IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 27, 2020, accepted April 15, 2020, date of publication April 20, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988877

Detecting Stealthy Domain Generation Algorithms Using Heterogeneous Deep Neural Network Framework

LUHUI YANG[1], GUANGJIE LIU[2], (Member, IEEE), YUEWEI DAI[2], JINWEI WANG[3], (Member, IEEE), AND JIANGTAO ZHAI[2]

[1]School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
[2]School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]Department of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Corresponding author: Guangjie Liu (gjieliu@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U1836104, Grant 61702235, and Grant 61921004, and in part by the Fundamental Research Funds for the Central Universities under Grant 30918012204.

ABSTRACT Distinguishing malicious domain names generated by various domain generation algorithms (DGA) is critical for defending a network against sophisticated network attacks. In recent years, stealthy domain generation algorithms (SDGA) have been proposed and revealed significantly stronger stealthiness comparing to the traditional character-based DGA. Existing state-of-the-art detection schemes are not effective enough for detecting SDGA. In this paper, we exploit the character-level characteristics of the SDGA domain names and propose a heterogeneous deep neural network framework (HDNN) for detecting SDGA. HDNN employs a proposed improved parallel CNN (IPCNN) architecture with multi-sizes of convolution kernel for extracting multi-scale local features from a domain name. The framework also contains a proposed self-attention based bidirectional long short term memory (SA-Bi-LSTM) architecture which can extract the bidirectional global features with attention mechanism from a domain name. Besides that, the focal loss function is introduced to mitigate the imbalance of the sample quantity in the training phase. The benchmark experiments are carried out based on the database composed of the collected benign domain names, real-world DGA and SDGA ones. Compared to the 6 influential deep-learning-based DGA detection schemes, the proposed scheme has achieved state-of-the-art detection results on SDGAs, and also achieved state-of-the-art results on binary and multiclass classification for traditional DGAs.

INDEX TERMS Convolutional neural network, cyber security, domain generation algorithm, deep learning, long short term memory.

I. INTRODUCTION

Injected hosts in a compromised network have to connect to the remote servers for downloading or updating the function codes, receiving the command information, uploading the eavesdropped datum and so on. Sometimes, the server is only a beacon, the rendezvous, which will guide to the real command and control (C&C) network, or it is just the C&C server itself. Using hard-coded IP addresses or domain names in malwares is easy to be located in the malware codes and banned by the blacklist maintained by network security product vendors, the threat intelligence alliance, the opensource threat intelligence community and so on. Therefore, over the decade, domain generation algorithms (DGA) have been wildly used for the more imperceptible network.

A DGA program consists of a random seed that is shared by the malware side and the C&C side such as the current timestamp, a dictionary that contains the basic elements for making up a domain name, and an algorithm for generating a domain name. When a malware needs to keep in touch with the C&C network, the shared random seed of DGA will be activated and some elements will be selected from the dictionary to randomly generate a domain name according to the algorithm. At the C&C side, a set of domain names has been generated based on the same DGA and has been registered at Internet service providers to respond to the query

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

request from the victim hosts validly. On the side of the cyber attacker, by using DGA, every connection between the malware and the C&C server will rely on different domain names, which will reduce the possibility of being discovered and blocked by the cyber defender. It is revealed that DGAs have been widely used by numerous botnets, so far as to the advanced persistent threat (APT). To reduce the potential threats of DGAs to cybersecurity, a mountain of research has been done on the analysis of disclosed DGA domain names. According to the basic constituent elements, DGA can be divided into three main categories: the character-based method, the word-based method and the hybrid one. Among them, the character-based DGA is the most primitive one, the dictionary of it is mostly consist of alphabet and numbers [1].

Most disclosed DGAs use some relatively simple algorithms to generate domain names which maintain much stronger randomness in character level compared to those benign domain names. Various kinds of entropy-based detection methods have been proposed and proved to be effective. Furthermore, some deep-learning-based methods based on the basic neural network such as long short term memory (LSTM) and convolutional neural network (CNN) models have been able to achieve state-of-the-art detection results on most traditional DGAs. For resisting the entropy-based detection methods, Fu *et al.* [2] proposed the stealthy domain generation algorithms (SDGAs) which employed hidden Markov models (HMMs) and probabilistic context-free grammars (PCFGs) to select characters from the dictionary to generate domain names mimicking the character distribution of benign domain names. Most traditional entropy-based methods can not detect SDGAs effectively. The basic LSTM and CNN model also does not perform well.

There are total three key challenges in the SDGAs detection problem. Firstly, the length of a SDGA domain name is quite short which leads to less distinctive features comparing with benign domain names, thus the effective short text classification methods in the natural language processing (NLP) field may lose their effect in this situation. Secondly, there is a large quantity gap between benign domain names and the SDGA ones. The former can be collected from various resources and can reach dozens of millions in the amount, while the latter is less than 100,000. The imbalance of samples increases the classification difficulty of machine-learning-based methods. Finally, a large part of the quantities of SDGA samples is easy to be classified while quite a fraction of them are difficult to be distinguished. Thus, the detection of SDGAs still is a challenging work.

In this paper, based on the analysis of SDGA principle, we proposed a heterogeneous deep neural network framework, which employs an improved parallel CNN (IPCNN) architecture and a self-attention based bidirectional LSTM (SA-Bi-LSTM) architecture. IPCNN is proposed to extract local features from a domain name, and SA-Bi-LSTM is proposed to extract the global features with an attention mechanism. To deal with imbalanced difficulty, the specific loss function named focal loss is adopted. The proposed scheme is benchmarked on benign domain samples collected from the public datasets and malicious samples in [2]. A series of experiments are carried on to compare the proposed scheme with some state-of-the-art detection methods based on deep learning. The experimental results show that the proposed scheme can achieve significantly higher detection accuracy compared to the existing state-of-the-art DGA detection schemes. The proposed scheme is also benchmarked on the binary classification and multiclass classification tasks on a total of 20 types of DGA families from the real world, and the comparison classification results denote that the proposed scheme can achieve higher accuracy on both the binary and multiclass classification.

The main contributions of this paper are as follows:

1) A Heterogeneous Deep Neural Network (HDNN) framework is proposed to make SDGAs and DGAs detection. This framework can extract effective character-level local features and global features, which can be used for more accurate detection and classification. The comparative experiments denote that the proposed framework has achieved state-of-the-art results.
2) An IPCNN architecture and an SA-Bi-LSTM architecture are proposed. IPCNN architecture can extract local features in multiple width and depth scales. SA-Bi-LSTM architecture can extract global features with more attention.
3) The focal loss function is introduced into the model training phase, it can reduce the loss of easy-to-detect samples and increase the loss of hard-to-detect samples. In this way, it can effectively mitigate the classification problem caused by the imbalance of training samples.

The remainder of the paper is organized as follows. In the next section, we summarize the related work on the development of DGA and existing DGA detection methods. In Section 3, we firstly describe the proposed detection framework including the IPCNN and SA-Bi-LSTM architecture. Next, we describe the detection scheme based on the heterogeneous deep neural network framework. Then we introduce focal loss and describe the basis for improvement. In Section 4, the binary experiment is conducted on SDGA, and a series of binary and multiclass classification experiment is conducted on the traditional DGAs. Finally, in Section 5, we conclude this paper and discuss future work.

II. RELATED WORK

Over the past decade, DGAs have become an essential component in many kinds of network attacks especially broadly used in Botnets. According to the fundamental function of DGAs, they are classified into two categories according to [1], one is binary-based DGA which is always used for directing to the C&C server, the other is script-based DGA which is embedded in JavaScript code loaded in the browser. Binary-based DGAs is the most widely studied branch which

has further developed six main categories. Among which, the first generates word-composed domain names based on the dictionary with specific words, the second is based on dynamic DNS used by CDN providers, the third generates alphabet-layout domain names using alphabetic letters randomly, the fourth generates number-layout domain names using numbers randomly, the fifth generates alphanumeric domain names using both letters and numbers randomly, and the last generates hybrid-layout domain names using letters, numbers, and words randomly.

All the types of binary-based DGAs can be concluded as the character-based DGAs except for the dictionary-based type. In the beginning, most DGAs are designed in a straightforward way, such as using the GMT date as the seed of random number generator to provide random strings with 6-11 characters. These simply designed DGAs will generate a domain name with a quite obvious character-level randomness, which can be detected by checking statistical measurement of characters, their pairs or triples, such as edit distance and Kullback-Leibler divergence. In order to generate domain names that appear more normal, Crawford and Aycock [3] proposed to generate domain names using the Markov process on English syllables.

Traditional DGA detection schemes fall into two main types: domain-based and behavior-based ones. Most works are focused on the domain-based methods for real-time online detection purpose, since it is easy to be implemented without long periods of observations and additional information. Yadav *et al.* [4], [5] proposed the detection methods based on linear regression classifier with the extracted features composed of Kullback-Leibler divergence, Jaccard index coefficients, edit distance and so on. The method is verified effective for most DGAs with simple design logics, whereas the detection efficiency is inadequate caused by the low-dimension features and the simple classifier. Furthermore, Schiavoni [6], [7] introduced the meaningful character ratio and the n-gram normality score which are two basic linguistic features, respectively. The meaningful character ratio calculates the ratio of characters in a domain name, and the n-gram normality score is obtained by finding n-grams within a domain and the English words. Bilge *et al.* [8] proposed the so-called EXPOSURE system which extracted 15-dimensional features from a domain name and employed J-48 decision tree for classification. Similarly, Schales *et al.* [9] proposed to employ the 17-dimensional features and four kinds of weighted confidence scores to make detection. As a comparison study, Zago *et al.* [10] compared all the algorithms mentioned above and found out that only a minor fraction of the defined features is indeed practical and informative.

The behavior-based methods need auxiliary information to make detection. For example, Mowbray *et al.* [11] proposed a detection scheme based on the length distribution of DNS requests, which can be partially effective for detecting some 0-day DGAs. Raghuram *et al.* [12] proposed an anomaly-based detection method which used natural language processing method to analyze the character features of benign domain names. Grill *et al.* [13] proposed a detection method based on network flow information of DNS request traffic. Nguyen *et al.* [14] proposed an offline DGA detection method based on whitelist filtering and unsupervised clustering method. Wang *et al.* [15] proposed BotMeter which relied on a long period of analysis on the DGA-bot population landscapes in large-scale networks. Shi *et al.* [16] proposed an extreme machine learning method based on the construction-based features, IP-based features, TTL-based features, and WHOIS-based features. It is worth to pointing out that the detection of DGA in the real-time and large-scale network scenario have to rely on the method only based on domain names, which is because that some auxiliary information are usually difficult to retrieval immediately in the real-world DGA detection scenario.

The handcrafted features with low dimensions have proven to be not enough to achieve higher accuracy detection, which leads to the development of deep-learning-based detection methods. Woodbridge *et al.* [17] proposed an LSTM based model to detect various DGAs and the detection accuracy on most character-based DGA families can achieve about 100%. Inspired by [17], a CNN-based DGA detection algorithm was proposed in [18], and the comparative analysis denotes that the CNN and LSTM based methods perform significantly better than the machine learning method based on random forest. Yu *et al.* [19] compared a series of character-level text classification algorithms, found out that a parallel CNN architecture which was originally proposed in [20] achieved the best detection accuracy. Berman *et al.* [21] tried to use the new deep learning architecture CapsNet for DGA detection and it performed as well as the CNN-based algorithms. Tran *et al.* [22] firstly noticed the data imbalance of the multiclass DGA and benign domain names and proposed a cost-sensitive LSTM framework to mitigate the unbalanced difficulty of DGA domain names and improved the detection rate. Qiao *et al.* [23] proposed a classification method based on LSTM with attention mechanism. Xu *et al.* [24] suggested an n-gram based domain name representation, used a parallel CNN architecture for classification, and achieved significantly better results on specific DGA families. Li *et al.* [25] proposed a machine learning framework for DGA detection, a combination of DNN, clustering and HMM methods is proposed for better results.

Since most of the traditional DGAs can be effectively detected by various of the existing detection methods, Fu *et al.* [2] proposed stealthier domain generation algorithms which used hidden Markov models (HMMs) and probabilistic context-free grammars (PCFGs), respectively. HMMs employed different numbers of history symbols used to calculate state transition probabilities within dictionaries of English words and benign domain names. PCFGs employed the syllables form English, IPv4 domain names, and numbers to generate domain names. SDGAs can generate domain names with less abnormity and resisting the existing detection algorithms.

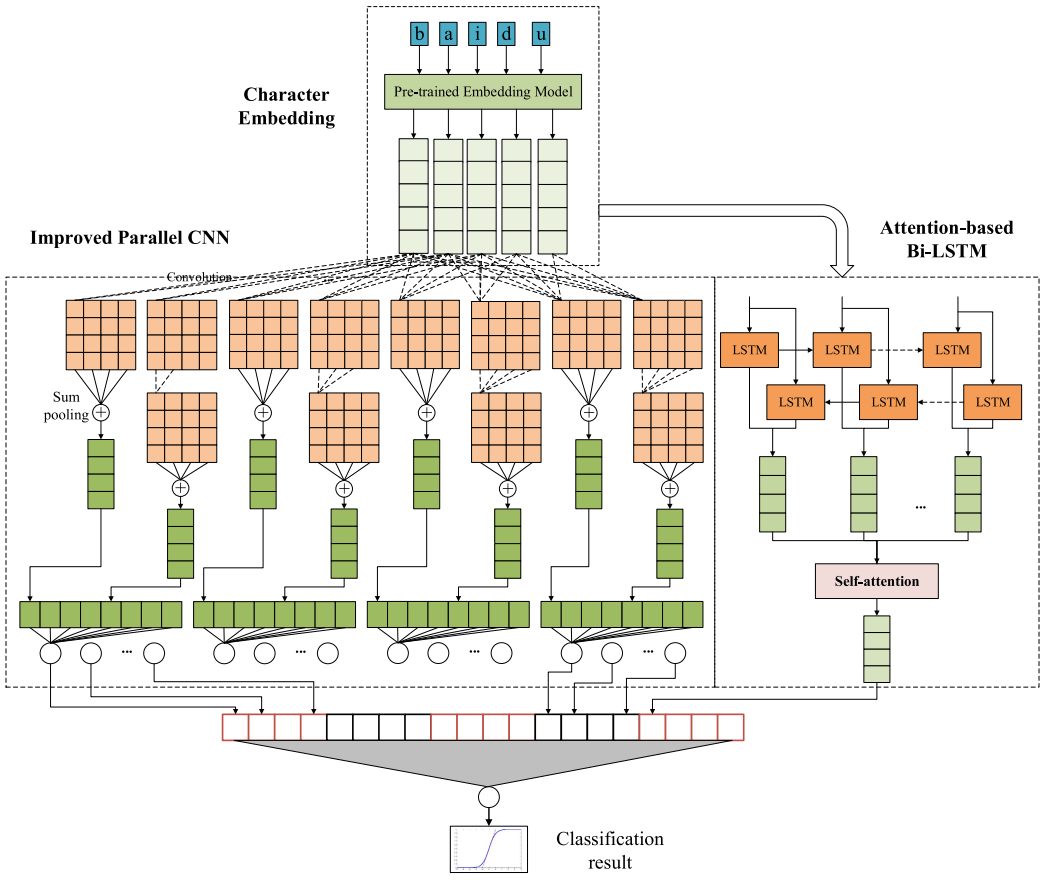


FIGURE 1. Architecture of HDNN.

III. HETEROGENEOUS DEEP NEURAL NETWORK FRAMEWORK

In this section, a heterogeneous deep neural network framework (HDNN) is proposed. To implement the entire framework, A Character Embedding (CE) architecture, an Improved Parallel CNN (IPCNN) architecture, and a Self-Attention based Bidirectional LSTM (SA-Bi-LSTM) architecture are proposed. CE can map characters to vectors based on the character co-occurrence. IPCNN is proposed for extracting multi-scale local features of domains based on multi-scale depth and width CNN architecture. SA-Bi-LSTM is proposed for extracting global features of domains based on Bi-LSTM architecture and self-attention mechanism. Besides the HDNN framework, the focal loss is introduced in the model training phase for higher classification accuracy of unbalanced samples. Figure 1 shows the architecture of the HDNN.

A. CHARACTER EMBEDDING ARCHITECTURE

When using a deep neural network to deal with domain names, it is necessary to quantify the characters of domain names into numeric vectors, this process is usually called Embedding. In natural language processing, there are generally two types of character embedding methods, one is one-hot representation model, the other is distributed representation model. The one-hot representation model uses a long vector to represent a character, which consists of a single one and the rest of zeros. The single one represents the position of the character in the dictionary which is made up of all possible characters. The disadvantage of one-hot representation is that the dimensions tend to be too high and can not effectively express the relationship between two characters. The distributed representation model uses a fixed dimensional float vector to uniquely represent a character. In HDNN, the distributed representation model is adopted to fulfill character embedding. The embedding architecture used in the proposed framework is based on Glove [26], which is a word representation model based on word co-occurrence. For character embedding, 87 characters possibly appear in collected domain names are recorded, and the occurrence frequency of each character is calculated. Then a character co-occurrence matrix is produced by calculating the co-occurrence probability of characters in a large number of legitimate domain names. At last, the character vectors are trained based on the character co-occurrence matrix. In this work, every input domain is padded with zeros to the length of 128, and each character is embedded into a 128-dimensional vector. The embedding architecture is shown in Figure 2.

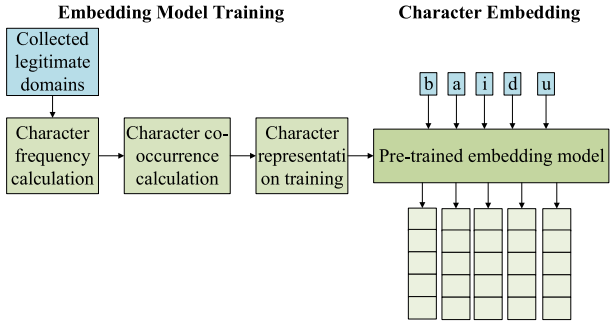


FIGURE 2. Character Embedding architecture.

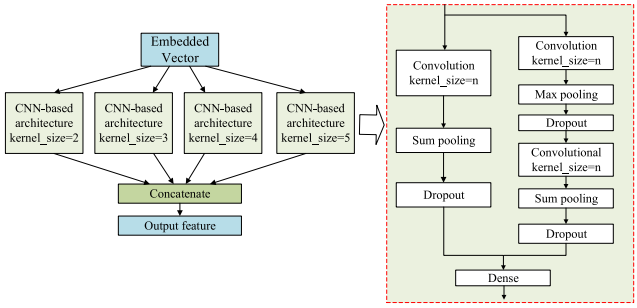


FIGURE 3. Improved parallel CNN architecture.

B. IMPROVED PARALLEL CNN ARCHITECTURE

The convolutional neural network was originally used for image classification and recognition tasks, and has demonstrated breakthrough improvements [27]. CNN structure is very suitable for extracting pixel-level features from a 2-d image by the convolution computation of pixel values. Likewise, the convolution model can also be used for calculating the word-level characteristic relations after word embedding operation. Kim *et al.* [28] firstly proposed a scheme for sentence classification using word2vec and CNN. After words are embedded into vectors, the convolution operation of adjacent n vectors is similar to extracting n-gram features commonly used in the traditional NLP field. Since the word-to-vector mapping depends on a large number of corpus resources, and the accuracy of the word vector has a great impact on the accuracy of the model. Zhang *et al.* [29] have proved that character-level convolutional networks can be also effective for text classification.

For better classification of some types of short text, such as malicious URLs and file paths, a parallel CNN (PCNN) which consists of four CNN branches with single CNN layer was proposed in [20], which was also introduced into DGA detection and achieved better detection result compared to other four types of deep learning architecture with CNN and LSTM layers. The parallel CNN architecture is comprised of 4 branches of CNN-based architecture, each branch is made up of 1 CNN layer, 1 sum-pooling layer, and 1 dropout layer, and the size of the convolution kernel is from 2 to 5.

To improve the PCNN architecture, a novel Improved Parallel CNN (IPCNN) architecture is proposed. The convolutional structure with different convolution kernel sizes is adopted in the PCNN architecture, which can effectively increase the receptive field of the model and extract more accurate character-level features. However, single layer CNN can only extract the characteristics of strings of different lengths, but can not obtain the characteristics between substrings at the deeper level. Taking the domain name word "baidu" as an example, single-layer convolution with different sizes of convolution kernel can extract the features of "ba", "bai", "baid" and "baidu". However, for meaningful domain names, there is also a connection between different substrings in the domain name. For example, "bai" and "du" represent two words in Chinese, and there is a correlation between these two words. For extracting the relationship and features between the meaningful substrings, we improved the PCNN architecture by adding a convolution branch with two convolutional layers to each parallel branch. IPCNN architecture is shown in Figure 3. The added convolution branch consists of two convolutional layers, one max-pooling layer, two dropout layers, and one sum-pooling layer. The output of the new branch is concatenated with the original branch and as the input to a dense layer. The dense layer is a fully connected layer for extracting and outputting specific dimensional features. The keypoint of IPCNN architecture can be expressed according to

$$\begin{cases} C(t,k) = k * x_t * x_{t-1} \\ R_1 = \sum_{j \in K} \sum_{i=0}^{t} C(i,j) \\ I(t,k) = k * C(t,k) * C(t-1,k) \\ R_2 = \sum_{j \in K} \sum_{i=0}^{t} I(i,j) \\ R = R_1 + R_2 \end{cases} \tag{1}$$

where x_t is a member of the input vector $[x_1, x_2, \cdots, x_n]$, $x_t * x_{t-1}$ reprents convolution, and k is kernel size of the convolution. R_1 is the result of the first convolutional layer. K is the size of the convolution kernel at different scales. In this paper, $K = [2, 3, 4, 5]$. $I(t, k)$ is convolution based on the result of first convolutional layer with kernel size of k. In this way, features can be extracted in depth scale. R_2 is the reuslt of the second convolutional layer. R is the reuslt combining R_1 and R_2.

It can be seen from Equation 1 that the convolution is the operation on the local area of the input, which means the convolutional feature can be regarded as local feature of the input. With different kernel size, the local feature can be extracted in multiple width scale. With the first and sencond convolutional layer, local feature can be extracted in multipple depth scale. Overall, IPCNN architecture can extract multiscale character-level local features in multiple width and depth scales.

C. SELF-ATTENTION BASED BI-LSTM ARCHITECTURE

Recurrent Neural Network (RNN) [30] is an extension of the conventional feed-forward neural network which is effective for the classification of sequence data. To overcome the gradient vanishing problem, the long short-term memory network (LSTM) [31] was proposed and have achieved superior success for lots of classification tasks. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.

Since LSTM just owns the ability to extract the single directional relationship of words, Zhou *et al.* [32] proposed the bidirectional LSTM (Bi-LSTM) for extracting bidirectional relationship between words in a sentence. Bi-LSTM can be also used to extract the bidirectional relationship between characters in a domain name.

The attention mechanism has been widely used in various NLP tasks for the past few years. The key point of the attention mechanism is to find out the weight of the hidden state corresponding to each word in a sentence, which usually requires global information. However, the global information is not valid for determining the weight of every character in a domain name since the number of characters in a domain name is a limited minority while the domain name is of huge quantity. The self-attention mechanism [33] is adopted into our model since it only calculates the correlation coefficient between every word in a sentence.

Based on the Bi-LSTM architecture and self-attention mecanism, we proposed a self-attention based bidirectional LSTM (SA-Bi-LSTM) architecture. The structure of SA-Bi-LSTM is shown in Figure 4. The input of the architecture is a 1-dimensional embeded character vector $[x_1, x_2, \cdots, x_n]$, input the vector to a Bi-LSTM model and the output is a vector $[h_1, h_2, \cdots, h_n]$ which is consists of $[l_1, l_2, \cdots, l_n]$ and $[r_1, r_2, \cdots, r_n]$ generated by the backward and forward LSTM model respectively. The $[l_1, l_2, \cdots, l_n]$ is generated according to

$$\begin{cases} X = \begin{bmatrix} l_{t-1} \\ x_t \end{bmatrix} \\ f_t = \sigma\left(W_f \cdot X + b_f\right) \\ i_t = \sigma\left(W_i \cdot X + b_i\right) \\ o_t = \sigma\left(W_o \cdot X + b_o\right) \\ C_t = f_t * C_{t-1} + i_t * \tanh\left(W_C \cdot X + b_C\right) \\ l_t = o_t * \tanh\left(C_t\right) \end{cases} \tag{2}$$

where f_t, i_t, o_t represent the forget gate, input gate, and output gate respectively, and W_f, W_i, W_o represent the weighted matrics during the training process, b_f, b_i, b_o represent the biases. σ is the sigmoid function and tanh is the tangent function. x_t is one of the input vector. The $[r_1, r_2, \cdots, r_n]$ is generated in the same way.

To obtain the weight of every hidden corresponding to each character in a domain name by computing the correlation coefficient between the character and the rest in a domain name. The self-attention process can be described as

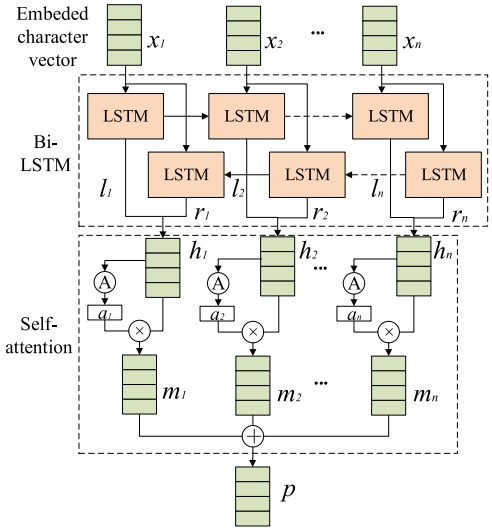


FIGURE 4. SA-Bi-LSTM architecture.

follows:

$$\begin{cases} h_t = l_t + r_t \\ e_t = \sigma\left(W_a \cdot h_t + b_a\right) \\ a_t = soft\max\left(e_t\right) \\ m_t = a_t \cdot h_t \\ p = \sum_{i=1}^{n} m_i \end{cases} \tag{3}$$

where h_t is the input of the attention layer, which is the output of the Bi-LSTM model. And W_t, W_x, W_a represent the weighted matrics during the training process, b_t, b_a represent the biases. The a_t makes up the A in Figure 4, which means the attention matrix. p is the output of the attention layer.

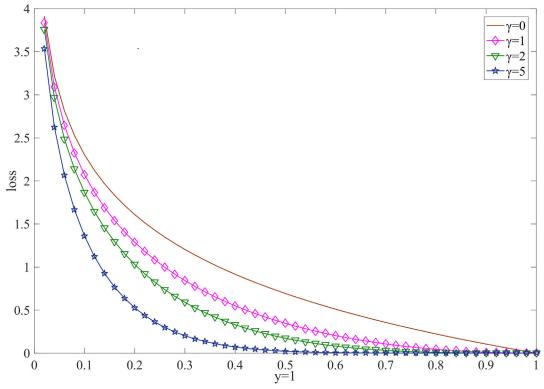
According to Equation 2 and Figure 4, LSTM takes the input as sequential input, and the next step is based on all the previous steps, so as to achieve the complete perception of the whole input. We regard the LSTM-based features as the global features, and the Bi-LSTM can extract bidirectional global features, the self-attention mechanism can make better attentioned global features. Overall, The proposed SA-Bi-LSTM can make full use of the advantages of Bi-LSTM and self-attention mechanism, and realize more effective character-level global feature extraction.

D. FOCAL LOSS FUNCTION

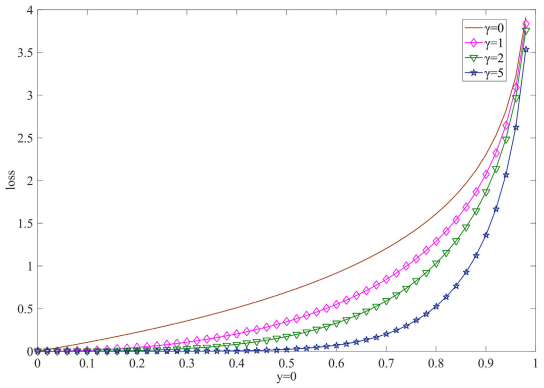
Cross entropy function is one of the most commonly used loss functions for most classification tasks. The formula of the binary classification cross-entropy function is

$$L = \begin{cases} -\log y', & y = 1 \\ -\log(1 - y'), & y = 0 \end{cases} \tag{4}$$

where y is the true value of sample and the y' is the predicted result of the classification model. For a positive sample, that is, when $y = 1$, the closer the y' gets to 1, the smaller is the loss value. For a negative sample, the closer the y' gets to 0,



(a) The curve of loss value when $y = 1$



(b) The curve of loss value when $y = 0$

FIGURE 5. Loss value of focal loss function with different value of γ.

the smaller is the loss value. When there are a large number of simple samples in the training set, the loss produced by these simple samples will cause the model may not be optimized to the optimal result. In order to reduce the loss of simple samples and make the model pay more attention to the classification of difficult samples, an improved cross-entropy loss function named focal loss was proposed in [34], Equation 5 denotes the keypoint of the improvement.

$$L_{fl} = \begin{cases} -(1-y')^{\gamma} \log y', & y = 1 \\ -y'^{\gamma} \log(1-y'), & y = 0 \end{cases} \tag{5}$$

where γ is a tunable parameter for it can adjust the changing intensity of loss value. The curve of loss value is shown as Figure 5.

Considering the unbalanced quantity of the benign domain names and DGA domain names, a fixed weight was introduced to the Equation 5, the complete formula for focal loss is shown as Equation 6.

$$L_{fl} = \begin{cases} -\alpha(1-y')^{\gamma} \log y', & y = 1 \\ -1-\alpha y'^{\gamma} \log(1-y'), & y = 0 \end{cases} \tag{6}$$

The DGAs domain names are with different intensities of anomaly according to different principles used by the algorithms. Take SDGA as an example, the domain names generated by it are closely related to the dictionary it uses. To improve the detection ability of SDGA and other hard-to-detect DGAs, we adopted the focal loss as the loss function in the training phase.

E. DETECTION SCHEME

With the proposed heterogeneous deep neural network framework and the introduced focal loss function, the SDGA detection scheme is as follows.

- Preprocessing phase: The characters that appear in domain names are analyzed and the total 87 different characters are determined. A character embedding model is trained and the pre-trained model is used for mapping characters to vectors to complete Character Embedding.
- Training phase: A sufficient number of benign and DGA domain names are collected as the training set, which is further divided into the training subset and validating subset. The labels of benign and DGA domain names are set as 0 and 1 respectively. The samples in the training subset are fed into HDNN for many epochs with the focal loss as the loss function. The samples in the validating subset are used to make the prediction driven by the trained model for every epoch. The predicted results are compared to the ground truth of the labels. When the accuracy of the predicted results on the validating subset is no longer improving at some epoch, the trained model will be selected as the ultimate model.
- Testing phase: The test samples are fed into the trained model, and the softmax activation outputs are obtained. A testing set is used to evaluate the detection accuracy. The detection threshold is set to 0.5, it is to say that if the prediction output is greater than 0.5, the input sample is determined as DGA, else the input sample is determined as benign.

The proposed scheme is mainly designed to make binary classification between SDGA domain names and benign domain names. It is worthy to point out that it also can be used to make multiclass classification among different DGA families, which will be exhibited by the following experiments.

IV. EXPERIMENT AND ANALYSIS

In this section, a series of experiments are performed and the corresponding results are analyzed in detail. In the beginning, the basic statistical measures for evaluating the proposed detection scheme are introduced. Then, the dataset for evaluating the detection effectiveness is introduced. Next, a comparative experiment to verify SDGAs detection performance is conducted between the proposed scheme and several influential algorithms proposed in recent years. Moreover, the proposed scheme is also tested on the traditional DGAs detection and classification task, the comparative experiment is also conducted to verify the superiority of the proposed scheme.

TABLE 1. The dataset of benign and SDGA domain names.

Classes	Description	Quantity
Benign	Benign domain names collected by Cisco	352,000
SDGA	Domain names generated by SDGA, and there are several types: **DNL1**: domain names generated from an English dictionary, and the corresponding number of characters is 1. **DNL2**: domain names generated from an English dictionary, and the corresponding number of characters is 2. **DNL3**: domain names generated from an English dictionary, and the corresponding number of characters is 3. **DNL4**: domain names generated from an English dictionary, and the corresponding number of characters is 4. **9ML1**: domain names generated from a dictionary composed of 9 million domain names, and the corresponding number of characters is 1. **500KL1**: domain names generated from a dictionary composed of 500 thousand domain names, and the corresponding number of characters is 1. **500KL2**: domain names generated from a dictionary composed of 500 thousand domain names, and the corresponding number of characters is 2. **500KL3**: domain names generated from a dictionary composed of 500 thousand domain names, and the corresponding number of characters is 3. The larger is the corresponding number of characters, the less intensity of anomaly of the generated domain names.	88,000

TABLE 2. Results on SDGA dataset for binary classification.

Detection scheme	precision	Recall	Accuracy	F1-Score
LSTM	0.9575	0.7286	0.8481	0.8275
CNN	0.9653	0.6641	0.8201	0.7869
CapsNet	0.9673	0.488	0.7358	0.6487
Parallel-CNN	**0.9686**	0.7144	0.8456	0.8223
LSTM-IM	0.9416	0.7721	0.8621	0.8485
Attention-based LSTM	0.9488	0.7489	0.8543	0.8371
HDNN (proposed)	0.9176	**0.8421**	**0.8833**	**0.8782**

A. EVALUATION METRICS

In order to evaluate the detection effectiveness of the proposed scheme, the following terms are used for determining the quality of the classification models:

- True Positive (TP) – the number of DGA domain names correctly classified to the DGA class.
- True Negative (TN) – the number of benign domain names correctly classified to the benign class.
- False Positive (FP) – the number of DGA domain names wrongly classified to the benign class.
- False Negative (FN) – the number of benign domain names wrongly classified to the DGA class.

Based on the aforementioned terms, the following most commonly used evaluation metrics are considered.

- Accuracy: It estimates the ratio of the correctly recognized connection records to the entire test dataset. It is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

- Precision: It estimates the ratio of the correctly identified AGDs to the total number of samples classified to AGD class. It is defined as

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

- Recall: It estimates the ratio of the correctly identified AGDs to the number of all AGDs. It is also called True Positive Rate (TPR). It is defined as

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

- F1-Score: It is the harmonic mean of Precision and Recall. It is defined as

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

- False Positive Rate (FPR): It estimates the ratio of the number of AGDs flagged as benign ones to the total number of samples classified to benign class. It is defined as

$$FPR = \frac{FP}{FP + TN} \tag{11}$$

- Receiver Operating Characteristics (ROC) curve: ROC is plotted based on the trade-off between the TPR on the y axis to FPR on the x axis across different thresholds. Area Under the ROC Curve (AUC) is the size of the area under the ROC curve used along with ROC as a comparison metric for the machine learning models. Generally, the AUC is higher, the model is better. The AUC is defined as

$$AUC = \int (TPR)\, d\,(FPR) \tag{12}$$

B. EXPERIMENT ON SDGA

Since the domain names generated by SDGAs are more difficult to detect for most existing detection models, a dataset is collected including 352000 benign domain names and 88,000 SDGA domain names. The proposed detection scheme is compared to LSTM-based scheme [17], basic CNN scheme [18], PCNN scheme [19], CapsNet scheme [21], LSTM-IM scheme [22], and attention-based LSTM scheme [23]. It is a pity that because of some key details are not clear, the implemented result of [24] is also insufficient. Due to fairness, it is not taken into consideration for making the comparison.

1) DATASET DESCRIPTION

The benign samples are all from the top 1 million domain names collected by Cisco [35], the reason we select samples from Cisco Umbrella instead of Alexa [36] is that Cisco collects the domain names for various of internet services while

TABLE 3. Detection accuracy on every type of domain names in SDGA dataset.

Detection scheme	benign	DNL1	DNL2	DNL3	DNL4	9ML1	500KL1	500KL2	500KL3
LSTM	0.9709	0.5693	0.4745	0.4112	0.5009	0.9115	0.9785	0.9565	0.8249
CNN	0.9784	0.5166	0.3975	0.3280	0.4322	0.8938	0.9825	0.9610	0.7959
CapsNet	**0.9829**	0.2060	0.1465	0.1248	0.1499	0.7498	0.9766	0.9180	0.5988
Parallel-CNN	0.9766	0.5649	0.4908	0.4069	0.4953	0.9287	0.9894	**0.9741**	0.8267
LSTM-MI	0.9482	0.6247	0.5843	0.5908	0.6083	0.9112	0.9856	0.9682	0.8564
Attention-based LSTM	0.9589	0.6079	0.5804	0.5316	0.5848	0.9173	0.9855	0.9659	0.8432
HDNN(proposed)	0.9244	**0.7553**	**0.7364**	**0.6856**	**0.7401**	**0.9218**	**0.9874**	0.9702	**0.9141**

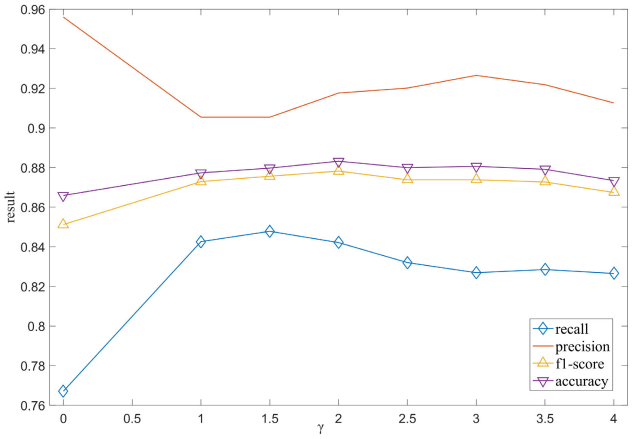


FIGURE 6. Detection results with different value of γ.

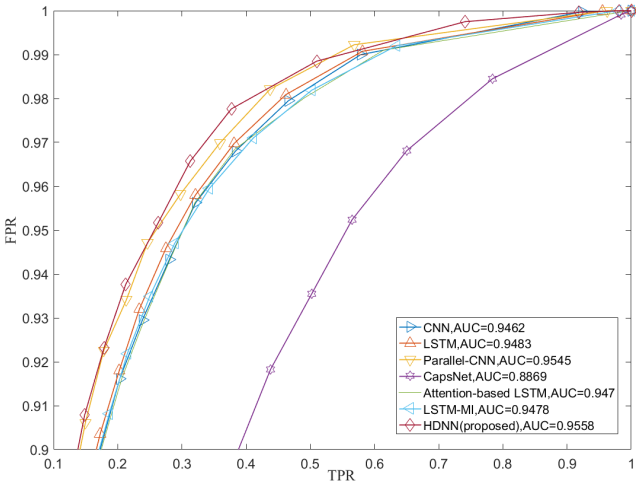


FIGURE 7. ROC curve of the detection results on SDGA.

Alex collects domain names only for web service. The SDGA domain names are acquired from the authors of [2]. Since the length of any SDGA domain name is limited to larger than 4 and less than 11, the benign samples are correspondingly selected in the same range. The detail of the dataset is shown in Table 1. The number of benign domain names and SDGA domain names is 352,000 and 88,000 respectively. Total 8000 benign domain names and 8000 SDGA domain names are randomly selected from the dataset as the test set, and the rest is divided into the training set and verification set according to the ratio of 9:1. The detailed description of the dataset is as shown in Table 1.

2) PARAMETER SELECTION

The γ in focal loss function represents the intensity of the loss caused by the difficult and simple samples. The α is the weight of SDGA samples. Since the ratio of benign domain names to SDGA domain names in the training set is 4:1, α is set to 0.8. Since the value of γ is related to the difficulty of the sample, the optimal value cannot be calculated, so we choose different values of γ and select the optimal γ through experiments. The experimental results are shown in Figure 6.

It can be seen that without focal loss function, the f1-score of the proposed scheme for SDGA detection is 0.8512 and the accuracy is 0.8659. The use of focal loss significantly improved detection accuracy and f1-score. The comparative experimental results denote that when $\gamma = 2$ and $\alpha = 0.8$, the proposed scheme achieves the optimal detection accuracy and f1-score which are 0.8833 and 0.8782 respectively.

3) COMPARATIVE EXPERIMENTAL RESULTS

To benchmark the detection effectiveness on SDGA domain names, the proposed scheme is compared to a total of 6 influential algorithms. The results are shown in Table 2. The detection performance details of each type of SDGAs domain names are listed in Table 3. Figure 7 shows the ROC of all the detection schemes.

It can be seen from the result that the proposed scheme performs best in an overall view. As a contrast, the CapsNet scheme performs worst with the Recall result being only 0.488, which denotes that the CapsNet network cannot perform as well as CNN or LSTM network on the SDGA detection task. The basic LSTM scheme performs better than the basic CNN scheme on detection accuracy, while performs close to the PCNN scheme, which denotes that the parallel architecture of CNN is more suitable for the SDGA detection task than the other two. The attention-based LSTM scheme performs better than the basic LSTM scheme, which denotes that the attention mechanism is effective for enhancing the ability of LSTM. With the use of IPCNN and SA-Bi-LSTM, as well as the training process optimized by the focal loss function, the average detection accuracy of the proposed scheme performs significantly better than all other detection schemes. On the view of the detail of detection results on each type of SDGA domain names as shown in Table 3, the detection accuracies of DNL1, DNL2, DNL3, and DNL4 have remarkably improved 13.06%, 15.21%, 9.48%, 13.18% respectively compared to the state-of-the-art

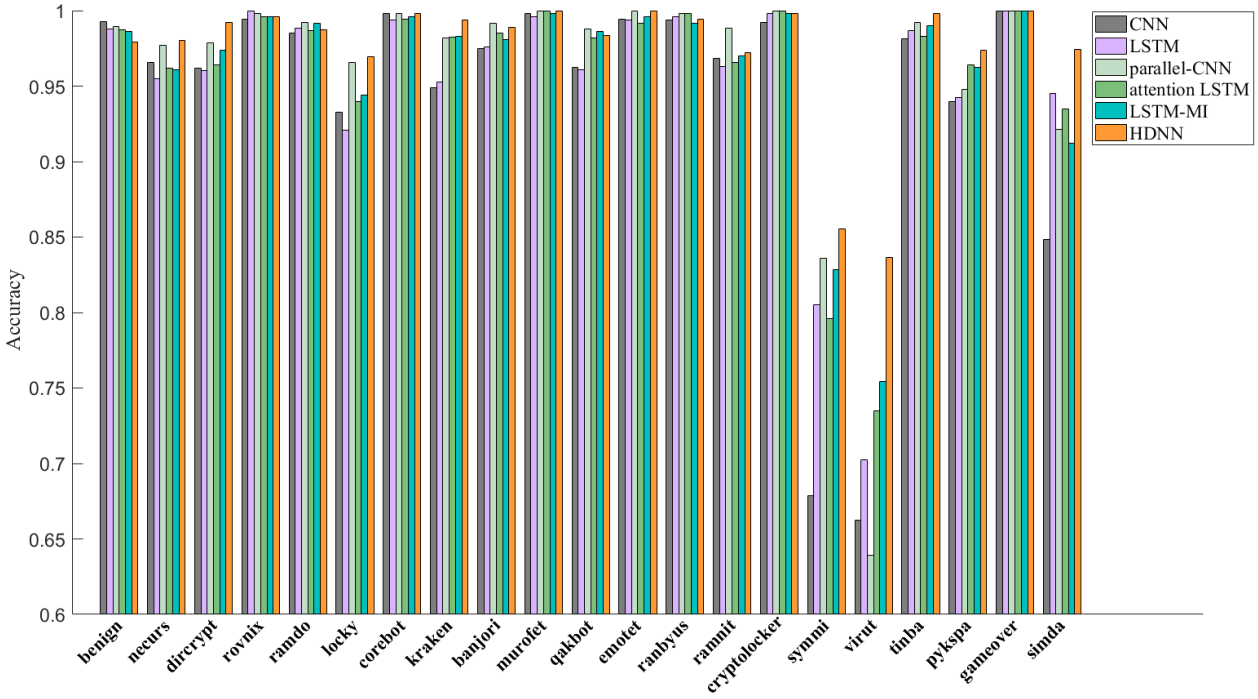


FIGURE 8. Classification accuracy of benign and 20 types of traditional DGA domain name.

detection scheme LSTM-MI. The detection accuracies on the other four types of SDGA are also got improved, while the trade-off is that the detection accuracy on benign domain names only decreases by 2.38%. The binary classification results on benign domain names and SDGA domain names denote that the proposed scheme can detect SDGA with state-of-the-art accuracy, especially for those hard-to-detect SDGA types.

C. EXPERIMENTS ON TRADITIONAL DGA

For better evaluation of the proposed model, experiments are conducted on various traditional DGAs. The training and testing sets are composed of benign domain names and a total of 20 types of DGA domain names collected from public resources. Binary and multiclass classification experiments are conducted among the proposed scheme and LSTM scheme [17], basic CNN scheme [18], parallel-CNN scheme [19], LSTM-IM [22], and attention-based LSTM scheme [23]. The detection result of CapsNet scheme [21] is obviously worse than other methods for comparison, so it's experimental results are not listed in the following discussion.

1) DATASET DESCRIPTION

The benign samples are from the top 1 million domain names dataset collected by Cisco [35], and the first 400,000 domain names are selected to make the training dataset. The DGA samples are from the dataset collected by 360netlab [37] which is a DGA dataset updated daily, a total of 20 types of DGAs with sufficient quantity of samples are chosen and 5000 samples of each DGA category are selected to make the training dataset. 8000 of the benign domain names and 8000 of the DGA domain names are chosen to be the testing set. The dataset description is as shown in Table 4.

TABLE 4. Dataset description of benign and DGA domain names.

Classes	Description	Quantity
Benign	Benign domain names collected by Cisco	400,000
DGA	Domain names generated by DGA families including: Gameover, Murofet, Dircrypt, Tinba, Necurs, Ramdo, Ranbyus,Emotet, Cryptolocker, Corebot,Banjori,Qakbot, Rovnix, Kraken, Ramnit, Locky, Pykspa, Simda, Symmi,and Virut.	100,000

TABLE 5. Results on traditional DGA dataset for binary classification.

Detection scheme	Precision	Recall	Accuracy	F1-Score
LSTM	0.9853	0.9536	0.9697	0.9692
CNN	**0.9912**	0.9406	0.9661	0.9652
Parallel-CNN	0.9898	0.9574	0.9738	0.9733
LSTM-MI	0.9873	0.9626	0.9751	0.9748
Attention-based LSTM	0.9852	0.9600	0.9728	0.9724
HDNN (proposed)	0.9793	**0.9751**	**0.9773**	**0.9772**

2) BINARY CLASSIFICATION EXPERIMENT

The binary classification experiment is to perform to distinguish DGA domain names for benign domain names. The experimental results are compared among the proposed scheme and the chosen 5 influential and effective schemes. Table 5 shows the overview results of the total 6 DGA detection schemes including our proposed scheme. The contrast classification results of each type of DGAs are shown in Figure 8. Figure 9 shows the ROC results of all the schemes.

TABLE 6. Multiclass classification results of LSTM, CNN, parallel CNN on traditional DGA.

	LSTM			CNN			Parallel-CNN		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
tinba	0.80	0.99	0.88	0.78	1.0	0.87	0.68	0.98	0.80
gameover	0.99	0.95	0.97	1.0	0.95	0.97	1.0	0.96	**0.98**
Kraken	0.60	0.68	0.64	0.67	0.85	0.75	0.82	0.83	0.82
banjori	0.98	0.98	0.98	1.00	0.98	0.99	0.99	0.98	0.98
cryptolocker	0.78	0.60	0.68	0.89	0.87	0.88	0.90	0.87	0.88
symmi	0.93	0.99	0.96	0.91	1.0	0.95	0.96	0.97	0.97
pykspa	0.94	0.78	0.85	0.90	0.76	0.83	0.96	0.74	0.83
virut	0.94	0.98	0.96	0.93	0.98	0.96	0.86	0.96	0.91
corebot	0.99	1.00	0.99	1.0	1.0	1.0	1.0	1.0	1.0
locky	0.21	0.32	0.26	0.27	0.19	0.22	0.23	0.36	0.28
ramdo	0.98	0.99	0.99	0.94	0.99	0.97	0.97	1.0	0.98
qakbot	**0.74**	0.23	0.35	0.67	0.29	0.40	0.70	0.23	0.35
ramnit	0.20	0.23	0.22	**0.29**	**0.36**	**0.32**	0.20	0.07	0.10
simda	0.98	0.99	0.98	0.98	1.0	0.99	0.97	0.99	0.98
murofet	0.79	0.76	0.78	0.75	0.87	0.81	0.74	0.79	0.76
emotet	0.69	0.97	0.81	0.73	0.94	0.82	0.39	0.86	0.54
rovnix	1.00	0.99	0.99	1.00	0.99	1.0	1.00	0.99	0.99
ranbyus	**0.60**	0.96	**0.74**	0.59	0.95	0.73	0.44	0.57	0.50
dircrypt	**0.55**	**0.34**	**0.42**	0.48	0.30	0.37	0.46	0.28	0.35
necurs	**0.48**	0.17	0.25	0.48	0.18	0.26	0.42	0.14	0.22
Macro-averaging	0.7589	0.7451	0.7345	0.7623	0.7725	0.7541	0.7345	0.7289	0.7118

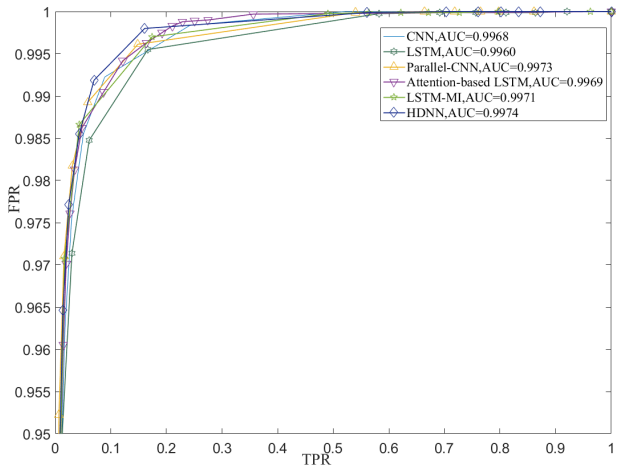


FIGURE 9. ROC curve of the detection results on traditional DGA.

The overview results denote that all the 6 schemes have achieved high detection accuracy, while the proposed scheme is also slightly better than the other 5 existing schemes. The classification accuracies of basic CNN and LSTM are close to each other. And the PCNN scheme performs better than the basic CNN and LSTM scheme. The attention-based LSTM scheme performs unexpectedly a little bit worse than the PCNN scheme but still better than the basic CNN and LSTM schemes. The LSTM-MI scheme achieves the best classification result among the existing 5 schemes. Since having paid more attention to the hard-to-detect DGA domain names, the recall results of the proposed scheme are significantly higher than the 5 existing schemes. The proposed scheme achieves the best results if only considering the average accuracy and F1-score indices. Figure 8 denotes that the proposed scheme can significantly improve the classification accuracy of some hard-to-detect DGA families. It is shown in the contrast results that for most DGA families, the detection results are close to each other with high accuracy, while for several hard-to-detect DGA families such as Symmi, Virut, and Simda, the classification accuracy of the proposed scheme is significantly higher than other 5 existing schemes. The binary classification results denote that with the use of IPCNN and SA-Bi-LSTM architecture, the proposed HDNN scheme achieves the state-of-the-art classification result on benign domain names and traditional DGA domain names, especially, with the use of focal loss function, the proposed scheme significantly improves the detection accuracy of hard-to-detect DGA families.

3) MULTICLASS CLASSIFICATION EXPERIMENT

As mentioned in [22] and [23], binary classification can help to recognize if a domain name is malicious. After that, multiclass classification can help to identify the specific DGA family that a domain name belongs to. To benchmark the multiclass classification ability of the proposed scheme, a dataset consists of 20 DGA families with 5000 samples in each family is selected to be the training set, and 500 samples are selected randomly from each family to be the testing set. Table 6 and Table 7 show the classification results of the proposed scheme and the other 5 existing schemes for comparison. In the experiment, the indices, precision, recall F1-score are chosen as the evaluation metrics.

It can be seen from Table 6 and Table 7 that different from the result of binary classification, the basic CNN scheme performs better than the basic LSTM scheme, as well as the parallel CNN scheme. The f1-score of the basic scheme is 0.7541, significantly better than the 0.7345 of the basic LSTM scheme and 0.7118 of parallel CNN scheme. Considering the attention-based LSTM and LSTM-MI scheme, the classification results are also worse

TABLE 7. Multiclass classification results of attention-based LSTM, LSTM-MI, HDNN on traditional DGA.

	Attention-based LSTM			LSTM-MI			HDNN(proposed)		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
tinba	0.78	0.99	0.87	0.78	1.0	0.87	**0.82**	0.99	**0.90**
gameover	0.96	0.94	0.95	0.99	0.93	0.96	**1.00**	0.95	0.97
kraken	0.53	0.74	0.62	0.47	0.82	0.59	**0.85**	**0.87**	**0.86**
banjori	0.97	0.98	0.98	0.98	0.99	0.98	0.99	0.98	**0.99**
cryptolocker	0.76	0.64	0.69	0.79	0.51	0.62	**0.96**	**0.89**	**0.92**
symmi	0.95	1.0	0.97	0.96	0.98	0.97	0.97	0.96	0.96
pykspa	0.87	0.77	0.82	0.93	0.78	0.85	0.93	0.74	0.82
virut	0.91	0.95	0.93	0.95	0.98	0.96	0.89	0.98	0.93
corebot	1.0	1.0	1.0	1.00	0.99	0.99	1.0	1.0	1.0
locky	0.21	0.12	0.15	0.23	0.21	0.22	**0.27**	**0.48**	**0.34**
ramdo	0.95	0.99	0.97	0.98	0.99	0.98	0.98	1.0	0.99
qakbot	0.59	0.32	0.42	0.71	0.26	0.38	0.64	**0.38**	**0.47**
ramnit	0.23	0.19	0.21	0.25	0.28	0.26	0.35	0.12	0.18
simda	0.99	0.98	0.98	0.97	0.99	0.98	0.97	1.0	0.98
murofet	0.69	0.85	0.76	0.71	0.88	0.79	**0.81**	**0.89**	**0.85**
emotet	0.68	0.94	0.79	0.71	0.90	0.79	**0.73**	**0.96**	**0.83**
rovnix	1.0	1.0	1.0	1.0	1.0	1.0	1.00	0.99	1.0
ranbyus	0.57	0.92	0.71	0.56	0.91	0.69	0.57	**0.97**	0.72
dircrypt	0.33	0.21	0.26	0.50	0.27	0.35	0.45	0.28	0.35
necurs	0.34	**0.25**	**0.29**	0.40	0.13	0.19	0.39	0.18	0.25
Macro-averaging	0.7160	0.7388	0.7183	0.7431	0.7408	0.7234	**0.7780**	**0.7802**	**0.7653**

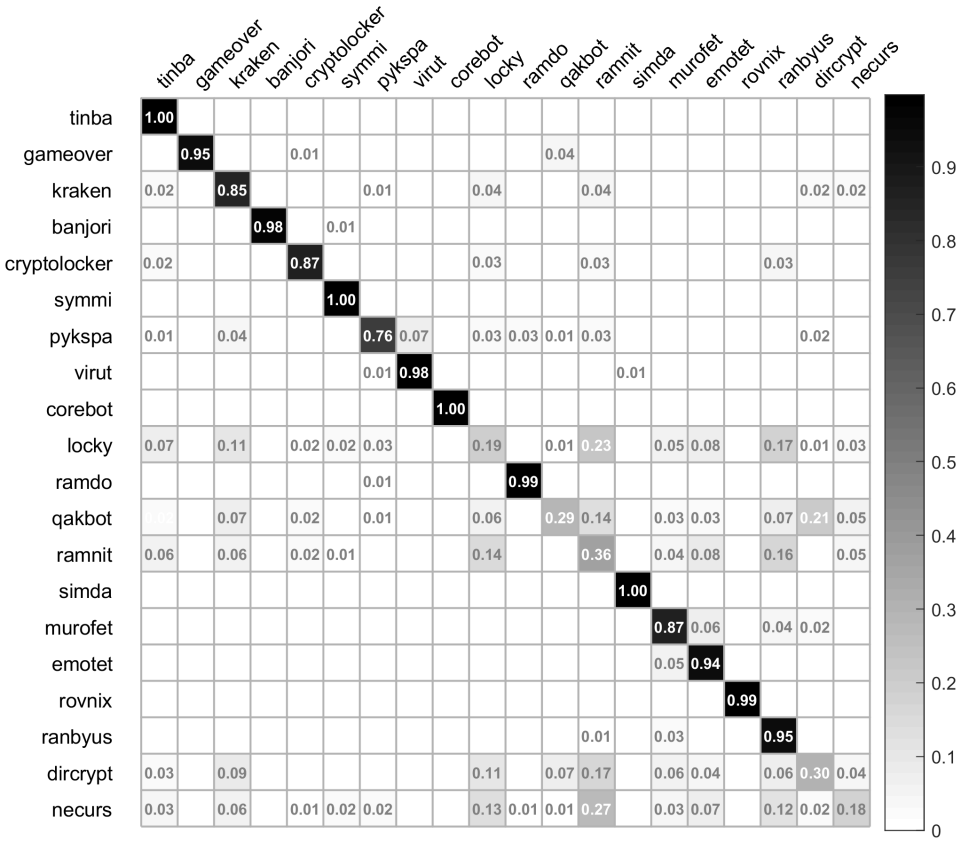


FIGURE 10. Confusion matrix related to the DGA families classified by the proposed HDNN scheme.

than the basic CNN scheme, the f1-scores of them get only 0.7183 and 0.7234 respectively, and even are worse than the basic LSTM scheme. The beyond imagination result denotes that the multiclass classification task is different from the binary classification. However, with the integration of several heterogeneous deep neural network architectures, the proposed HDNN architecture has achieved state-of-the-art multiclass classification results with 0.7653 of f1-score which is significantly higher than the other 5 schemes. As shown in Figure 10, the proposed HDNN scheme can achieve

effective classification results on 15 of the 20 DGA families. The Qakbot, Rammit, Dircrypt, and Necurs are tending to be classified as Locky. The reason is that the collected DGA families originally come from some real-world malicious botnet or Trojan, these malwares have the generator which provides uniform distribution over the letters [22]. It will lead to confusion in multiclass classification. Overall, the proposed HDNN scheme is effective for the multiclass classification of DGA families, and the classification result is better than the 5 existing schemes.

V. CONCLUSION

In this study, we proposed a heterogeneous deep neural network framework for detecting and classifying domain names generated by stealthy domain generation algorithms and other real-world DGA domain names. The proposed HDNN framework contains an IPCNN architecture and an SA-Bi-LSTM architecture. The proposed IPCNN can be used for extracting multi-scale local features from a domain name, meanwhile, the proposed SA-Bi-LSTM can extract bidirectional global features. Besides the more sophisticated feature extraction scheme, the focal loss function is introduced to mitigate the imbalance of the quantity and difficulty in the training samples. To benchmark the proposed scheme, a serial of experiments are conducted. Firstly, a binary classification experiment is conducted on SDGAs, and the detection results are compared among the proposed scheme and other 6 influential deep-learning-based schemes. Then the binary and multiclass classification experiments are conducted on benign domain names and 20 types of domain names generated by different DGA families. The detection and multiclass classification results are compared among the proposed scheme and other 5 influential deep-learning-based schemes. The comparative experiments denote that the proposed scheme has achieved state-of-the-art performance both on SDGAs and traditional DGAs detection tasks, and also on the traditional DGAs multiclass classification tasks.

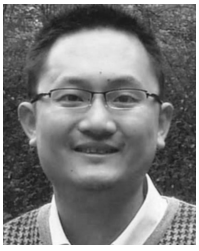
Although the proposed scheme has achieved better results on the detection of SDGA, the average detection accuracy is still not higher than 0.9. The future work on the detection of SDGA domain names needs to be carried out from more perspectives such as the side information of DNS request behaviors and so on. Another deficiency of the proposed scheme is that for extracting more comprehensive and robust character-level features from domain names, the proposed scheme is computationally intensive, which will limit its implementation in real-world scenarios. The next step of our work will focus on light-weight well effective deep neural network architecture.

REFERENCES

[1] A. K. Sood and S. Zeadally, "A taxonomy of domain-generation algorithms," *IEEE Secur. Privacy*, vol. 14, no. 4, pp. 46–53, Jul. 2016.
[2] Y. Fu, L. Yu, O. Hambolu, I. Ozcelik, B. Husain, J. Sun, K. Sapra, D. Du, C. Beasley, and R. Brooks, "Stealthy domain generation algorithms," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1430–1443, Jun. 2017.
[3] H. Crawford and J. Aycock, "Kwyjibo: Automatic domain name generation," *Softw., Pract. Exper.*, vol. 38, no. 14, pp. 1561–1567, Nov. 2008.
[4] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proc. 10th Annu. Conf. Internet Meas. (IMC)*. New York, NY, USA: Association for Computing Machinery, 2010, pp. 48–61, doi: 10.1145/1879141.1879148.
[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, Oct. 2012.
[6] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Tracking and characterizing botnets using automatically generated domains," 2013, *arXiv:1311.5612*. [Online]. Available: https://arxiv.org/abs/1311.5612
[7] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*, Cham, Switzerland: Springer, 2014, pp. 192–211.
[8] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, pp. 1–28, Apr. 2014, doi: 10.1145/2584679.
[9] D. L. Schales, J. Jang, T. Wang, X. Hu, D. Kirat, B. Wuest, and M. P. Stoecklin, "Scalable analytics to detect DNS misuse for establishing stealthy communication channels," *IBM J. Res. Develop.*, vol. 60, no. 4, pp. 3:1–3:14, Jul. 2016.
[10] M. Zago, M. Pérez, and G. M. Pérez, "Scalable detection of botnets based on DGA: Efficient feature discovery process in machine learning techniques," *Soft Comput.*, Jan. 2019.
[11] M. Mowbray and J. Hagen, "Finding domain-generation algorithms by looking at length distribution," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops*, Nov. 2014, pp. 395–400.
[12] J. Raghuram, D. J. Miller, and G. Kesidis, "Unsupervised, low latency anomaly detection of algorithmically generated domain names by generative probabilistic modeling," *J. Adv. Res.*, vol. 5, no. 4, pp. 423–433, Jul. 2014.
[13] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, "Detecting DGA malware using NetFlow," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1304–1309.
[14] T.-D. Nguyen, T.-D. Cao, and L.-G. Nguyen, "DGA botnet detection using collaborative filtering and density-based clustering," in *Proc. 6th Int. Symp. Inf. Commun. Technol. (SoICT)*, 2015, pp. 203–209.
[15] T. Wang, X. Hu, J. Jang, S. Ji, M. Stoecklin, and T. Taylor, "BotMeter: Charting DGA-botnet landscapes in large networks," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 334–343.
[16] Y. Shi, G. Chen, and J. Li, "Malicious domain name detection based on extreme machine learning," *Neural Process. Lett.*, vol. 48, no. 3, pp. 1347–1357, Dec. 2018.
[17] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," *CoRR*, vol. abs/1611.00791, 2016. [Online]. Available: http://arxiv.org/abs/1611.00791
[18] B. Yu, D. L. Gray, J. Pan, M. D. Cock, and A. C. A. Nascimento, "Inline DGA detection with deep networks," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 683–692.
[19] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, "Character level based detection of DGA domain names," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[20] J. Saxe and K. Berlin, "EXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys," *CoRR*, vol. abs/1702.08568, 2017. [Online]. Available: http://arxiv.org/abs/1702.08568
[21] D. S. Berman, "DGA CapsNet: 1D application of capsule networks to DGA detection," *Information*, vol. 10, no. 5, p. 157, Apr. 2019.
[22] 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, Nov. 2017.
[23] Y. Qiao, B. Zhang, W. Zhang, A. K. Sangaiah, and H. Wu, "DGA domain name classification method based on long short-term memory with attention mechanism," *Appl. Sci.*, vol. 9, no. 20, p. 4205, Oct. 2019.
[24] C. Xu, J. Shen, and X. Du, "Detection method of domain names generated by DGAs based on semantic representation and deep neural network," *Comput. Secur.*, vol. 85, pp. 77–88, Aug. 2019.
[25] Y. Li, K. Xiong, T. Chin, and C. Hu, "A machine learning framework for domain generation algorithm-based malware detection," *IEEE Access*, vol. 7, pp. 32765–32782, 2019.

[26] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[27] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, vol. 25, Jan. 2012, pp. 1097–1105.
[28] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Aug. 2014, pp. 1–6.
[29] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, Sep. 2015, pp. 649–657.
[30] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
[31] S. Hochreiter and J. J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Dec. 1997.
[32] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 207–212.
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Jun. 2017, pp. 5998–6008.
[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[35] Cisco. (2020). *Umbrella Popularity List*. Accessed: Jan. 8, 2020. [Online]. Available: http://s3-us-west-1.amazonaws.com/umbrella-static/index.html
[36] Alexa. (2020). *Top Site on the Web*. Accessed: Jan. 8, 2020. [Online]. Available: https://www.alexa.com/topsites
[37] 360netlab. (2020). *DGA*. Accessed: Jan. 8, 2020. [Online]. Available: https://data.netlab.360.com/dga/

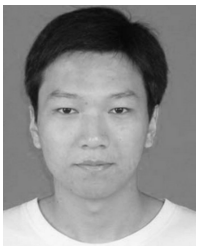
LUHUI YANG received the B.S. degree in automation from the Nanjing University of Science and Technology, Nanjing, in 2013. He is currently pursuing the Ph.D. degree with the Nanjing University of Information Science and Technology. His research interest includes deep learning and its applications in network traffic analysis.



GUANGJIE LIU (Member, IEEE) received the B.S. degree in electrical and computer engineering and the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology, Nanjing, in 2002 and 2007, respectively. He is currently an Associate Professor with the Nanjing University of Information Science and Technology. His research interests include multimedia systems and deep learning.

YUEWEI DAI received the B.S. and M.S. degrees in system engineering from the East China Institute of Technology, in 1984 and 1987, respectively, and the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology, in 2002. He is currently a Professor with the Nanjing University of Information Science and Technology. His research interests include multimedia security, system engineering theory, and network security.

JINWEI WANG (Member, IEEE) received the B.S. degree in automatic control from the Inner Mongolia University of Technology, in 2000, and the Ph.D. degree in information security from the Nanjing University of Science and Technology, in 2007, where he was a Visiting Scholar with the Service Anticipation Multimedia Innovation (SAMI) Laboratory, France Telecom Research and Development Center, Beijing, in 2006. He was a Senior Engineer with the 28th Research Institute, CETC, from 2007 to 2010. He was a Visiting Scholar with the New Jersey Institute of Technology, NJ, USA, from 2014 to 2015. He is currently a Professor with the Nanjing University of Information Science and Technology. He has published over 50 articles. He has hosted and participated in more than ten projects. His research interests include multimedia copyright protection, multimedia forensics, multimedia encryption, and data authentication.



JIANGTAO ZHAI received the B.S. degree in electrical and computer engineering and the M.S. and Ph.D. degrees in control science and engineering from the Nanjing University of Science and Technology, Nanjing, in 2007, 2009, and 2013, respectively. He is currently an Associate Professor with the Nanjing University of Information Science and Technology. His research interests include multimedia communication and wireless sensor networks.