

Article

DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network

Jisheng Tang ¹, Yiling Guan ¹ , Shenghui Zhao ^{1,*}, Huibin Wang ¹ and Yinong Chen ² 

¹ School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China; 2021212074@mail.chzu.edu.cn (J.T.); 2021211908@mail.chzu.edu.cn (Y.G.); wanghuibin@chzu.edu.cn (H.W.)

² School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85287, USA; yinong@asu.edu

* Correspondence: zsh@chzu.edu.cn

Abstract: Botnets pose a significant challenge in network security by leveraging Domain Generation Algorithms (DGA) to evade traditional security measures. Extracting DGA domain samples is inherently complex, and the current DGA detection models often struggle to capture domain features effectively when facing limited training data. This limitation results in suboptimal detection performance and an imbalance between model accuracy and complexity. To address these challenges, this paper introduces a novel multi-scale feature fusion model that integrates the Transformer architecture with the Rapid Selective Kernel Network (R-SKNet). The proposed model employs the Transformer's encoder to couple the single-domain character elements with the multiple types of relationships within the global domain block. This paper proposes integrating R-SKNet into DGA detection and developing an efficient channel attention (ECA) module. By enhancing the branch information guidance in the SKNet architecture, the approach achieves adaptive receptive field selection, multi-scale feature capture, and lightweight yet efficient multi-scale convolution. Moreover, the improved Feature Pyramid Network (FPN) architecture, termed EFAM, is utilized to adjust channel weights for outputs at different stages of the backbone network, leading to achieving multi-scale feature fusion. Experimental results demonstrate that, in tasks with limited training samples, the proposed method achieves lower computational complexity and higher detection accuracy compared to mainstream detection models.

Keywords: botnet; domain generation algorithm; transformer model; depthwise separable convolution



Citation: Tang, J.; Guan, Y.; Zhao, S.; Wang, H.; Chen, Y. DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network. *Electronics* **2024**, *13*, 4982. <https://doi.org/10.3390/electronics13244982>

Academic Editors: Wentao Li, Huiyan Zhang, Tao Zhan and Chao Zhang

Received: 16 September 2024

Revised: 28 November 2024

Accepted: 11 December 2024

Published: 18 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous development and iteration of Internet technology, people's lifestyles have become more convenient. However, this progress has also led to a rise in both the number and complexity of malware, giving rise to black-market industries and cybercriminal activities, such as botnets and phishing. Attackers use malware to infect networks, issuing commands to compromised machines or botnets to steal sensitive data, launch Distributed Denial of Service (DDoS) attacks, and carry out other cyberattacks, posing significant challenges to global network security. To evade security checks and maintain control over compromised hosts for extended periods, thereby maximizing economic benefits, modern malware attackers frequently integrate DGA into malware. DGA enables the rapid generation of numerous dynamic domains to connect with Command and Control (C&C) servers. The malware client and attacker both use the same DGA algorithm to generate a list of candidate domains. When preparing for an attack, a subset of these domains is registered to establish communication. Due to the high randomness and dynamic nature of DGA-generated domains, malware can rotate domains throughout its attack cycle, making it nearly impossible for network security tools to block all malicious domains via blacklists. Consequently, accurately detecting DGA-generated domains and disrupting the communication between botnets and C&C servers is crucial for malware defense.

In recent years, the rapid development of deep learning has demonstrated its vast potential across various fields, leading researchers to explore its application in DGA detection. Compared to machine learning methods based on domain traffic characteristics and manually defined statistical features, deep learning eliminates the need for manual feature extraction and significantly improves the accuracy of DGA detection. Therefore, deep learning has become a key technology for enhancing the efficiency and precision of DGA detection.

Current DGA detection models commonly use Transformer encoders to capture complex long-range dependencies between characters within domain names. The resulting feature sequences are then processed through a multi-branch convolutional network architecture to extract features from different receptive fields, followed by a Bidirectional Long Short-Term Memory (Bi-LSTM) network to obtain bidirectional sequence features. The final classification is achieved through feature fusion. However, this network architecture still has several weaknesses:

- While parallel multi-scale convolution benefits from multiple receptive field features, the information between branches remains independent and lacks effective coupling. Additionally, the features extracted from each receptive field are complex, and a simple parallel structure cannot adequately capture comprehensive features across all scales, reducing detection accuracy. Furthermore, although ordinary convolution can extract more comprehensive features, it results in high training costs.
- The outputs of each stage in the backbone network are merely concatenated, which prevents the extraction of hierarchical features at different stages. This leads to feature redundancy and negatively impacts detection accuracy.

To address these weaknesses, this paper proposes R-SKNet and EFAM to deeply extract domain features, enhancing model performance through rapid adaptive selection of receptive field scales and efficient feature fusion across different stages. The contributions of this paper can be summarized as follows:

- This paper proposes R-SKNet, which allows for rapid adaptive selection of multi-scale receptive fields within the network, effectively coupling feature information across branches.
- This paper proposes an improved feature fusion strategy, EFAM, which facilitates efficient multi-scale feature fusion and enhances hierarchical features. It further refines the convolution strategy to capture spatial focus points better.
- Based on Transformer, R-SKNet, and EFAM, this paper proposes a DGA domain detection framework, TransFlexNet, which takes the advantages and the strengths of these three models. It outperforms mainstream models under both large and small sample conditions.

With the growing complexity of network threats, existing DGA detection methods often struggle with effective feature extraction and fusion, making it difficult to counter the evolving tactics of malware attacks. To address these challenges, this paper introduces the TransFlexNet detection framework, designed to improve the precision and efficiency of DGA detection and strengthen network security defenses.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 describes the proposed DGA detection model. Section 4 discusses and analyzes the datasets and experimental results. Section 5 concludes the paper with future directions.

2. Related Work

In today's cybersecurity field, DGA detection has become a crucial research topic. With the continuous advancement and application of machine learning and deep learning technologies, these advanced techniques have been integrated into traditional algorithmic models to address the challenges in DGA detection. These challenges include the need for continuous updates to detection methods and models as attackers use various algorithms, the difficulty of feature extraction due to the high similarity between random and legitimate

domain names, and the impact of dataset bias on the generalization ability of models. Building on these points, we analyze existing DGA detection techniques.

2.1. Machine-Learning-Based Detection Methods

Traditional machine learning methods rely on large amounts of annotated data, which are difficult to obtain in real-world applications. To address this issue, Yan et al. [1] proposed a semi-supervised learning strategy that constructs domain relationship graphs and utilizes unlabeled data to enhance the identification of DGA family evolution and new variants. This approach maintains high efficiency and accuracy, even with limited labeled resources. Anand et al. [2] integrated C5.0, Random Forest, Gradient Boosting, and CART, and used 44 comprehensive lexical and statistical features, significantly improving the recognition accuracy of DGA domain names and the robustness of the model. Satoh et al. [3] conducted word-level feature analysis, using specific lexical generation patterns to identify dictionary-based DGA malware, highlighting the importance of semantic and linguistic features. Vranken et al. [4] applied TF-IDF techniques to analyze n-grams in domain names, reducing dependence on large-scale annotated data and improving the accuracy of DGA domain name detection.

Although these methods have achieved breakthroughs in specific scenarios, they still have limitations in terms of feature dependence and generalization capabilities. Semi-supervised learning heavily relies on the quality of labeled data, and its performance degrades when the labeling quality is poor. Anand's integrated model relies on complex and time-consuming feature engineering, which does not take the advantages of certain critical features. TF-IDF relies on known vocabularies and performs poorly when facing the novel DGA variants. In contrast, deep learning demonstrates greater advantages through automated feature extraction.

2.2. Deep-Learning-Based Detection Methods

Deep learning techniques offer new solutions to problems in machine learning. Yang et al. [5] developed the N-Trans parallel detection model, which combines the N-gram algorithm and the Transformer model. By adding flag bits, it effectively extracts domain name features, allowing precise differentiation between legitimate and malicious domain names. Namgung et al. [6] proposed an integrated model combining BiLSTM and CNN, utilizing an attention mechanism to more effectively learn both local and global information in domain name sequences, significantly enhancing DGA domain detection performance. Shahzad et al. [7] developed an RNN-based DGA domain classifier that does not require context information or manual features. By analyzing a dataset of over 2 million domain names, they demonstrated the effectiveness of the RNN architecture. Liang et al. [8] found that DGA detection models are sensitive to domain name length and proposed a heterogeneous DGA detection model based on CNN and Random Forest, optimized for domain names of different lengths. Qi et al. [9] introduced the CNN-LSTM model, which combines CNN and LSTM to extract N-gram features through CNN and process them with LSTM, effectively classifying and predicting dictionary-based DGA domain names. Jiang et al. [10] proposed the CNN-GRU-Attention model, which uses CNN to extract spatial features, GRU to extract temporal features, and an attention mechanism to improve detection accuracy.

The aforementioned deep learning methods have achieved promising detection results. However, their use of standard parallel convolution in feature extraction fails to capture comprehensive multi-scale features and incurs high training costs. To address these issues, this paper introduces ECA, which computes attention weights without dimensionality reduction, thereby reducing the model's parameter count.

2.3. Detection Methods Based on Additional Mechanisms and Other Algorithms

In certain scenarios, incorporating additional modules into deep learning frameworks can enhance performance. Ren et al. [11] proposed the ATT-CNN-BiLSTM framework,

which combines CNN and BiLSTM with an attention mechanism to extract and optimize domain name features effectively. Pan et al. [12] improved word embedding methods by enhancing character features through the extraction of semantic features and adding additional mechanisms to improve detection accuracy.

Some detection algorithms do not use traditional machine learning or deep learning algorithms. Fang et al. [13] used an improved Word2Vec algorithm and incremental word embedding methods to capture the interactions and time series patterns between domains and endpoint hosts. By automatically learning the features of over 1.9 million domains, they developed a simple classifier to distinguish between malicious and benign domain names.

In summary, traditional machine learning heavily relies on feature engineering, while deep learning offers the advantage of automatic feature extraction. Building upon existing DGA models, our research demonstrates that integrating the ECA module enables efficient multi-scale feature fusion, effectively reduces model parameters, and enhances accuracy.

3. DGA Domain Detection Model

This section presents the domain detection framework, model, its layers and components, as well as their functions.

3.1. Overall Framework

To address the limitations of parallel multiscale convolution in terms of feature coupling and focus aggregation, this paper employs the Selective Kernel Network (SKNet) [14] for feature extraction, optimizing the traditional parallel convolution architecture. SKNet enhances inter-branch feature fusion by implementing an attention mechanism and adaptively adjusts the receptive field weights. However, the dimensional scaling approach used by SKNet may compromise feature integrity and increase training costs. To mitigate this, we introduce the ECA [15] mechanism, which allocates attention across receptive fields through limited cross-channel interactions. Additionally, the paper incorporates Deep Separable Dilated Convolution [16], a method that expands the receptive fields without increasing parameter counts, providing richer semantic information for the backbone network.

To address the issues of insufficient hierarchical feature extraction and feature redundancy resulting from the simple concatenation of outputs from each stage of the backbone network, this paper introduces the FPN [17] to achieve multi-scale feature fusion. By layering up-sampling and lateral connections between different levels, FPN effectively integrates features across scales to enhance detection accuracy. To further enhance fusion efficiency, this paper proposes an improved fusion strategy, EFAM, which reduces feature redundancy and improves aggregation by incorporating an efficient attention mechanism and refining the convolution strategy to better focus on spatial details.

Building on these enhancements, this study introduces a multi-scale feature fusion model utilizing Transformer and R-SKNet. The model consists of three layers: an input layer that converts domain name strings into vectors for word embedding; a feature extraction layer composed of the Transformer Encoder, R-SKNet, and EFAM; and an output layer that employs 1D convolution instead of traditional fully connected layers, outputting the final classification results via a Softmax function. The entire framework is shown in Figure 1.

3.2. Input Layer

The input layer preprocesses raw domain names and converts them into a vector form that machines can interpret. In this study, the length of all domain names is standardized to L . Domain names longer than L are truncated, while those shorter than L are padded with zeros on the right, ensuring that all domain name texts have a uniform length of L , as shown in Figure 2. Let the domain name string be S , where c_i is the i -th character of the domain name, as shown in (1).

$$S = c_1 c_2 c_3 \cdots c_L \quad (1)$$

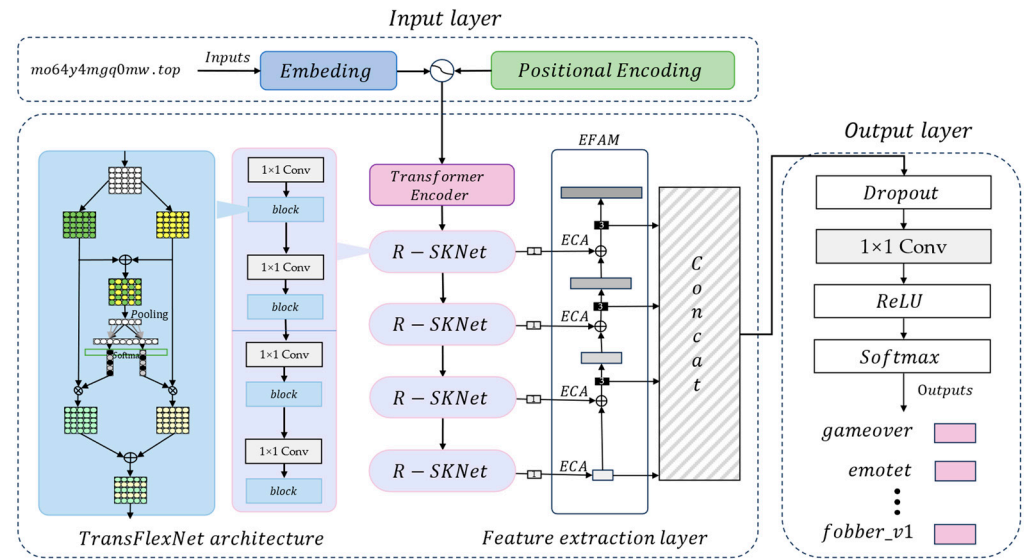


Figure 1. Overall framework.

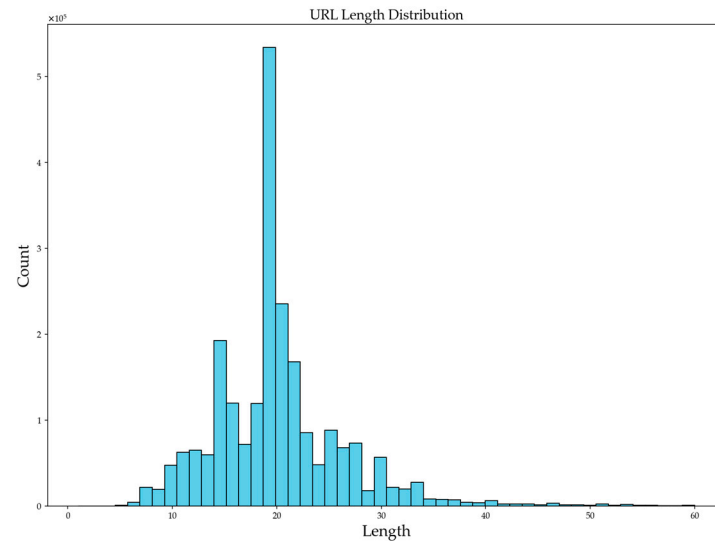


Figure 2. Sample domain length.

As shown in Figure 2, the samples with a length greater than 40 are relatively few. To ensure efficient use of the dataset and to improve data processing efficiency, this study sets the length threshold, L , to 40.

In this paper, a domain name is represented as a unique type of string. The domain name text is split into individual characters, each assigned an index to create an indexed dictionary, D . The original domain name samples are then processed with a tokenizer to generate a list of domain name characters, SD , as shown in (2).

$$D = [D(c_1)D(c_2) \dots D(c_L)] \quad (2)$$

where $D(c_i)$ is the index value of the domain character c_i in the dictionary D , and $i \in \{1, 2, \dots, L\}$. The embedding layer's dimension is set to d . According to the embedding layer dimension d and domain name length L , a domain name vector $R^{L \times d}$ is generated by random initialization. For the character vector $x_i \in R^{L \times d}$, it is multiplied by the weight matrix $A^{d \times L}$ to obtain Y_i , and they are concatenated to obtain a new domain name vector E . The weight matrix A is updated during downstream task training to obtain the optimal embedding representation, with d set to 64.

3.3. Feature Extraction Layer

The feature extraction layer integrates the global information encoding capabilities of the Transformer with the rapid adaptive kernel selection mechanism of R-SKNet to target different features effectively. It further incorporates the EFAM efficient fusion attention module to enhance multi-scale feature fusion. This comprehensive design optimizes the model's parameter count and reduces structural complexity while improving the efficiency and accuracy of feature extraction.

3.3.1. Transformer Encoder

The structure of the transformer encoder module is shown in Figure 3.

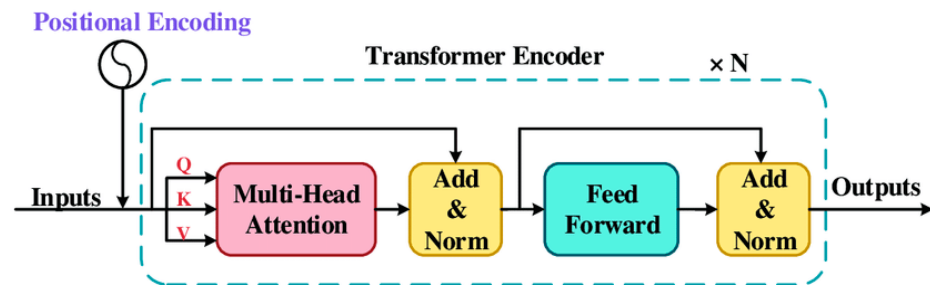


Figure 3. Transformer encoder module.

After processing through the embedding layer, domain names are converted into vector representations, E . These vectors are then passed through the Transformer's multi-head self-attention mechanism, which captures complex interactions between characters by applying multiple attention heads, each focusing on different subspaces of the input. Each head performs linear transformations on the input vector E to generate Q (query), K (key), and V (value). The computation process of the multi-head attention mechanism is expressed as shown in (3).

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^o \quad (3)$$

where each head_i is the output of a single attention head, and W^o is a weight matrix for linear transformation. The calculation process of each head_i is shown in (4).

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \quad (4)$$

where Q_i (query), K_i (key), and V_i (value) matrices are obtained by linear transformation of the original domain name vectors R . The dimension d_k of the key vector is used for normalization, adjusting the softmax distribution to stabilize gradient flow during training.

To ensure that the model adapts to the dynamic changes in input features, the output of the multi-head attention mechanism is normalized through residual connection, as shown in (5).

$$x'_i = \frac{x_i - \mu}{\sigma} \quad (5)$$

where x'_i represents the normalized output and μ and σ represent the mean and standard deviation across the dimensions. Normalization helps to stabilize the training dynamics. The normalized output is then fed into a feed-forward neural network (FFNN), which generates the output of the Transformer encoder, providing a refined feature set for subsequent analysis.

3.3.2. Rapid Selective Kernel Network

In traditional DGA domain detection models, feature gains from different receptive fields are typically achieved through a multi-branch network structure with fixed receptive

field sizes. This design leads to an increase in the number of model parameters and higher computational complexity. SKNet introduces additional parameters through its multi-path kernel selection mechanism, which complicates the model, and increases computational demand by applying convolutional kernels of various sizes across multiple branches to enhance feature processing.

To address this, we propose R-SKNet, which integrates an ECA module into SKNet. R-SKNet achieves multi-receptive field feature extraction by adaptively selecting convolution kernel sizes within a single network, simplifying design and optimizing complexity. The ECA module further reduces parameters while enhancing performance, demonstrating improved detection accuracy with a lightweight model. The structure of each R-SK convolution is shown in Figure 4.

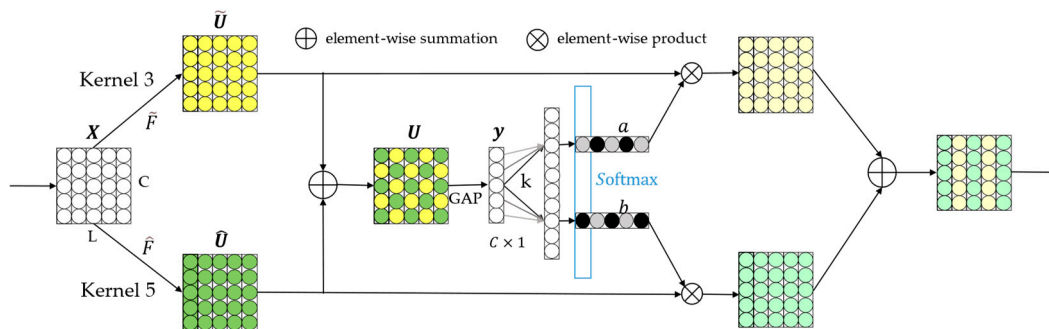


Figure 4. R-SK convolution structure.

In Figure 4, the input domain name matrix $X \in R^{C' \times L'}$ undergoes two different transformation operations, denoted as: $\tilde{F} : X \rightarrow \tilde{U} \in R^{C \times L}$ and $\hat{F} : X \rightarrow \hat{U} \in R^{C \times L}$. These operations use one-dimensional convolutional kernels of different sizes (3 and 5) for feature extraction. To enhance model efficiency, traditional convolutions are replaced by depthwise separable dilated convolutions. Batch normalization and the ReLU activation function are applied after each convolution to improve non-linearity and generalization capability. In particular, the 1D convolution with a kernel size of 5 is replaced by a dilated convolution with a kernel size of 3 and a dilation rate of 2.

The depthwise separable convolution technique decomposes standard convolution into depthwise convolution and pointwise convolution, processing each dimension of the input matrix separately. This approach allows each convolution kernel to independently correspond to and process a single dimension, focusing on learning spatial features, while pointwise convolution is responsible for learning features between channels. This decomposition significantly reduces the number of model parameters while preserving learning efficiency.

To gather information from all branches and assign weights to the grouped convolution results, the processing results of each branch are first obtained, and then the features are fused, as shown in (6).

$$U = \tilde{U} + \hat{U} \quad (6)$$

After obtaining the fused features from different branches, global average pooling is used to generate channel-level statistics. In the DGA domain name detection task, the use of global average pooling (GAP) can achieve better feature aggregation effects. After GAP, the channel-level statistical information $y \in R^C$ is generated, as shown in (7).

$$y = \frac{1}{CL} \sum_{i=1, j=1}^{C, L} U_{ij} \quad (7)$$

To introduce a channel attention mechanism that adaptively adjusts channel weights, two non-linear fully connected (FC) layers are typically employed, followed by a Sigmoid function to generate the weights. Although this approach enhances the model's ability to

capture non-linear cross-channel interactions, it significantly increases model parameters and disrupts the correspondence between channels and weights. To address this, we propose using the ECA module, which replaces traditional FC layers to facilitate local cross-channel interactions without dimensionality reduction. The ECA module effectively facilitates appropriate inter-channel interactions without sacrificing the correspondence between channels, significantly enhancing model performance. The ECA module is shown in Figure 5. The ECA module uses a band matrix W_k to learn channel attention, as shown in Figure 6.

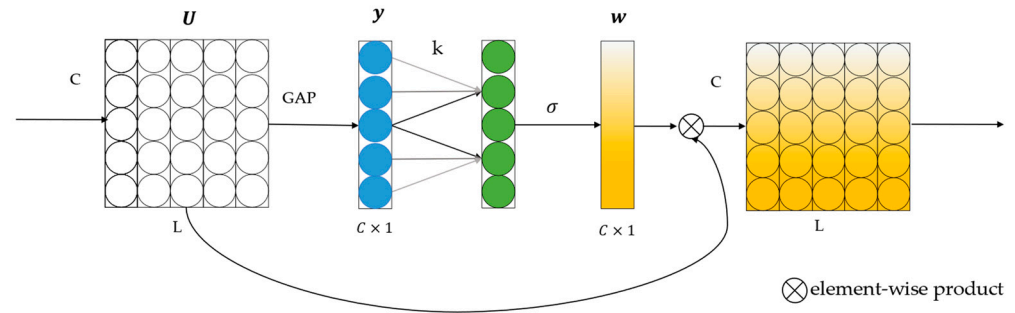


Figure 5. ECA module.

$$\begin{bmatrix} w^{1,1} & \dots & w^{1,k} & 0 & 0 & \dots & \dots & 0 \\ 0 & w^{2,2} & \dots & w^{2,k+1} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & w^{C,C-k+1} & \dots & w^{C,C} \end{bmatrix}$$

Figure 6. Band matrix.

The parameters involved in the band matrix W_k are $K \times C$. Through (8), the weight of y_i considers only the relationship between y_i and its k neighbors. Additionally, all channels share the same learned parameters, enabling local cross-channel interaction. This can be implemented using a 1D convolution with a kernel size of k , as shown in (9).

$$Uw_i = \sigma \left(\sum_{j=1}^k w^j y_i^j \right), y_i^j \in \Omega_i^j \quad (8)$$

$$w = \sigma(C1D_k(y)) \quad (9)$$

where Ω_i^k denotes the set of k neighboring channels of y_i , σ represents the Sigmoid function, and $C1D$ denotes the one-dimensional convolution. The value of k is dynamically determined by (10).

$$k = \left\lceil \frac{\log_2(2^{(C)})}{\gamma} + \frac{b}{\gamma} \right\rceil_{odd} \quad (10)$$

where $|t|_{odd}$ denotes the nearest odd number to t , and C represents the channel dimension. In all experiments, γ and b are set to 2 and 1, respectively. The ECA module, which involves only b parameters, achieves better performance than FC layers, which require more complex operations. Based on the information of w , softmax is performed on the channel dimension to obtain a_c and b_c , as shown in (11).

$$a_c = \frac{e^{\tilde{U}_c w}}{e^{\tilde{U}_c w} + e^{\hat{U}_c w}}, b_c = \frac{e^{\hat{U}_c w}}{e^{\tilde{U}_c w} + e^{\hat{U}_c w}} \quad (11)$$

where a and b are the soft attention vectors of $\tilde{U}$ and $\hat{U}$ respectively. $\tilde{U}_c \in R^{C \times 1}$, with a_c being the c -th element of a . Similarly, $\hat{U}_c$ and b_c . The final batch feature map V is derived from the attention weights of each kernel from the branches $\tilde{U}$ and $\hat{U}$, as shown in (12).

$$V_c = \frac{a_c \cdot \tilde{U}_c + b_c \cdot \hat{U}_c}{a_c + b_c}, \quad a_c + b_c = 1 \quad (12)$$

where $V = [V_1, V_2, V_3, \dots, V_C]$, and $V_c \in R^{C \times 1}$. After obtaining the feature matrix V , a residual connection is made to complete the R-SK convolution operation.

3.3.3. Efficient Fusion Attention Module

The EFAM is a key component of the TransFlexNet architecture. By incorporating the ECA module before both lateral and vertical connections in the FPN and performing convolution after feature fusion, EFAM enhances the efficiency of feature fusion and strengthens feature representation. In the R-SKNet backbone, the output from the last residual block of each R-SK unit, with varying numbers of convolution kernels, is selected as input and passed to the EFAM for feature fusion. The structure of the EFAM is shown in Figure 7.

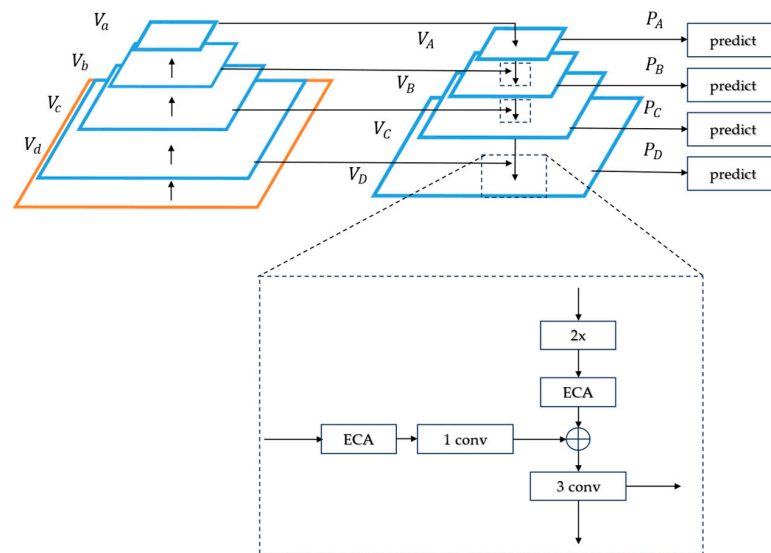


Figure 7. EFAM structure.

Starting with the input feature matrices $\{V_a, V_b, V_c, V_d\}$ from the R-SKNet backbone, each matrix first passes through the ECA module to compute channel-level attention weights. These weights are applied to the original matrices, enhancing the network's response to key features and improving representation capabilities. A 1D convolution with a kernel size of 1 is then used to adjust the channel number of each weighted matrix to 128, resulting in $\{V_a, V_b, V_c, V_d\}$.

To build a multi-scale feature representation hierarchy and promote bottom-up information flow, pooling operations are applied to these adjusted matrices. Specifically, max pooling with a kernel size of 3 and stride of 2 is applied to V_b , average pooling with a kernel size of 3 and stride of 4 to V_c , and average pooling with a kernel size of 3 and stride of 8 to V_d . These operations create feature maps at different scales, with strides set to $\{5, 10, 20, 40\}$. By upsampling higher-level features, the EFAM generates feature matrices with coarser spatial resolution but richer semantic information, establishing a top-down feedback path.

Lateral connections are then used to fuse top-down and bottom-up feature paths. After each fusion, the ECA module readjusts channel attention, followed by a 1D convolution

with a kernel size of 3 to extract features, resulting in $\{V_a, V_b, V_c, V_d\}$, which match the original input dimensions. These matrices $\{V_a, V_b, V_c, V_d\}$ serve as the output of the feature extraction layer and the input for the next layer in the pyramid, further strengthening features in the top-down path.

3.3.4. Output Layer

First, the outputs of the feature extraction layer $\{P_A, P_B, P_C, P_D\}$ are fused into a comprehensive feature vector P in the output layer, achieving feature information integration, as shown in (13).

$$P = [P_A, P_B, P_C, P_D] \quad (13)$$

Next, the Dropout layer is used to probabilistically discard elements from the feature vector P to improve the generalization ability of the model and to reduce the risk of overfitting. Subsequently, a convolutional layer with a kernel size of 1 is used to replace the traditional fully connected layer, followed by a ReLU layer and a global average pooling layer, which significantly reduces the number of model parameters, adds non-linear activation, and enhances the network's expressive ability. Finally, the softmax function is used to obtain the probability y of each class, achieving the final classification.

4. Experiments and Result Analysis

4.1. Experimental Environment and Datasets

The experimental setup includes an Intel Core i7-11800H processor, NVIDIA RTX3070 GPU, and 32 GB RAM, operating on Windows 11 with Python 3.11 and PyTorch 2.2.1 for neural network modeling.

We have successfully deployed the detection model on low-performance devices in real-world scenarios, demonstrating its practical applicability. To further validate the model, we designed and implemented a prototype system for model deployment and detection. The system was put on the URL: https://github.com/make666999/jishe_DGA (accessed on 20 November 2024). However, due to the limited occurrence of large-scale DGA domain malicious attack incidents in real network environments, this study utilizes multiple publicly available datasets to construct and validate the domain name classification model.

To improve the breadth and accuracy of the research, different types of datasets are used in this experiment. The first dataset, sourced from Alexa [18] and 360NetLab [19], consists of the top 55,000 benign domains from Alexa and malicious domains from 360NetLab's DGA dataset. The second dataset is sourced from Alexa and DGArchive [20], comprising 57,500 benign domains and 71,841 malicious domains, respectively.

Table 1 lists the three initial samples from 360NetLab and DGArchive. Table 2 lists the three initial samples from Alexa.

Table 1. 360NetLab and DGArchive initial samples.

Dataset	DGA Family	Domian	Start Valid Time	End Valid Time
360NetLab	locky	labywvrw.pw	2020-10-26 00:00:00	2020-10-27 23:59:59
	nymaim	jkybwgxpfr.com	2020-10-27 00:00:00	2020-10-27 23:59:59
	qadars	sqgaoag606cm.top	2020-10-27 00:00:00	2020-11-02 23:59:59
DGArchive	bazarloader_dga_bc7bb913	idfuuhom.bazar	2021/1/1 0:00	2021/1/31 23:59
	bazarloader_dga_bc7bb913	biyxtuom.bazar	2021/12/1 0:00	2021/12/31 23:59
	bazarloader_dga_bc7bb913	etywydom.bazar	2021/10/1 0:00	2021/10/31 23:59

Table 2. Alexa initial samples.

Label	Url
BENIGN	facebook.com
BENIGN	youtube.com
BENIGN	www.google.com

Considering that the sample size of some malicious domain families is insufficient to support model training, families with fewer than 200 samples were excluded to support model training, selecting 33 families from 360NetLab and 42 from DGArchive. To mitigate sample imbalances, domains were randomly sampled from these larger families, and all domain names were standardized to 40 characters through padding or truncation. The processed datasets from 360NetLab and DGArchive were combined with the Alexa dataset, resulting in two comprehensive datasets: Alexa and 360NetLab with 107,541 domain instances and Alexa and DGArchive with 129,341 domain instances. Table 3 lists the results of several datasets after preprocessing.

Table 3. Example after preprocessing.

Label	Url
0	facebook.com
14	labywvrw.pw
18	jkybwgxpfr.com

For binary classification, benign domains are labeled as 0 and malicious domains as 1. In multiclassification, the Alexa and 360NetLab dataset contains 33 DGA families and benign domains, while the Alexa and DGArchive dataset includes 42 DGA families and benign domains, with labels assigned sequentially from 0. Both datasets were randomly divided into training and test sets, with test sets accounting for 20% of the total data. Dataset details and experimental hyperparameters are listed in Tables 4 and 5.

Table 4. Detailed information of dataset.

Category	Alexa and 360NetLab	Alexa and DGArchive
Total Number of Domains	107,541	129,341
Benign Domains	55,000	57,500
DGA Domains	52,541	71,841
Domain Length	40	40
Total Number of Categories	34	42

Table 5. Model hyperparameters.

Hyperparameters	Description
Epochs	100
Batch Size	256
Embedding Dimension	64
Learning Rate	0.001
Step Size for Learning Rate Scheduler	20
Gamma for Learning Rate Scheduler	0.5
Test set proportion	20%
Optimizer	Adam
Loss function	CrossEntropyLoss

4.2. Evaluation Metrics

The evaluation metrics for the experimental results include accuracy, precision, recall, F1 score, and detection time. The calculation formulas are shown in (14) to (17).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (17)$$

where TP is the number of true positives, FN is the number of false negatives, and FP is the number of false positives.

4.3. Binary Classification Experimental Results and Analysis

To verify the performance of the proposed method in DGA domain name detection, binary classification experiments and model parameter comparisons are conducted on the experimental sample set using the proposed method, MHA [21], SW-DRN [22], L-PCAL [23], and the multi-feature fusion model (referred to as TMF). The evaluation metrics included accuracy, recall, precision, F1 score, and parameter count. The experimental results for the Alexa and 360NetLab are shown in Table 6 and Figure 8, and the Alexa and DGArchive's results are shown in Table 7.

Table 6. Binary classification results and model parameters for Alexa and 360NetLab.

Method	Accuracy	Recall	F1 Score	# P
MHA	0.9269	0.9448	0.9260	406,306
SW-DRN	0.9679	0.9717	0.9669	1,963,524
L-PCAL	0.9684	0.9754	0.9676	274,326
TMF	0.9766	0.9754	0.9756	2,771,384
Proposed Method	0.9783	0.9834	0.9777	1,783,544

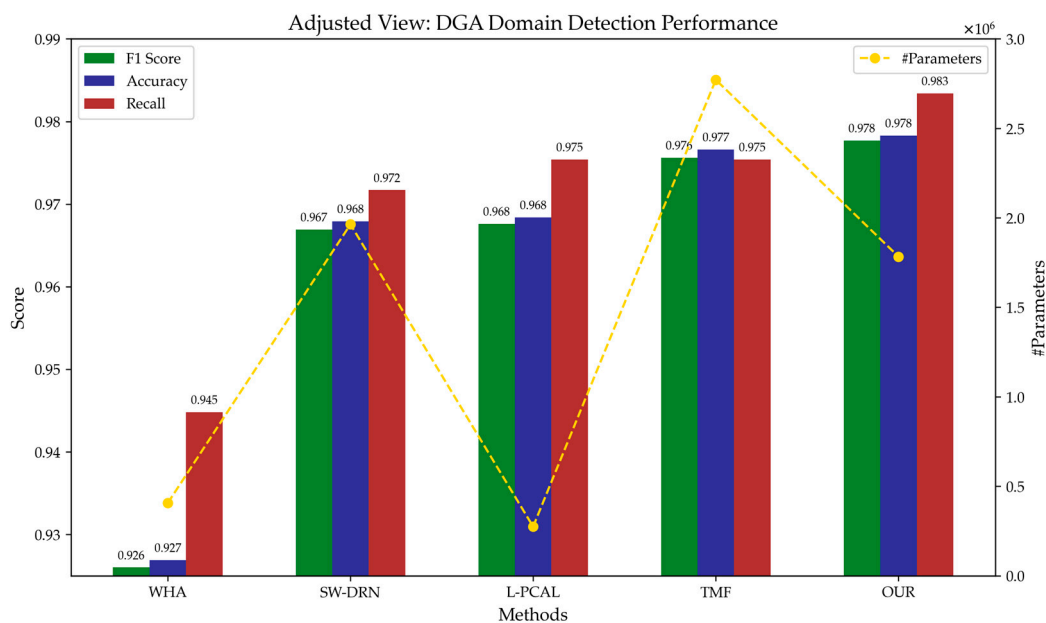


Figure 8. Binary Classification Results and Model Parameters Comparisons.

Table 7. Binary classification results for Alexa and DGArchive.

Method	Accuracy	Recall	Precision	F1 Score
MHA	0.9646	0.9646	0.9634	0.9640
SW-DRN	0.9799	0.9790	0.9802	0.9796
L-PCAL	0.9791	0.9766	0.9808	0.9787
TMF	0.9840	0.9822	0.9852	0.9837
Proposed Method	0.9852	0.9864	0.9852	0.9850

The proposed method integrates the Transformer encoder and SKNet with the ECA module to form the improved R-SKNet. The EFAM combines the ECA module with the FPN structure for multi-scale feature fusion, efficiently extracting feature information. As

presented in Table 6, the Alexa and 360NetLab datasets increase the accuracy, recall and F1 score by 1.87%, 1.83%, and 1.66% on average, with peak increases of 5.14%, 3.86%, and 5.17%. Furthermore, the model reduces the parameter count by 35.63% compared to the next-best model. From Table 7, it can be seen that the maximum increase in the Alexa and DGArchive dataset in accuracy, recall, precision, and F1 score reaches 2.12%, 2.06%, 2.18%, and 2.06%, respectively. The method in this paper is more effective and has fewer parameters compared to mainstream models.

4.4. Multi-Classification Experimental Results and Analysis

To validate the effectiveness of the proposed method for DGA domain family classification, we have conducted multi-class domain detection experiments using our method, MHA, SW-DRN, L-PCAL, and TMF on the experimental sample set. The experimental results are shown in Tables 8 and 9.

Table 8. Multi-classification results comparison for Alexa and 360NetLab.

Method	Accuracy	Recall	Precision	F1 Score
MHA	0.8937	0.8937	0.8908	0.8883
SW-DRN	0.9225	0.9225	0.9196	0.9206
L-PCAL	0.9315	0.9315	0.9330	0.9310
TMF	0.9335	0.9335	0.9334	0.9332
Proposed Method	0.9391	0.9391	0.9377	0.9378

Table 9. Multi-classification results comparison for Alexa and DGArchive.

Method	Accuracy	Recall	Precision	F1 Score
MHA	0.9045	0.9045	0.9022	0.8993
SW-DRN	0.9113	0.9113	0.9111	0.9102
L-PCAL	0.9234	0.9234	0.9228	0.9202
TMF	0.9214	0.9214	0.9207	0.9205
Proposed Method	0.9251	0.9251	0.9233	0.9210

From the results in Tables 8 and 9, it is clear that the accuracy, recall, precision, and F1 score for the proposed detection method on the Alexa and 360NetLab dataset are 93.91%, 93.91%, 93.77%, and 93.78%, respectively. On the Alexa and DGArchive dataset, the respective metrics are 92.51%, 92.51%, 92.33%, and 92.10%. The multiclassification detection performance of this method exceeds that of other approaches, with improvements noted across all evaluation metrics.

From the above comparison results, it can be seen that the proposed method performs better in detecting and classifying DGA domain families with fewer samples compared to the other four mainstream methods.

The MHA method uses an LSTM-encoded Transformer detection method, obtaining the dependency between individual characters and the entire domain name, but lacks the learning of multi-scale features and cannot extract deep semantic information, leading to poor detection performance.

The SW-DRN method uses multi-scale convolution to obtain domain name features but does not capture the dependency between characters and uses one-hot encoding, resulting in huge memory consumption.

The L-PCAL method uses multi-scale parallel convolution and LSTM with an attention mechanism to extract short-distance dependencies, but the extraction effect of long-distance dependencies is poor.

The TMF method also uses parallel convolution, and the multi-scale relationships between parallel branch convolutions are isolated and cannot be effectively coupled. The multi-branch feature fusion is simply summed, causing branch feature generalization and leading to reduced accuracy.

The experimental results indicate that, compared to other methods, the method presented in this paper performs best and requires fewer parameters. The main reasons can be divided into three aspects:

- The Transformer encoder part provides a global perspective on domain name character dependencies;
- The R-SKNet efficiently couples multi-scale feature branches, allowing the model to perform information interaction between branches and enhancing the ability to extract deep features;
- The EFAM effectively fuses the outputs of each stage of the backbone network, further enhancing the model's ability to extract domain name feature information.

4.5. Ablation Study

To evaluate the effectiveness of each module in the proposed TransFlexNet architecture, we have conducted an ablation study comparing various configurations of existing DGA domain detection models with and without the use of Transformer (TR), SKNet or R-SKNet, and FPN, or EFAM. The binary and multiclassification results for the Alexa and 360NetLab dataset are displayed in Tables 10 and 11, respectively.

Table 10. Binary classification performance for different model configurations: Alexa and 360NetLab.

Model Configuration	Accuracy	Recall	Precision	F1 Score
TR	0.9221	0.9476	0.8970	0.9216
SKNet	0.9526	0.9535	0.9487	0.9511
SKNet + FPN	0.9501	0.9580	0.9400	0.9489
SKNet + EFAM	0.9515	0.9607	0.9403	0.9504
R-SKNet	0.9498	0.9475	0.9486	0.9480
R-SKNet + FPN	0.9516	0.9610	0.9403	0.9505
R-SKNet + EFAM	0.9500	0.9519	0.9451	0.9485
TR + SKNet	0.9746	0.9773	0.9703	0.9738
TR + SKNet + FPN	0.9753	0.9765	0.9724	0.9745
TR + SKNet + EFAM	0.9761	0.9813	0.9697	0.9755
TR + R-SKNet	0.9761	0.9784	0.9724	0.9754
TR + R-SKNet + FPN	0.9760	0.9778	0.9727	0.9753
Proposed Method	0.9783	0.9834	0.9812	0.9777

Table 11. Multi-classification performance for different model configurations: Alexa and 360NetLab.

Model Configuration	Accuracy	Recall	Precision	F1 Score
TR	0.8838	0.8838	0.8806	0.8787
SKNet	0.9007	0.9007	0.8891	0.8943
SKNet + FPN	0.8989	0.8989	0.8922	0.8944
SKNet + EFAM	0.9031	0.9031	0.8990	0.9004
R-SKNet	0.9007	0.9007	0.8957	0.8973
R-SKNet + FPN	0.9011	0.9011	0.8959	0.8978
R-SKNet + EFAM	0.9066	0.9066	0.9016	0.9034
TR + SKNet	0.9300	0.9300	0.9214	0.9254
TR + SKNet + FPN	0.9365	0.9365	0.9354	0.9356
TR + SKNet + EFAM	0.9352	0.9352	0.9348	0.9345
TR + R-SKNet	0.9352	0.9352	0.9333	0.9339
TR + R-SKNet + FPN	0.9366	0.9366	0.9354	0.9357
Proposed Method	0.9391	0.9391	0.9377	0.9378

As indicated by Table 10, through detailed experimental comparisons, our method achieves the best results in all metrics for binary classification. The improved R-SKNet outperforms the original SKNet, EFAM surpasses FPN, and the use of these improved module combinations yields even better results.

As shown in Table 11, the method presented in this paper achieves the best results in accuracy, recall, precision, and F1 score for multiclassification tasks. Compared to the original SKNet and FPN structures, R-SKNet and EFAM exhibit superior performance under various module combinations.

5. Conclusions

To address the difficulties in extracting a small number of domain features in existing domain detection models, which had certain accuracy but a large amount of redundancy, this paper proposed a multi-scale feature fusion model based on Transformer and R-SKNet. In the input layer, the domain text was split into individual characters and indexed to achieve optimal word embedding and to reduce the difficulty of feature extraction. The feature extraction layer applied the TransFlexNet architecture, which used multi-scale feature fusion to flexibly handle long-distance dependencies and the dynamic changing local feature. The design resulted in complexity reduction. The output layer underwent multiple processing layers to optimize the number of parameters. Compared with existing models, the multi-feature model proposed in this paper achieved better feature extraction results, simplified structure, and improved accuracy. Experimental results showed that in both binary and multi-class classification experiments, the evaluation metrics of this model are mostly improved compared to existing methods, providing a better solution for dealing with botnets and malicious domains.

In future work, we will focus on optimizing the model architecture to enhance the efficiency of individual components and reduce overall complexity. Techniques such as quantization and model pruning will be employed to mitigate the computational overhead introduced by the Transformer architecture, enabling efficient operation on embedded devices and expanding the application scenarios for DGA detection. To further enhance model interpretability, we plan to leverage methods such as Local Surrogate models to provide clearer explanations and insights into the model's decision-making process.

Author Contributions: Conceptualization, J.T. and S.Z.; methodology, J.T., Y.G., S.Z. and Y.C.; validation, J.T. and Y.G.; formal analysis, J.T., Y.G. and H.W.; investigation, J.T., Y.G. and H.W.; data curation, J.T. and Y.G.; writing—original draft preparation, J.T. and Y.G.; writing—review and editing, J.T., Y.G., S.Z., H.W. and Y.C.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University Synergy Innovation Program of Anhui Province (grant number GXXT-2023-052), and the Chuzhou Science and Technology Plan Project (grant number 2023CI001).

Data Availability Statement: The data presented in this study includes the Alexa and 360NetLab dataset and the Alexa and DGArchive dataset, they are all publicly available on the Internet.

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

References

1. Yan, F.; Liu, J.; Gu, L.; Chen, Z. A Semi-Supervised Learning Scheme to Detect Unknown DGA Domain Names Based on Graph Analysis. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020; pp. 1578–1583.
2. Anand, P.M.; Kumar, T.G.; Charan, P.S. An ensemble approach for algorithmically generated domain name detection using statistical and lexical analysis. *Procedia Comput. Sci.* **2020**, *171*, 1129–1136. [\[CrossRef\]](#)
3. Satoh, A.; Fukuda, Y.; Kitagata, G.; Nakamura, Y. A word-level analytical approach for identifying malicious domain names caused by dictionary-based DGA malware. *Electronics* **2021**, *10*, 1039. [\[CrossRef\]](#)
4. Vranken, H.; Alizadeh, H. Detection of DGA-generated domain names with TF-IDF. *Electronics* **2022**, *11*, 414. [\[CrossRef\]](#)
5. Yang, C.; Lu, T.; Yan, S.; Zhang, J.; Yu, X. N-trans: Parallel detection algorithm for DGA domain names. *Future Internet* **2022**, *14*, 209. [\[CrossRef\]](#)
6. Namgung, J.; Son, S.; Moon, Y.-S. Efficient deep learning models for DGA domain detection. *Secur. Commun. Netw.* **2021**, *2021*, 8887881. [\[CrossRef\]](#)
7. Shahzad, H.; Sattar, A.R.; Skandaraniyam, J. DGA domain detection using deep learning. In Proceedings of the 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP), Zhuhai, China, 8–10 January 2021; pp. 139–143.
8. Liang, J.; Chen, S.; Wei, Z.; Zhao, S.; Zhao, W. HAGDetector: Heterogeneous DGA domain name detection model. *Comput. Secur.* **2022**, *120*, 102803. [\[CrossRef\]](#)
9. Qi, G.; Mao, J. An Improved CNN-LSTM Algorithm for Detection of DGA Domain Name. In Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering, Xiamen China, 20–22 October 2023; pp. 1293–1298.

10. Jiang, Y.; Jia, M.; Zhang, B.; Deng, L. Malicious domain name detection model based on CNN-GRU-attention. In Proceedings of the 2021 33rd Chinese control and decision conference (CCDC), Kunming, China, 22–24 May 2021; pp. 1602–1607.
11. Ren, F.; Jiang, Z.; Wang, X.; Liu, J. A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network. *Cybersecurity* **2020**, *3*, 4. [[CrossRef](#)]
12. Pan, R.; Chen, J.; Ma, H.; Bai, X. Using extended character feature in Bi-LSTM for DGA domain name detection. In Proceedings of the 2022 IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS), Zhuhai, China, 26–28 June 2022; pp. 115–118.
13. Fang, X.; Sun, X.; Yang, J.; Liu, X. Domain-embeddings based DGA detection with incremental training method. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6.
14. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
15. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
16. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
17. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
18. Liang, Y.; Deng, J.; Cui, B. Bidirectional LSTM: An innovative approach for phishing URL Identification. In *Innovative Mobile and Internet Services in Ubiquitous Computing*; Springer: Cham, Switzerland, 2020; Volume 994, pp. 326–337.
19. Xu, Y.; Yan, X.; Wu, Y.; Hu, Y.; Liang, W.; Zhang, J. Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2946–2957. [[CrossRef](#)]
20. Plohmann, D.; Yakdan, K.; Klatt, M.; Bader, J.; Gerhards-Padilla, E. A comprehensive measurement study of domain generating malware. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 263–278.
21. Zhang, X.; Cheng, H.; Fang, Y. A DGA domain name detection method based on Transformer. *Comput. Eng. Sci.* **2020**, *42*, 411–417.
22. Zhang, B.; Liao, R.J. Malicious domain name detection model based on CNN and LSTM. *J. Electron. Inf. Technol.* **2021**, *43*, 2944–2951.
23. Liu, X.Y.; Liu, J.M.; Liu, C.; Zhang, Y.H. Novel Botnet DGA domain detection method based on character level sliding window and deep residual network. *Acta Electron. Sin.* **2022**, *50*, 250–256.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.