various classes so that they are as balanced as possible. The value of $\alpha$ in this paper is 0.4. The reason for this value will be explained in the next section. The relative sampling coefficient of each DGA class is $w_i, i \in N$ and the
The specific optimization process is shown in Algorithm 1. Firstly, the relative sampling coefficients \( w_1^{(0)}, \ldots, w_N^{(0)} \) of DGA are initialized. Because different initial values will directly lead to different final sampling proportions, we propose an initialization method to calculate the relative sampling coefficients of each DGA class sample according to Eq. (6). After initialization, the optimal solution is searched by iterating the relative sampling coefficients of different classes of DGA so that the solution obtained by each search gradually approaches the final optimal solution in the right direction. The optimization process proposed in this paper can be implemented based on a variety of heuristic optimization algorithms, such as genetic algorithm, Particle Swarm Optimization (PSO), and bat algorithm. In this work, the proposed method is realized according to PSO, and the optimal sampling ratio of DGA domain name samples is determined.

In fact, the better solution usually occurs around the initial value, and searching the whole space is unnecessary. Searching the whole space would be time-consuming because each individual must train the LSTM model, and the deep neural network model has a complex structure and massive training dataset, thus, it requires a large amount of calculation for training. In the actual experiment, we also find that if \( w \) is not limited, the elements in \( w \) may sometimes be optimized to be negative values, which is inconsistent with the actual situation. Therefore, upper bound \( w^{(up)} \) and lower bound \( w^{(down)} \) constraints are set for the search range of each better solution in each iteration process to enable heuristic and efficient searching for the better solution in the right direction around each better solution. This step controls the correct search direction of the better solution, improves the search efficiency of the better solution, and reduces the computational complexity of our method. In this paper, the \( w^{(down)} \) is set to be half of the initial \( w \) and the \( w^{(up)} \) is set to be three times the ratio of the maximum number of samples to the number of such samples of the initial \( w \) (Line 9). During exploration of a reasonable limit of the search range of the better solution, we find that such a setting can limit the reasonable search range according to the number of specific samples of each class; thus, this limit can not miss any better solution but also appropriately narrow the search range to ensure efficient searching of the better solution in the right direction. The optimization process proposed in this paper can be implemented based on a variety of heuristic optimization algorithms, such as genetic algorithm, Particle Swarm Optimization (PSO), and bat algorithm. In this work, the proposed method is realized according to PSO, and the optimal sampling ratio of DGA domain name samples is determined. PSO is a group-based evolutionary algorithm that finds the optimal solution heuristically through cooperation and information sharing among individuals in a group. Its advantage lies in the simplicity of the algorithm.

Algorithm 1 LSTM.PQDO

| Input: dataset, \( \alpha \), \( N \), max; |
| Output: trainset; |
| for \( i \leq N \) do |
| \( w_i^{(0)} = \text{pow} ((\text{max } c) / \text{cnt}_i, \alpha) \); |
| end |
| \( W^{(0)} = [w_1^{(0)}, w_2^{(0)}, \ldots, w_N^{(0)}]; \) |
| while \( t < \text{maxIterations} \) do |
| \( \text{tempDataset} \leftarrow \text{resample}(W^{(t)}); \) |
| \( \text{state} \leftarrow \text{trainModel(tempDataset)}; \) |
| for \( i \leq N \) do |
| \( w_i^{(t)}_{\text{down}} = 0.5 \times w_i^{(0)}; \) |
| \( w_i^{(t)}_{\text{up}} = 3 \times w_i^{(0)}; \) |
| \( W^{(t+1)} \leftarrow \text{OptSearch}(W^{(t)}, \text{state}); \) |
| if \( w_i^{(t+1)} \leq w_i^{(t)}_{\text{down}} \) then |
| \( w_i^{(t+1)} = w_i^{(t)}_{\text{down}} \) |
| end |
| if \( w_i^{(t+1)} \geq w_i^{(t)}_{\text{up}} \) then |
| \( w_i^{(t+1)} = w_i^{(t)}_{\text{up}} \) |
| end |
| trainset \leftarrow \text{resample}(W^{(\text{mthr})}); \) |
| Return(trainset); |

Training model is an LSTM training function based on the tempdataset optimized by the resampling function. The current combination of relative sampling coefficients is evaluated by the state obtained from the training process (Lines 6 and 7). The OptSearch function searches for a better solution (Line 11) based on the current combination of relative sampling coefficients and state and then searches for the optimal combination of relative sampling coefficient \( W^{(t)} \) when the accuracy is determined to be optimal by iteration.

Yijing Chen et al.: DGA-Based Botnet Detection Toward Imbalanced Multiclass Learning
PSO has been widely used in function optimization, neural network training, fuzzy system control, and other applications of genetic algorithms. The implementation of PSO is shown in Algorithm 2. PSO initializes a group of random particles, the position vector \( W_k^{(0)} = [w_{k_1}^{(0)}, w_{k_2}^{(0)}, \ldots, w_{k_N}^{(0)}] \) of each particle according to Eq. (6), and the velocity vector \( V_k^{(0)} = [v_{k_1}^{(0)}, v_{k_2}^{(0)}, \ldots, v_{k_N}^{(0)}] \) of each particle; here, the position and velocity vectors represent the relative sampling coefficient of each class of DGA samples and the speed of weight updating (Lines 3 and 4), respectively.

PSO searches for the optimal solution of the DGA domain name sampling proportion through iteration of various combinations of relative sampling coefficients. In each iteration, the \( w_{\text{up}} \) and \( w_{\text{down}} \) are set as the search range of the better solution to ensure efficient searching of the better solution in the right direction around the better solution. In this experiment, the \( w_{\text{down}} \) is set to be half of the initial \( w \), and the \( w_{\text{up}} \) is set to be three times the ratio of the maximum number of samples to the number of such samples of the initial \( w \) (Lines 21 and 22). In the iteration process, the personal best pbest, which is the optimal solution of the DGA sampling proportion currently searched by each particle, and the global best gbest, which is the optimal sampling proportion of DGA currently searched by all particles, are recorded. In each iteration, an objective function determines the fitness.

The flow chart of the LSTM.PQDO algorithm in a single iteration is shown in Fig. 3. \( F[\text{particle}_k] \) is based on the comprehensive consideration of macro and micro F1-scores and used to evaluate the optimization degree of the candidate DGA weight combination, thus updating the pbest and gbest. After pbest and gbest are obtained, the weight updating speed \( V_k \) and the combination of relative sampling coefficients \( W_k \) are updated by Eqs. (7) and (8), respectively. The LSTM.PQDO algorithm iterates the process continuously to find a better \( W \).

\[
W_k^{(t+1)} = W_k^{(t)} + b_2 \times r_2 \times (\text{gbest}^{(t)} - W_k^{(t)}) + b_1 \times r_1 \times (\text{pbest}_k^{(t)} - W_k^{(t)})
\]

(7)

\[
W_k^{(t+1)} = W_k^{(t)} + V_k^{(t+1)}
\]

(8)

where \( k = \{1, 2, \ldots, K\} \), \( K \) is the size of the particle swarm, and \( t \) is the number of iterations. \( \omega \) is the inertia factor representing the velocity or inertia before the particle and it weighs the local and global optimum abilities. \( W \) can induce particles to maintain their inertia of motion so that they can expand the space necessary to find a better solution of DGA sampling proportion. Moreover, \( r_1 \) and \( r_2 \) are random numbers.

**Algorithm 2 LSTM.PQDO based on PSO**

| Input: dataset, \( \alpha \), N, max, K; |
| Output: trainset; |
| for particle\(_k\in\) Particle do |
| for \( i \leq N \) do |
| \( \text{Randomly initialize } v_{k_i}^{(0)} \text{ for each particle}_k; \) |
| \( w_{k_i}^{(0)} = \text{pow}((\max \_\alpha)/\text{cnt}_i, \alpha); \) |
| \( W_k^{(0)} = [w_{k_1}^{(0)}, w_{k_2}^{(0)}, \ldots, w_{k_N}^{(0)}]; \) |
| \( V_k^{(0)} = [v_{k_1}^{(0)}, v_{k_2}^{(0)}, \ldots, v_{k_N}^{(0)}]; \) |
| \( \text{tempDataset} \leftarrow \text{resample}([W_k^{(t)}]); \) |
| \( F[\text{particle}_k] \leftarrow \text{trainModel(tempDataset);} \) |
| if \( F[\text{particle}_k] > F[\text{pbest}_k] \text{ then} \) |
| \( \text{pbest}_k = W_k^{(t)} \) |
| if \( F[\text{particle}_k] > F[\text{gbest}] \text{ then} \) |
| \( \text{gbest} = W_k^{(t)} \) |
| for \( i \leq N \) do |
| \( w_{k_i}^{(t)} \_\text{down} = 0.5 \times w_{k_i}^{(0)}; \) |
| \( w_{k_i}^{(t)} \_\text{up} = 3 \times w_{k_i}^{(0)}; \) |
| if \( w_{k_i}^{(t+1)} \geq w_{k_i}^{(t)} \_\text{down}, w_{k_i}^{(t+1)} \leq w_{k_i}^{(t)} \_\text{up} \) then |
| \( \text{Update } W_k^{(t+1)}, V_k^{(t+1)} \) |
| \( \text{tempDataset} \leftarrow \text{resample(gbest);} \) |
| Return(trainset); |

**Fig. 3 Flow chart of LSTM.PQDO algorithm in a single iteration.**
with values of [0, 1] to ensure the diversity of the particle swarm, and \( b_1 \) and \( b_2 \) are learning factors representing the acceleration of each particle moving to the gbest and pbest, respectively. \( b_1 \) confers each particle with cognitive ability, i.e., the particle holds the memory of its own DGA sampling proportion, and \( b_2 \) confers each particle with the ability to share information so that it can reach a new search space. Therefore, particles can continuously search for better solutions for the DGA sampling ratio through information sharing and their cognitive ability. In our experiment, the initial value of \( \omega \) is set to 0.8, the initial values of \( r_1 \) and \( r_2 \) are set to 0.3 and 0.4, respectively, and the initial values of \( b_1 \) and \( b_2 \) are both set to 2.

4 Experimental Evaluation and Discussion

4.1 Dataset description and experimental setting

In this paper, we use public datasets for the experiments. The data source is the OSINT DGA feed from Bambenek Consulting\(^{[21]} \) which is the same dataset used by Tran et al.\(^{[13]} \) and a widely used dataset in DGA detection research. This dataset contains 36 DGA classes. Table 1 shows the names of all DGA classes and the number of all samples in the dataset. Table 1 reveals that Cryptowall has only 92 samples, whereas Banjori has 452,426 samples. The proportion of the two classes is approximately 1:5000. This serious imbalance of numbers of samples will lead to an imbalance of the learning of different classes of features in the training process; thus, the expected classification accuracy may not be achieved.

In the experiment, Macro-PRE, Macro-REC, Macro-F1, and AVG are used to measure the effectiveness of the method. TP, FP, and FN represent true positive, false positive, and false negative, respectively. PRE and REC are the accuracy and recall rate, respectively and calculated according to Eqs. (9) and (10):

\[
PRE = \frac{TP}{TP + FP} \quad (9)
\]
\[
REC = \frac{TP}{TP + FN} \quad (10)
\]

Macro-PRE is the average macro-precision for all classifications and Macro-REC is the average macro-recall for all classifications; these parameters are calculated according to Eqs. (11) and (12), respectively. In addition, Micro-F1 is the harmonic mean of Macro-PRE and Macro-REC, and Micro-F1 is the harmonic mean of Micro-PRE and Micro-REC. Macro-F1 and Micro-F1 are calculated according to Eqs. (15) and (16), respectively:

\[
Macro-PRE = \frac{1}{n} \sum_{i=1}^{n} PRE_i \quad (11)
\]
\[
Macro-REC = \frac{1}{n} \sum_{i=1}^{n} REC_i \quad (12)
\]

Micro-PRE is the harmonic mean of Micro-PRE and Micro-REC, and Micro-F1 is the harmonic mean of Micro-PRE and Micro-REC. Macro-F1 and Micro-F1 are calculated according to Eqs. (15) and (16), respectively:

\[
Macro-F1 = \frac{2 \times Macro-PRE \times Macro-REC}{Macro-PRE + Macro-REC} \times 100\% \quad (15)
\]

\[
Micro-F1 = \frac{2 \times Micro-PRE \times Micro-REC}{Micro-PRE + Micro-REC} \times 100\% \quad (16)
\]

AVG is the mean value of Micro-F1 and Macro-F1, calculated according to Eq. (17):

\[
AVG = \frac{Micro-F1 + Macro-F1}{2} \times 100\% \quad (17)
\]

4.2 Analysis for hyper parameter

In this section, three initialization methods for DGA relative sampling coefficients are compared and analyzed. In Mode 1, the relative sampling coefficients are determined by the original number of samples. The relative sampling coefficients of all DGA classes are
initialized to 1, i.e., \( W_i = [1, 1, \ldots, 1] \). In Mode 2, the relative sampling coefficients of all DGA classes are randomly initialized. In Mode 3, the relative sampling coefficients of each DGA class are calculated according to Eq. (6) proposed in this paper.

We carry out experiments on the above three initialization methods and comprehensively analyze and evaluate their performance according to the five indicators of Micro-F1, Macro-PRE, Macro-REC, Macro-F1, and AVG. The experimental data are shown in Table 2. Because the AVG of Mode 1 is only 72.29\%, which is the lowest result obtained from all initialization methods tested, sampling according to the original sample number is not advisable compared with other methods. The AVG of Mode 2 is 74.89\%, which is slightly improved compared with that of Mode 1, but still shows space for optimization.

The AVG of Mode 3 varies with the change in value of \( \alpha \). We observe that the maximum value of AVG is 76.89\%, when the value of \( \alpha \) is 0.4. Compared with those of the two other modes, the AVG of Mode 3 shows a certain degree of improvement and has better performance in other indicators. After comparing and analyzing the three modes, we choose Mode 3 and initialize the relative sampling coefficients of each category by taking \( \alpha \) as 0.4.

In this paper, the relative sampling coefficient of DGA samples is optimized according to the PSO algorithm,

| Mode | Init-method | Micro-F1 | Macro-PRE | Macro-REC | Macro-F1 | AVG  |
|------|-------------|----------|-----------|-----------|----------|------|
| 1    | Original    | 89.31    | 61.22     | 54.46     | 55.26    | 72.29|
| 2    | Random      | 86.37    | 62.68     | 70.79     | 63.42    | 74.89|
|      | \( \alpha \) |          |           |           |          |      |
|      | =0.1        | 89.10    | 58.18     | 58.39     | 57.00    | 73.05|
|      | =0.2        | 89.40    | 62.61     | 58.40     | 57.77    | 73.59|
|      | =0.3        | 87.68    | 64.04     | 67.93     | 62.98    | 75.33|
|      | =0.4        | 88.81    | 63.91     | 69.26     | 64.97    | 76.89|
|      | =0.5        | 84.70    | 58.95     | 67.64     | 60.88    | 72.79|
|      | =0.6        | 86.75    | 61.49     | 73.88     | 63.96    | 75.35|
|      | =0.7        | 85.70    | 61.77     | 72.16     | 63.15    | 74.42|
|      | =0.8        | 85.15    | 61.30     | 74.68     | 63.44    | 74.30|
|      | =0.9        | 81.46    | 54.43     | 73.38     | 57.27    | 69.37|
|      | =1.0        | 76.67    | 57.04     | 71.69     | 58.10    | 67.39|

Fig. 4  Actual number of DGA classes and relative sampling coefficients optimized by LSTM.PQDO method.
and the results are shown in Fig. 4. Here, sample number represents the original number of different classes in the DGA dataset.

Figure 4 shows that the original number of samples of DGA classes is inversely proportional to their optimized relative sampling coefficients. For example, the original number of samples of Tinba is 53,351 and the relative sampling coefficient of PSO is 2.02. By comparison, the original number of samples of Qadars is 32 and the relative sampling coefficient of PSO is 19.87. The optimization results are consistent with the experimental rules of common resampling and common human understanding, i.e., reducing the number of large classes and increasing the proportion of small classes achieve balance in each class. However, different classes of DGAs have different characteristics, such as different sample complexities, intra-class distances, N-gram frequencies, and transliteration abilities. During optimization of sampling proportions, the characteristics of various samples are as important as the original number of samples. Although the existing optimization methods can reduce the degree of data imbalance to a certain extent, they still ignore the important factor of different sample characteristics. Therefore, the current optimization methods present some limitations and optimization space. In this paper, an optimization method based on the comprehensive consideration of the original number and characteristics of samples is proposed. Therefore, the experimental results do not strictly follow the inverse relationship between the original number of samples and relative sampling coefficients. For example, the numbers of original samples of Tinba and Banjori are 53,351 and 33,733, respectively and the optimized relative sampling coefficients are 2.02 and 0.87, respectively. The number of original samples and the relative sampling coefficient of Tinba are larger than those of Banjori. The results demonstrate that the proposed method can fully consider the characteristics of various samples during optimization, along with the number of samples, to further reduce the impact of imbalanced datasets on classification results.

Figure 5 shows how the relative sampling coefficients $w_i$ of each class of DGA change after optimizing the sampling coefficients of each class by PSO. When the weight change ratio is greater than 0, the class is oversampled; conversely, when the weight change ratio is less than 0, the class is undersampled.

Analysis of the experimental data reveals that PSO does not increase the sampling ratio for all DGA classes. We can observe that the relative sampling coefficients of Bamital, Padcrypt, and Ranbyus decrease compared with the initial number, which further shows that determination of the optimal sampling proportion must consider not only the original sample number of various classes but also the characteristics of the DGA samples.

The experimental results confirm that the method of determining relative sampling coefficients based on the original number of samples has some limitations. The LSTM.PQDO method takes into account the original number of samples and the characteristics of
various samples, overcomes the shortcomings of existing research, and achieves the maximum optimization of sampling proportion.

4.3 Effectiveness analysis of the long short-term memory-based property and quantity dependent optimization approach

In this section, the traditional LSTM-based and PSO-based LSTM.PQDO detection methods are compared and analyzed. The confusion matrix of the two methods is shown in Fig. 6. Each row in the obfuscation matrix represents the actual class and each column represents the predicted class. The specific elements in the matrix are shown in Eq. (18):

\[ X_{ij} = \frac{c_{i,j}}{c_{i}} \]

(18)

where \( c_{i} \) is the total number of samples of DGA domain names with class \( i \) and \( c_{i,j} \) is the number of DGA samples misjudged as class \( j \) for the actual class \( i \).

From the overall distribution of the confusion matrix of the two methods, we can see that the data in the confusion matrix of LSTM.PQDO are more concentrated on the diagonal line of the matrix than the data based on LSTM. This result shows that LSTM.PQDO has higher accuracy and, correspondingly, fewer misclassification cases than the traditional LSTM-based method. Analysis of the confusion matrix reveals that LSTM.PQDO performs better than LSTM in terms of classification accuracy.

We use PRE, REC, and F1 to analyze and evaluate LSTM and LSTM.PQDO. The specific experimental data are shown in Table 3. According to statistics, the indices of PRE, REC, and F1 for 24, 22, and 27 DGA classes are higher than those of the non-optimized LSTM algorithm, which confirms that the LSTM.PQDO algorithm performs better in the classification task of most DGA classes. Because the Macro index offers greater reference value than the Micro index in multiclassification, we mainly analyze the LSTM and LSTM.PQDO based on Macro index. Compared with the non-optimized LSTM method, the LSTM.PQDO method has higher accuracy (15.32% increase in Macro-AVG PRE) and recall rate (18.77% increase in Macro-AVG REC) and better overall performance (13.94% increase in Macro-AVG F1 score). This result shows that the LSTM.PQDO method has good performance in solving multi-class imbalance problems.

4.4 Comparison with other approaches

To analyze the effectiveness of the LSTM.PQDO method, we compare its results with those of traditional oversampling methods. Here, we select state-of-the-art methods to compare the data- and algorithm-level optimization methods for imbalanced classification problems, namely, QDBP and LSTM.MulticlassImbalance (LSTM.MI). Oversampling is a method of balancing datasets that continuously re-samples a few classes into training sets. QDBP is a method proposed by Chen et al.\cite{14} to take different sampling proportions for each class based on the quantity difference of each class when calculating the cost function, so as to realize the consideration of the influence of the difference between classes. LSTM.MI is a multi-classification model proposed by Tran et al.\cite{13} that combines the binary and multi-classification models and introduces cost items during back-propagation learning to realize a multi-classification model.
considering class differences. The dataset used by Tran et al.\cite{13} to implement the LSTM.MI method is identical to the dataset used in this paper. Therefore, we directly quote these experimental data for comparative analysis. The oversampling method and dataset used by Chen et al.\cite{14} to implement QDBP are different from the dataset and method used in this paper, but both methods are easy to be implemented. Therefore, LSTM.MI and QDBP are implemented based on the dataset used in this paper, and the experimental results are compared and analyzed.

Table 4 shows the PRE, REC, and F1-score of four classification methods for imbalanced datasets. Analysis of the data in Table 4 reveals that the oversampling method performs the worst among the methods tested, and its respective Micro-AVG and Macro-AVG are 9.27\% and 3.44\% lower than those of QDBP and 11.83\% and 8.29\% lower than those of LSTM.PQDO, respectively. Moreover, its Micro-AVG is 10.32\% lower than that of LSTM.MI because it determines sampling proportions only by the number of samples, which may lose some characteristics of classes. The QDBP and LSTM.MI methods perform similarly and show only a 4\% difference in Macro-AVG F1 index. Compared with the oversampling method, the classification performance of these two methods is slightly improved. Compared with the traditional oversampling method, the LSTM.PQDO method proposed in this paper reveals a great improvement in classification performance; indeed, the Micro-AVG and Macro-AVG F1 indices of this method are improved by 11.83\% and 8.29\%, respectively. In addition, compared with QDBP and LSTM.MI, the performance of LSTM.PQDO method is improved by 5.56\% and 9.09\% in Macro-AVG F1, respectively.

5 Conclusion

Botnets based on the DGA mechanism pose great challenges to the current inspection work because of their strong concealment. Traditional DGA detection methods based on artificial feature extraction have high cost, the extracted features are not flexible, the methods cannot cope with the complexity of the DGA domain name, and a serious imbalance in the number of DGA domain name datasets is observed. Traditional research schemes based on data- or algorithm-level optimization, in which the number of original samples is the most important determinant of the sampling ratio, cannot achieve the optimal sampling ratio, because they ignore differences in the characteristics of various samples. In this paper, the LSTM.PQDO method is proposed to automatically extract the known and potential linguistic statistical features of DGA domain names based on the deep neural LSTM network and dynamically optimize the sampling proportion based on the comprehensive consideration of the number and nature of the original samples. In this method, the optimal solution obtained by the existing
optimization method is used as the initial iteration value to iterate the sampling ratio, and the optimal solution is searched heuristically around the initial solution in the right direction. In contrast to existing optimization methods, the proposed method is based on the number of original samples and fully considers the impact of different classes of samples so as to minimize the impact of imbalanced datasets on the classification results.

To evaluate the performance of the LSTM.PQDO method, we conducted a series of experiments on existing imbalanced datasets. Compared with the non-optimized LSTM method, our proposed method yields a 13.94% improvement in Macro-AVG F1 index with a 0.25% difference in Micro-AVG F1 index. Compared with those of several existing optimization methods, the Macro-AVG F1 index of our method increased by 4.58%–9.09%. Overall, the experimental results showed that our method can overcome the challenge of imbalanced datasets.
datasets and optimize the detection performance of DGA domain names.

Acknowledgment

This work was partially funded by the National Natural Science Foundation of China (No. 61272447), the National Entrepreneurship & Innovation Demonstration Base of China (No. C700011), and the Key Research & Development Project of Sichuan Province of China (No. 2018G20100).

References

[1] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, Botnet in DDoS attacks: Trends and challenges, IEEE Commun. Surv. Tutorials., vol. 17, no. 4, pp. 2242–2270, 2015.

[2] C. L. Zhou, K. Chen, X. X. Gong, P. Chen, and H. Ma, Detection of fast-flux domains based on passive DNS analysis, (in Chinese), Acta Sci. Natur. Univ. Pekinensis, vol. 52, no. 3, pp. 396–402, 2016.

[3] C. D. Chang and H. T. Lin, On similarities of string and query sequence for DGA botnet detection, in Proc. 2018 Int. Conf. on Information Networking, Chiang Mai, Thailand, 2018, pp. 104–109.

[4] J. Kwon, J. Lee, H. Lee, and A. Perrig, PsyBoG: A scalable botnet detection method for large-scale DNS traffic, Comput Networks, vol. 97, pp. 48–73, 2016.

[5] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, Detecting algorithmically generated domain-flux attacks with DNS traffic analysis, IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1663–1677, 2012.

[6] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, Phoenix: DGA-based botnet tracking and intelligence, presented at 11th Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment, Egham, UK, 2014, pp. 192–211.

[7] D. T. Truong and G. Cheng, Detecting domain-flux botnet based on DNS traffic features in managed network, Secur. Commun. Networks, vol. 9, no. 14, 2016, pp. 2338–2347.

[8] V. Tong and G. Nguyen, A method for detecting DGA botnet based on semantic and cluster analysis, in Proc. Seventh Symp. on Information and Communication Technology, Ho Chi Minh City, Vietnam, 2016, pp. 272–277.

[9] J. Mathew, M. Luo, C. K. Pang, and H. L. Chan, Kernel-based SMOTE for SVM classification of imbalanced datasets, in Proc. 41st Conf. of the IEEE Industrial Electronics Society, Yokohama, Japan, 2015, pp. 1127–1132.

[10] W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, Clustering-based undersampling in class-imbalanced data, Inf Sci, vol. 409–410, pp. 17–26, 2017.

[11] J. Ha and J. S. Lee, A new under-sampling method using genetic algorithm for imbalanced data classification, presented at 10th Int. Conf. on Ubiquitous Information Management and Communication, Danang, Vietnam, 2016.

[12] S. Gazzah, A. Hechkel, and N. E. B. Amara, A hybrid sampling method for imbalanced data, in Proc. 2015 IEEE 12th Int. Multi-Conference on Systems, Signals & Devices, Mahdia, Tunisia, 2015, pp. 1–6.

[13] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, A LSTM based framework for handling multiclass imbalance in DGA botnet detection, Neurocomputing, vol. 275, pp. 2401–2413, 2018.

[14] Y. C. Chen, Y. J. Li, A. Tseng, and T. Lin, Deep learning for malicious flow detection, arXiv preprint arXiv: 1802.03358, 2018.

[15] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, Predicting domain generation algorithms with long short-term memory networks, arXiv preprint arXiv: 1611.00791, 2016.

[16] Y. Li, K. Q. Xiong, T. Chin, and C. Hu, A machine learning framework for domain generation algorithm-based malware detection, IEEE Access, vol. 7, pp. 32 765–32 782, 2019.

[17] F. Zeng, S. Chang, and X. C. Wan, Classification for DGA-based malicious domain names with deep learning architectures, Int. J. Intell. Inf. Syst., vol. 6, no. 6, pp. 67–71, 2017.

[18] B. Athiwaratkun and J. W. Stokes, Malware classification with LSTM and GRU language models and a character-level CNN, in Proc. 2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 2017, pp. 2482–2486.

[19] B. Yu, J. Pan, J. M. Hu, A. Nascimento, and M. De Cock, Character level based detection of DGA domain names, in Proc. 2018 Int. Joint Conf. on Neural Networks, Rio de Janeiro, Brazil, 2018, pp. 1–8.

[20] L. L. Gao, Z. Guo, H. W. Zhang, X. Xu, and H. T. Shen, Video captioning with attention-based LSTM and semantic consistency, IEEE Trans. Multimedia., vol. 19, no. 9, pp. 2045–2055, 2017.

[21] Bambenek Consulting-Master feeds, http://osint. bambenekconsulting.com/feeds/, 2019.

Yijing Chen received the bachelor degree from Sichuan University, Chengdu, China in 2020. Her research interests include deep learning of network security and big data analysis. She has won the network security scholarship of Sichuan University in 2017 and 2018, the first prize of the 12th China University Computer Design Competition in 2019, the Honorable Mention of the Mathematical Contest in Modeling in 2019, etc.

Bo Pang received the bachelor degree from Sichuan University, Chengdu, China in 2020. His research interests include Web security, deep learning for Web security, big data analysis, etc. He has received numerous honors and awards, including the fifth place in the Sichuan University AI Challenge in 2018 and the third prize of Sichuan University Student Information Security Technology Competition.
Guolin Shao received the BS and PhD degrees from Sichuan University, Chengdu, China in 2013 and 2018, respectively. His research interests include deep learning for cyber security and big data analysis. He has published more than 16 peer reviewed papers. He has received numerous honors and awards, including the National Cyber Security Scholarship in 2016 and 2018, the National Scholarship in 2017, the Top Ten Academic Star of Sichuan University, and the First Prize Scholarship in 2013, 2015, and 2017, respectively.

Guozhu Wen received the bachelor degree from Sichuan University, Chengdu, China in 2020. His research interests include Web security, deep learning for Web security, big data analysis, etc. He has received numerous honors and awards, including the fifth place in the Sichuan University AI Challenge in 2018 and the third prize of Sichuan University Student Information Security Technology Competition.

Xingshu Chen is a full professor at College of Cybersecurity, Sichuan University, Chengdu, China. She received the master and PhD degrees from Sichuan University in 1999 and 2004, respectively. Her main research interest focuses on cloud computing, big data analysis, and network security.