Hybrid Transformer Framework for Domain Generated Algorithms Detection: A Fusion of Textual and Numeric Features

Suhad Malayshi
Department of Natural, Engineering and Technology Sciences, Arab American University, Palestine
s.malayshi@student.aaup.edu

Ahmad Hasasneh*
Department of Artificial Intelligence
Arab American University, Palestine
*Corresponding Author: ahmad.hasasneh@aaup.edu

Abstract: *A Domain Generation Algorithm (DGA) is a program that generates a large number of spam domain names, cyber criminals use domain generation algorithms to initiate a malware attack, making it important for cybersecurity teams to identify the DGA domains and strengthen the organization's defense against threats. This paper designs a state-of-the-art artificial Intelligence model for DGI domain detection, which is developed using an innovative fusion of semantic and statistical modalities. The textual features are processed using the BERT text transformer, while the numerical features are processed using a simple Multi-Layer Perceptron Network. The model is applied to a dataset of 160,000 Alexa domains labeled as DGA or Legit. The evaluation of the approach is done based on different measures such as accuracy, precision, recall F1-score and confusion matrix which showed a promising result for accurately detecting the DGA domains. Our model achieved an accuracy of 0.9932. The result demonstrated the effectiveness of the model in classifying the domain names and the ability to generalize the model to other unseen domains and many other real-world scenarios.*

Keywords: *Domain generated algorithms, cybersecurity, hybrid deep learning, text transformers, multimodality, feature importance*.

Received April 9, 2025; accepted September 10, 2025
https://doi.org/10.34028/iajit/23/1/9

1. Introduction

Cybersecurity plays a pivotal role in our modern world, as the amount of sensitive and confidential data is increasing. It protects internet connected devices, software and data from cyber-attacks. All types of organizations such as corporations, governments, banks and enterprises enforce the cybersecurity to prevent phishing, ransomware, theft, data breaches and ideally avoid financial losses [9]. One of the most common threats is the use of Domain Generated Algorithms (DGAs). DGAs are malicious software used to generate random and erratic domain names that allow the malware to receive instructions, upload data or download a malicious software, thus creating security risks [38]. DGAs can be classified into several families which is beyond of our scope here [38]. DGA use highly advanced approaches to compromise the end user which is considered as "stealth mechanism", the working principal is visualized in Figure 1 starting with infection where the attacker initiates the victim to visit the DGA domain, then the malware initiate a seed value which is a random number generated from time, string, number and exchange rate. This seed will be used to generate domain names which pointed to the IP of command-and-control servers of the attackers, after that the installed malware will spread in the network and at the end the data or information will be stolen [21]. Table 1 shows a sample of DGA domain and Legit domains as clearly observed for us that DGA domains are not human readable and much longer and contains digit.

Table 1. Samples of DGA and legit domains.

Legit domain	DGA domains
Google.com	xxmamopyipbfpk.ru
Mit.edu	zfd5szpi18i85wj9uy13l69rg.net
grafamania.net	jpqftymiuver.ru

The detection of DGA domains can be a challenging task, because of the tedious amount of new DGA domains and their high randomness, which makes traditional detection algorithms less efficient in detecting DGA domains. Machine learning and deep learning technologies have been employed in the field of cybersecurity, specifically, in DGA detection [38]. These cutting-edge technologies enhance the efficiency and accuracy of the domain detection.

DGA detection using deep learning has gained significant attention from security researchers. Most studies, including those in [27, 29], primarily focus on leveraging Long Short-Term Memory (LSTM) networks. These approaches typically involve converting domain names into character-level encodings, identifying sequential correlations through the LSTM layer, and subsequently passing the results to

alogistic regression layer for classification [8].

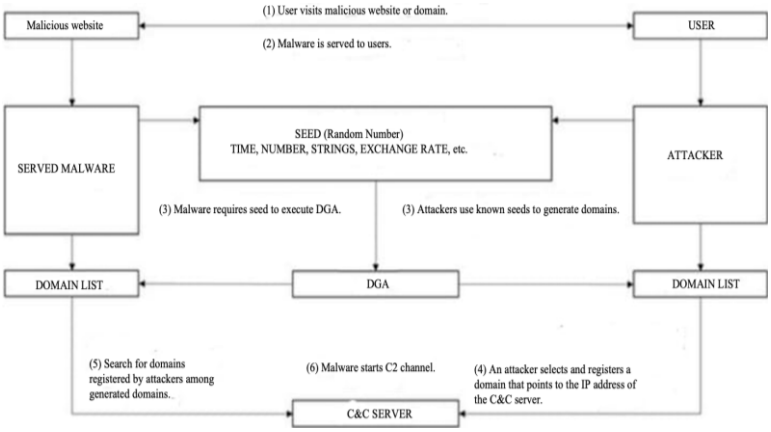


Figure 1. DGA domains principal of working [11].

Current research on DGA detection predominantly focuses on deep learning approaches that rely solely on domain names, without incorporating feature extraction or additional information. This paper emphasizes the potential of leveraging text transformers in combination with numerical feature fusion to enhance both the efficiency and accuracy of DGA detection. The primary contributions of this study include the detection of DGA domains through the fusion of semantic and numerical features, introducing a multidimensional approach. .In addition, this research work highlights the significance of numerical features in comparison to textual features. Unlike other studies, which rarely investigate feature embeddings, our approach sets a new benchmark and makes a significant contribution to the field by surpassing the traditional methods through the use of fully connected layers for joint learning of feature vectors.

The paper is structured as follows: section 2 reviews recent studies of DGA detection and classification, section 3 describes the proposed model and methodology, section 4 presents the obtained results followed by a detailed discussion, and finally section 5 concludes the paper with recommendations for future work.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

1. Literature Review

The detection of DGA domains has been an area of significant research for many years, particularly in the realm of traditional machine learning. Recently, there has been a shift towards exploring more sophisticated and effective deep learning models for this purpose, leading to significant advances in the field. This section provides a review of the literature on both machine learning and deep learning approaches to DGA domain detection, with a stronger emphasis on recent developments in deep learning models.

1.1. Machine Learning Detection Methods

In machine learning researches, a robust feature engineering has been worked on, in this study [20] trained and evaluated 14 machine learning and two deep learning comprehensive models on different datasets after applying robust feature engineering, some of the features used: length, entropy, digit ratio, length of vowels, length of prefix and mean frequency index. The highest F1-score achieved by MLP with 0.9602 followed by KNN with 0.9595 followed by XGB with 0.9590 and RF model with 0.9587 which represents a very good result for ML [20]. A botnet detection model based on machine learning model and text mining is used to analyze DGA domain names by taking advantage of n-gram features and PCA feature reduction technique [13]. They tested the proposed system using different models like random forest, logistic regression, SVM, and decision tree algorithm which resulted with 0.99, 0.93, 0.96 and 0.98 respectively [13].

A model of heterogeneous model named HAGDetector in employed in order to get rid of the sensitivity of the domain length over three stages, first calculate the length of the domain [16], then use different feature extraction methods for each length of the domain, then they used three classification models i.e., the extra-short DGA, moderate-length DGA and extra-short DGA domain names. Their proposed model is tested on DGArchive and Netlab360, which a chevied an accuracy of 0.9163 for short domain names, and 0.9444 for moderate length domains and 0.9875 for long domain length [16].

1.2. Deep Learning Detection Methods

The use of deep learning model to classify algorithmically generated domain is utilized in many

studies by employing Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LTSM), a review of top researches will be reviewed in this section. This study [6] utilizes LTSM and RNN. Their model used the domain names as input after which it's transformed into vectors, at the end fully connected layers and softmax function will classify the DGA and benign domains. It achieved an accuracy of 0.99 using AmritaDGA dataset [6]. Another study [18] also proposed a deep learning methods such as CNN and LSTM integrated with FastText for text embedding and extraction, the model was tested on Netlab360 and University of Murcia Domain Generation Algorithm Dataset (UMUDGA) and achieved an accuracy of 0.9770 and 0.9742 [13].The goals of this study [17] is to use only the contextual information features such as domain names using RNN based classifier. the experiment is done using Alexa top I million domains and cisco umbrella popularity list, achieving an accuracy of 0.87 trained over 15 epochs with 3 layers and 400 cells [17]. The use of LTSM also proposed efficient DGA detection method based on bidirectional LTSM [37] which improved the detection performance compared to CNN. they measure their experiment by using F1-score of 0.9618 and 0.9666 [37].

A multi head attention convolutional neural network method classifier in built, the extraction of features from domain names is done by employing shallow CNN, the model is tested on 360 DGA feeds resulting in a precision of 0.9868 [26]. Another study [30] which used different approach and developed a system called IDGADS using supervised deep learning methods, the system is used to learn from computable features from DNS queries without any external source of information. It achieved an accuracy of 0.99 on DGArchive [30]. This study [10] used Term Frequency-Inverse Document Frequency (TF-IDF) to measure to measure the importance of n-gram in domain names to compare between the deep MultiLayer Perceptron (MLP) model results, the results showed a performance of LSTM and MLP of 0.994 and 0.995 accuracies [10].

A hybrid neural network is developed by using the CNN and LSTM in parallel network, which was later trained on a big dataset of known dictionary-based domain generation [24]. The features extracted from CNN and LTSM were fed into ANN hidden layer and then flattened to produce output. The model achieved an accuracy of 0.9656 [24].

The use of multiple features like domain names, whois API, and n-gram is utilized in this study [32] the features were fed into input layer then it fed into BiLSTM network in order to generate hidden vector, after that an attention mechanism is used to assign degrees for the hidden layer. Finally, the result was fed into the CNN network and fully connected layers. This has been tested on 360netlab dataset which obtained the best classification accuracy by 0.9713 [32]. Another research [15] which propose hybrid CNN-BiLSTM which achieve a 0.9311 precision [15].

The use of RNN is also utilized in some studies [25] using Gated Recurrent Unites (GRUs) for domain name detection, without any effort of fracture extraction the model achieved an accuracy of 0.98 on AmritaDGA [25].

1.3. Transformer Based Studies

Various mechanisms, based on transformers, have already been used for detecting DGA domains. In one study [22], the authors proposed a hybrid embedding technique to extract text and bigram-level features, utilizing multi-head attention to detect DGA domains, and resulting in an impressive accuracy of 0.9896. Another transformer-based model proposed a multiclass feature fusion approach [12], using a kernel network for feature extraction and an attention mechanism through a transformer encoder. This model was tested on malicious domains from the 360NetLab and DGArchive DGA datasets. The model achieved 0.9783 on 360NetLab's and 0.9852 using DGArchive for binary classification and for multi-classification they achieved 0.9391 on 360NetLab's and 0.9251 on DGArchive [12].

Despite the existence of these related studies, the challenge of constructing an optimal feature set for detecting DGA domains across different modalities remains an open question and need further investigation. As previously mentioned, most existing research has focused primarily on using only textual features of domain names with LSTM [27, 29] or deep learning methods [6, 10, 15, 17, 18, 24, 25, 26, 30, 32, 37]. In contrast to studies that used feature fusion or hybrid embeddings (e.g., [12, 22]), our study introduces a novel numerical design based on domain specific linguistic features and structural patterns. Our method integrates the statistical and structural characteristics derived from domain names, which is clearly distinguishes our methodology from that of other studies. We incorporated N-gram analysis, entropy measures, consecutive consonant and vowel analysis and digit distribution analysis. These constructed numerical features with the text feature fusion enabled the model to capture the semantic and structural patterns, outperforming both hybrid [22] and kernel-based [12] fusion baselines.

This research study presents three key contributions: first, it combines textual features with numerical features, extracting and utilizing various scales of representation information; second, it employs a hybrid deep learning model that integrates a text transformer with numerical feature embeddings using multi-head self-attention, which is key component of the transformer architecture; and third, it distinguishes between DGA and legitimate domains through a fine-tuned hybrid deep learning model to achieve optimal detection accuracy. Finally, this research the importance of numerical features in DGA detection, proving their

value in comparison to textual features, which have often been overlooked in previous studies. The fine-tuned hybrid model achieved an impressive accuracy of 0.9932, surpassing the performance of all existing works in this domain.

2. Material and Methods

In this study, a methodical approach was employed to ensure a systematic and accurate results. Our approach utilizes a hybrid model that combines both textual features and numerical features within the learning process, as shown in Figure 2. The proposed approach includes data preprocessing, feature extraction, and feeding the extracted features into a machine learning classifier, with the output being a classification of the domain as either DGA or legitimate.

As illustrated in the Figure 2, our state-of-the-art model leverages only the encoder component of the transformer for training textual features, while combining hybrid embedding and CNN training for numerical features using a multi-head attention mechanism and dense layers. The subsequent subsections present a detailed explanation of each phase of the proposed approach.

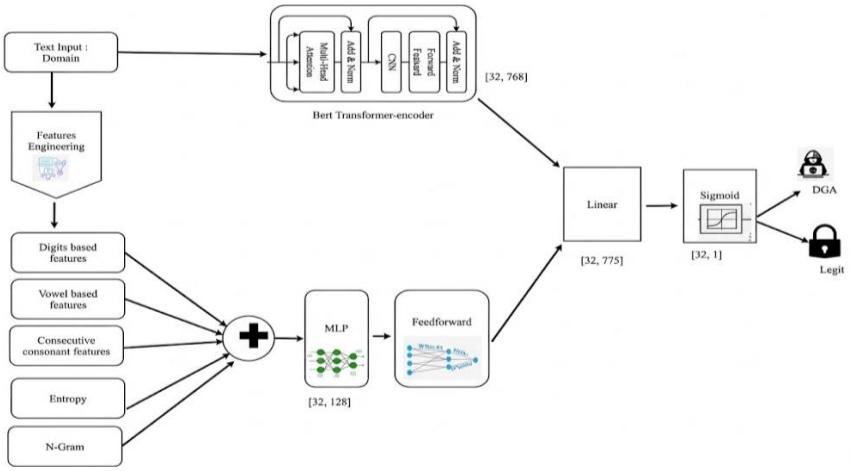


Figure 1. A general workflow of the proposed model.

2.1. Datasets Description

The dataset was collected from Alexa website ranking which contains a total of 160,000 domains labeled as "DGA" and "Legit", the dataset is balanced as shown in Figure 3 and made publicly available on kaggle. Alexa also provided the "top one million" legit domains dataset [1], which is used as a baseline for legitimate domains and feature extraction. Both datasets are publicly available on GitHub and Kaggle.

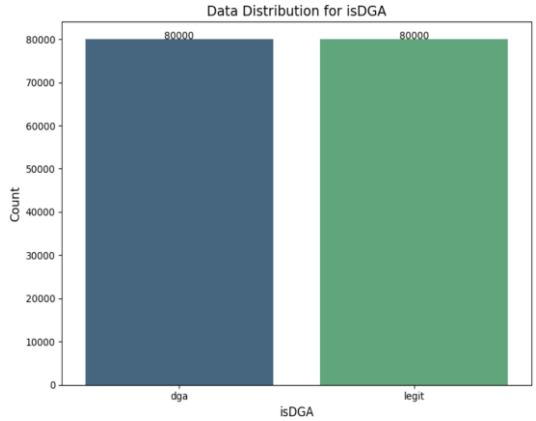


Figure 2. Alexa dataset distribution.

2.2. Feature Extraction and Text Preprocessing

In this research study, we conducted a feature engineering phase prior to modeling to analyze the domains effectively. within addition to extracting textual features, we have also performed a numerical feature extraction to investigate the impact of multimodalities on DGA detection. The numerical features include domain length, contains digit, digit ration, vowel ratio, consecutive consonant ratio, and entropy. These featureswere extracted from the domain names and are discussed in detail in the following subsections.

2.2.1. Digit Analysis

Some of the extracted features are based on the analysis of digits (0-9) in the domain names, such as contains_digit and digit_ratio, which focus on detecting the presence of numbers in the domain names. Digits are often considered as an indicative of DGA domains [36], as they assist distinguish between human readable and machine-generated domains. An example of domain names containing digits is shown in Table 2. As presented in Table 2, contain digits has two possible values: true or false, while digit_ratio calculates the proportion of numeric characters in the domain names, with a decimal value between zero and one.

Table 2. Sample of digits features.

Domain	IsDga	Contains_digit	Digit_ratio
m644136d0.tmodns.net	0	TRUE	0.35
c4w6wpg81xsbopy8a67.ddns.net	1	TRUE	0.25
pub.3gppnetwork.org. mcdonaldswifi.internal	0	TRUE	0.02381

2.2.2. Vowel Analysis

Vowel analysis is also a feature engineering in which we extract new numerical features based on vowel letters: a, e, I, o and u. This kind of analysis help in supporting the linguistic characteristic of domains which can be a feature to recognize the DGA domains [33]. vowel ratio is used to calculate the number of vowels within the domain names and divide it by the length of the domain as shown in Table 3.

Table 3. Samples of vowel ration features.

Domain	IsDga	Vowel_Ratio
m644136d0.tmodns.net	0	0.100000
c4w6wpg81xsbopy8a67.ddns.net	1	0.107143
pub.3gppnetwork.org.mcdonaldswifi.internal	0	0.261905

2.2.3. Consecutive Consonant Analysis

Consecutive Consonant Analysis is another linguistic feature that evaluates the presence of sequential consonant sounds within the domain names. This feature is important because it renders the domain difficult for humans to read, which serves as a distinguishing characteristic of DGA domains [34]. An example of this feature is in Table 4.

Table 4. Sample of consecutive consonant features.

Domain	IsDga	Consecutive_Consonants_ Ratio
m644136d0.tmodns.net	0	0.400000
c4w6wpg81xsbopy8a67.ddns.net	1	0.535714
pub.3gppnetwork.org.mcdonaldswifi.internal	0	0.595238

2.2.4. Domain Names Entropy

Typically, entropy measures the uncertainty in estimating the value of a random variable, indicating the randomness and unpredictability of characters within a domain name [3]. In our research, we utilized Shannon entropy, a widely used concept in information theory and numerical data analysis. The entropy was calculated using Equation (1):

$$H = -\sum p_i log_s(p_i) \quad (1)$$

Where H represents Shannon entropy, P_i is the probability of i^{th} character in the domain, and the summation is performed over all unique characters in the domain name [23]. A sample of domain name entropy values is shown in Table 5 below. Where higher values of entropy indicate that the domain is less human-readable and is more likely to be machine-generated.

Table 5. Entropy feature.

Domain	IsDga	Entropy
m644136d0.tmodns.net	0	3.621928
c4w6wpg81xsbopy8a67.ddns.net	1	4.235926
pub.3gppnetwork.org.mcdonaldswifi.internal	0	4.225185

2.2.5. N-Gram Analysis

N-gram analysis (where n ranges from 1 to 5) as illustrated in [5], is highly valuable as it helps identify meaningful words within noisy, ambiguous and diverse user inputs. This technique is crucial for tasks like information retrieval and NLP feature extraction. It works by breaking down the text into consecutive sequences of n characters, known n-grams, and analyzing each n-gram pattern to assess whether a domain is legitimate or not [7]. In this research study, we utilized the Alexa top one million domains dataset [1] as a reference for n-grams. Then, we generated trigrams, or 3-grams, which, as clearly illustrated in [5] provide richer semantics compared to 2-grams and 4-grams. Additionally, trigrams offer a balance between the increased contextual information it provides and the effective statistical methods used to handle sparse data when understanding words. For each domain and computed the intersection between the domain's n-grams and the reference n-grams. Finally, the number of matching n-grams in these intersections was then counted. An example of the n-gram data is presented in Table 6.

Table 6. N-gram features.

Domain	IsDga	Ngram_Matches
m644136d0.tmodns.net	0	9
c4w6wpg81xsbopy8a67.ddns.net	1	10
pub.3gppnetwork.org.mcdonaldswifi.internal	0	33

Numerical and textual features are both used to enhance the model performance by first tokenizing the domain name text feature and then fed into BERT text Transformer which produces a vector of shape [32, 768], then it is integrated with the numerical features such as n-gram, entropy, consecutive consonant features, vowel ratio and digits ration. The aforementioned numerical features are concatenated to produce a combined vector, which is fed into MLP feed-forward neural network. As clearly shown in Figure 2, the output of BERT encoder of shape [32, 768] and the output of MLP of shape [32, 128] are concatenated to produce a vector of [32, 896] in which it's passed to linear layer. This concatenation improves the performance of the model by making use of rich and diverse modalities through the process of features engineering, the model will learn from multi-modalities representations of numerical and text features which will increase the robustness of the model and reduce any chance of overfitting.

2.3. Proposed Model

After extracting the new numerical features from the domain names, we proceeded with modelling the proposed approach to detect the DGA domains. For this, we employed the BERT text transformer, leveraging its attention mechanism for enhanced performance.

2.3.1. BERT Model Structure and Parameters

Bidirectional Encoder Representations Transformers

(BERT) is a language representation model proposed by Google researchers of Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova which uses a combination of masked language modeling objective and next sentence prediction [14]. Thus, BERT text classifier, represented the best transfer learning has outperformed traditional ML models [31]. The BERT model requires text data format as input, then the input token must be modified. It requires two steps of preprocessing, first: Canonicalization, where numbers, punctuations, and special characters are removed and some uppercase characters are converted to lowercase. Second: Tokenization, using the Bert-base-uncased transformer, tokenization is done by separating the input text into new entities called tokens and transforming them into numerical format in order to be processed by the model [28]. Figure 4 shows the architecture of the transformer model, as a neural sequence transduction model, it has an encoder-decoder structure. The encoder contains N identical layers, each one of N layers has two components, multi-head self-attention mechanism and fully connected feed-forward network with residual connection and normalization between each layer. On the other hand, the decoder is also consisting of N identical layers with two sublayers exactly same as encoder with modified multi-head attention sublayer which is responsible on preventing positions from attending to subsequent positions [19]. In this research study, we used BERT base which is configured as in Table 7, model specifications.

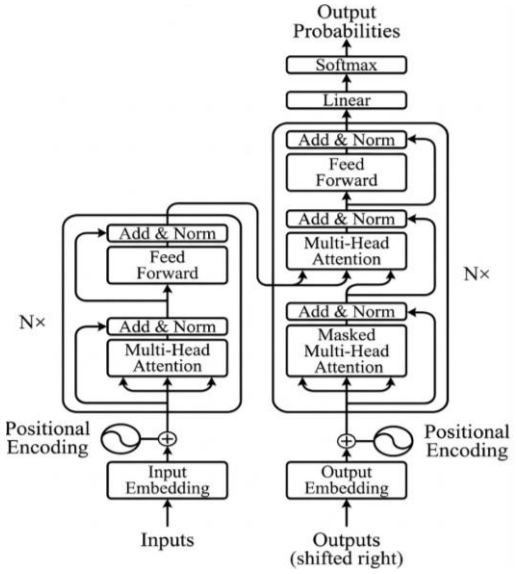


Figure 4. BERT text transformer structure [19].

Table 7. BERT transformer model specifications.

Transformer layers	12
Hidden size	768
Attention heads	12
Parameters	110

2.3.2. Attention Mechanisms

The self-attention mechanism [31] Differs from traditional attention mechanisms in that it directly captures relationships between features within the same sequence, allowing feature extraction and context acquisition to be processed in a unified manner. This mechanism has proven to be able to effectively compute the long-range dependence of features. In our experiment, we incorporate BERT to evaluate its effectiveness in text classification when used as a fixed feature extractor, following the approach proposed by [7]. More specifically, we use the bert-base-uncased version, which consists of 12 layers, 12 attention heads, and a hidden size of 768, resulting in a total of 110 million parameters. As BERT is not fine-tuned in our setup, its parameters remain frozen during the training process as suggested by [7].

2.3.3. Multi-Layer Perceptron (MLP)

The numerical features are fed into MLP. Perceptron which is the most basic form of neural network architecture without hidden layers. This neural network simplest form that can be used for classification problems by taking input, applying some weights, summing them and applying the activation function. In this work, we used ReLu activation function which is the typical one for simple MLPs [35].

2.4. Model Training and Evaluation

Prior to model training, it's important to adjust the model hyperparameter; these parameters control the learning process and evaluate the model performance on unseen data. it optimizes the performance metrics by testing different set of hyperparameter combinations to ensure the robustness and realizability of the model. This subsection describes the training parameters and evaluation metrics used in our DGA detection model.

2.4.1. Hyperparameter Fine-Tuning

The optimization of model hyperparameters has considered in order to adjust the training process, examples of hyperparameters are shown in Table 8. The selection of tokenizer, loss function, dropout rate, batch size, scheduler, optimizer, weight decay and number of epochs. During the tokenization phase the bert-base-uncased tokenizer is used because it's the most efficient tokenizer in the field of domain classification. The model was first trained with default parameters then it's tuned for the optimal results. As shown in Table 8, a learning rate of 2e-5 guaranteed the optimal results with a weight decay of 1e-4, which is the default implementation of l2 regularization [4], tradeoffs between these two main hyperparameters have been done in order to make the best of the model. AdamW optimizer was used since it's the most efficient in handling the weight decay, the model was fully converged with 5 epochs and stopped after two performance degradations. The Binary Cross Entropy Loss function was used because it's very effective for binary classification tasks. Furthermore, a learning

scheduler is used to dynamically tune the learning rate during the training process, which can help the model to converge and avoid getting stuck in the local minimum [2], we used ReduceLROnPlateau scheduler with scheduler patience equal to two consecutive epochs.

Table 8. BERT-MLP hyperparameter tuning.

Parameter name	Value
Tokenizer	bert-base-uncased
Loss Function	BCEWithLogitsLoss
dropout_rate	0.4
batch_size	32
scheduler	ReduceLROnPlateau
scheduler_patience	2
optimizer	AdamW
epochs	5
Learning rate	2e-5
weight_decay	1e-4

2.4.2. Model Evaluation

A lot of performance metrics are used to evaluate the proposed model, which include accuracy, confusion matrix, precision, recall, and F1-score. These metrics are calculated for our model as the following equations: Where TP is the true positive, TN is the True Negative, FP is the false positive and FN is the false negative. The accuracy, as shown in Equation (2), measures the model's overall performance. Precision, defined in Equation (4), indicates the rate of correct predictions among all positive predictions. In our study, precision represents the proportion of correctly predicted DGA domains out of all domains predicted as DGA or legitimate. Higher precision values indicate fewer False Negative (FN) cases. Recall, also known sensitivity and represented in Equation (3), measures the proportion of actual positive cases that are correctly identified by the model. In this research work, recall focuses on the proportion of correctly classified DGA domains out of all actual DGA domains. Also, higher recall values suggest fewer FN cases. The F1 score, shown in Equation (5), combines both the precision and recall into a single metric by calculating their harmonic mean. The significance of the F1-score lies in it is ability to balance the impact of both false positives and false negatives, providing a more comprehensive evaluation of the model's performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

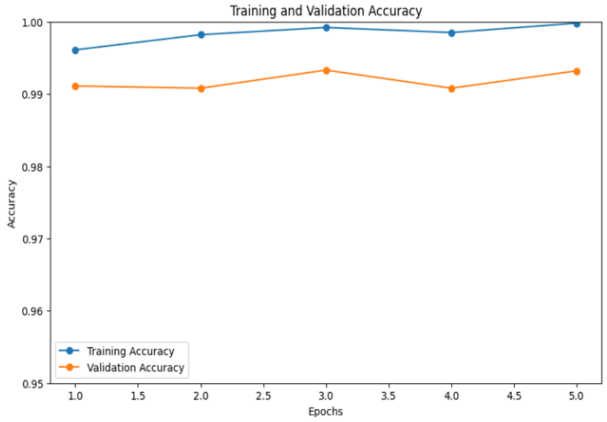
3. Results and Discussion

In this section, the presented the results obtained by applying our proposed methodology on the Alexa DGA domains dataset will be presented. Feature engineering such as numerical feature extraction and text embedding, is fused into the BERT transformer in order to achieve the best results of the DGA detection.

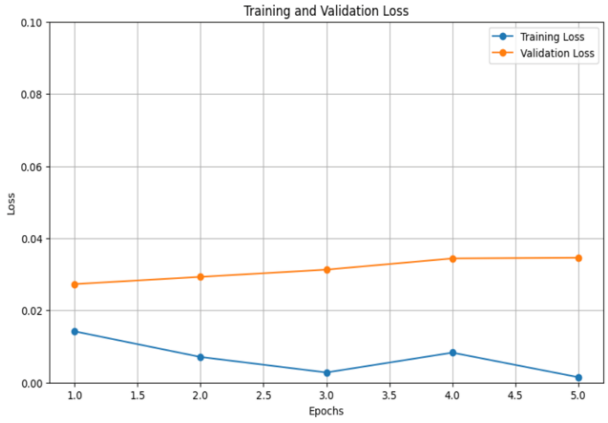
3.1. Training and Validation Accuracy and Loss Analysis

The accuracy and loss curves are considered the most important metrics to measure the model performance and generalization ability. For the hybrid BERT transformer model, both curves are steadily decreased and increased, respectively, in the first three epochs as shown in Figure 5-a) and (b). The training was stopped at epoch 5 where the loss values are almost zero. On the other hand, the training accuracy reached its maximum value of 1.0 while the validation accuracy is 0.9932. This result indicates that the model fits the data very well, with minimal signs of overfitting.

The training has been fluctuated at epoch 4, the validation accuracy did not improve or match the epoch 3 which cause the early stop to be triggered and stop the model training at epoch 5. In general, the accuracy curves show a stable upward trend, while the loss curves show a steady downward trend, indicating the models' stabilization and generalization of the model.



a) Training and validation accuracy



b) Training and validation-loss.

Figure 5. Performance Metrics Evaluation.

3.2. Classification Report Analysis

The Classification report provides very important performance indicators such as Accuracy, Precession,

Recall, F1-Score. The classification report shown in Figure 6 illustrates that model performs perfectly well on both classes of "DGA" and "Legit" achieving highest accuracy of 0.9932 and highest precision, recall and f1-score of DGA as 0.9935, 0.9928 and 0.9932 respectively. The very high values of recall and precision indicate that the model has very few cases of false positives and false negatives for both classes.

```
Classification Report:
              precision    recall  f1-score   support

       Legit     0.9928    0.9935    0.9932     16000
         DGA     0.9935    0.9928    0.9932     16000

    accuracy                         0.9932     32000
   macro avg     0.9932    0.9932    0.9932     32000
weighted avg     0.9932    0.9932    0.9932     32000
```

Figure 6. Classification report.

3.3. Confusion Matrix Analysis

For more details about the TP, TN, FP, and FN for each class we have taken advantages of the confusion matrix, depicted in Figure 7. It visualizes the performance of the classification model by summarizing the number of predictions that are true positives, true negatives, false positives, and false negatives for each class, with the diagonal representing correct predictions, and it's clear that for Legit the model successfully predicts 15896 out of 160000 and for DGA it successfully predicts 15886 out of 160,000 which reveal the state of art in the model used. Both the confusion matrix and the classification report show a very strong performance of our hybrid text transformer. With very high accuracy and reliability for both binary classes.

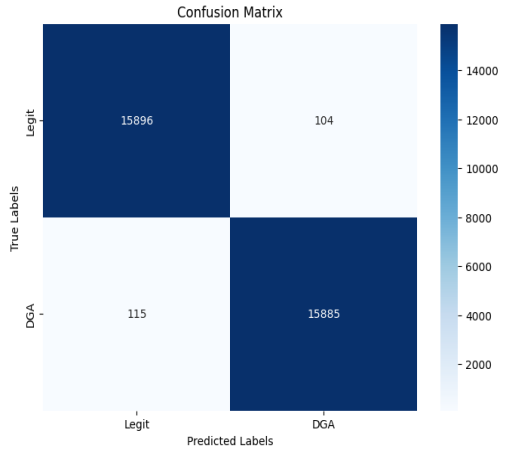
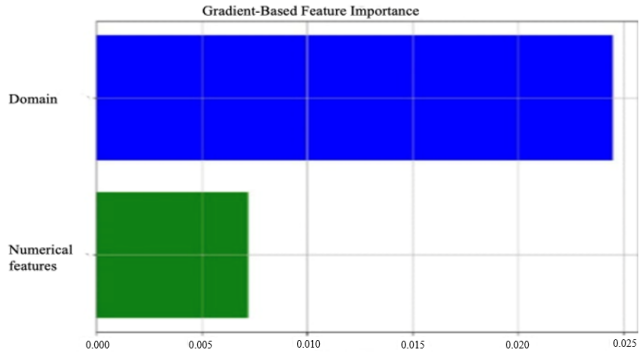


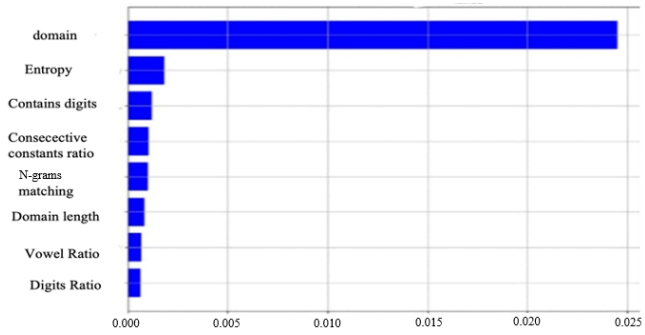
Figure 7. Confusion matrix.

3.4. Features Importance

To further demonstrate the effectives of numerical features in the domain of DGA detection, we calculated the gradient magnitude of these features. As shown in Figure 8, the numerical features have achieved an unneglectable importance compared to text features. More specifically, the numerical features have a gradient magnitude of 0.007. The absolute importance of each numerical feature is visualized on Figure 8-a), confirming that the model relies heavily on the numerical features extracted earlier. As shown in Figure 8-b) the most importance numerical features are Entropy, followed by contain-digit, and then connective-consonants-ratio, where the least important feature is the digit-ratio.



a) Gradient feature importance for text and numerical featurs.



b) Decomposition of numerical features importance.

Figure 8. Features importance for text and numerical features.

3.5. Models Comparison

The results presented in Table 9 provide a comparison between the performance of the proposed model and the recent transformer and neural network-based studies. One can see that the proposed model outperforms the other related studies in terms of accuracy and generalization to various datasets. Although traditional Neural Network methods, such as RNN and CNN [30, 37], achieved notable results, they only used just the domain name text features. In contrast, our proposed model utilized a combination of textual and numerical modalities such as domain names, digits-based features, vowel-based features, entropy, N-gram, consecutive consonant features. Our model outperformed the results of CNN by 0.53%. On the other hand, when compared to other transformer-based studies that utilized text transformers, the incorporation of multimodalities and the fusion of numerical features resulted in a 0.73% improvement accuracy.

Table 9. A comparison of the results achieved with the state of the art.

Method	Model	ACU	Year
Neural network traditional method	[30] RNN	0.988	2024
	[26] CNN-BiLSTM	0.9311	2023
	[37] CNN-BiLSTM	0.9713	2023
Transformer based studies	[24] DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network	0.9391	2024
Proposed model	BERT Transformer+ numerical features fusion	0.9932	2025

4. Conclusions

The growing prevalence of DGA domains poses a significant threat to cybersecurity, leading to both security vulnerabilities and financial losses. This paper demonstrates the effectiveness of utilizing a hybrid approach that combines text transformers with numerical feature, specifically the BERT transformer, for the early and accurate detection of DGA domains. The model achieved an impressive accuracy of 0.9932, demonstrating its reliability in detecting malicious DGA domains. This high accuracy is essential for early detection, which significantly reduces the risk of compromise by a malicious software, and thereby enhancing the security of the enterprises and end-users. In this work, an advanced feature fusion method was developed, which integrates numerical features using a fully connected feedforward network, which played a vital role in improving the performance of the proposed model, and thus ensuring robustness against variations in DGAs.

In addition, the evaluation metrics, including an F1-score of 0.9932, along with the confusion matrix, demonstrate exceptional agreement and performance in detecting the DGA domains. These metrics not only validate the effectiveness of the proposed model, but also highlight its generalizability and applicability to unseen data. The results underscore the urgent need to address the threat of DGA domains and advocate for the integration of AI-driven tools in cybersecurity to combat issues such as theft, phishing and data breaches. Future research should focus on expanding these models with larger datasets and exploring additional deep learning architectures to translate these advancements into practical applications. Moreover, the use of explainable AI (XAI) techniques will be crucial for interpreting the decision-making process in DGA domain detection.

Acknowledgment

The completion of this paper is the result of independent work. No external assistance was involved.

References

[1] Al Messabi K., Aldwairi M., Al Yousif A., Thoban A., and Belqasmi F., "Malware Detection Using DNS Records and Domain Name Features," *in Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, Amman, pp. 1-7, 2018. https://doi.org/10.1145/3231053.3231082

[2] Alawneh H. and A. Hasasneh. "Survival Prediction of Children after Bone Marrow Transplant Using Machine Learning Algorithms," *The International Arab Journal of Information Technology*, vol. 21, no. 3, pp. 394-407, 2024. https://doi.org/10.34028/iajit/21/3/4

[3] Ali A., Naeem S., Anam S., and Ahmed M., "Entropy in Information Theory from Many Perspectives and Various Mathematical Models," *Journal of Applied and Emerging Sciences*, vol. 12, no. 2, pp. 156-165, 2022. https://doi.org/10.36785/jaes.122548

[4] Al-Kababji A., Bensaali F., and Dakua S., "Scheduling Techniques for Liver Segmentation: ReduceLROnPlateau Vs OneCycleLR," *Intelligent Systems and Pattern Recognition*, pp. 204-212, 2022. https://doi.org/10.1007/978-3-031-08277-1_17

[5] Avasthi S., Chauhan R., and Acharjya D., "Processing Large Text Corpus Using N-Gram Language Modeling and Smoothing," *in Proceedings of the Second International Conference on Information Management and Machine Intelligence: ICIMMI*, pp. 21-32, 2020. https://doi.org/10.1007/978-981-15-9689-6_3

[6] Chen J., Qiu J., and Chen Y., "A Hybrid DGA DefenseNet for Detecting DGA Domain Names Bbased on FastText and Deep Learning Techniques," *Computers and Security*, vol. 150, pp. 104232, 2025. https://doi.org/10.1016/j.cose.2024.104232

[7] Devlin J., Chang M., Lee K., and Toutanova K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv Preprint*, vol. arXiv: 1810.04805, pp. 1-16, 2019. https://doi.org/10.48550/arXiv.1810.04805

[8] Ding L., Du P., Hou H., Zhang J., and et al., "Botnet DGA Domain Name Classification Using Transformer Network with Hybrid Embedding," *Big Data Research*, vol. 33, 2023. https://doi.org/10.1016/j.bdr.2023.100395

[9] Fadziso T., Thaduri U., Dekkati S., Desamsetti H., and Ballamudi V., "Evolution of the Cyber Security Threat: An Overview of the Scale of Cyber Threat," *Digitalization and Sustainability Review*, vol. 3, no. 1, pp. 1-12, 2023. https://upright.pub/index.php/dsr/article/view/79

[10] Highnam K., Puzio D., Luo S., and Jennings N., "Real-Time Detection of Dictionary DGA Network Traffic Using Deep Learning," *SN Computer Science*, vol. 2, no. 110, pp. 1-17, 2021. https://doi.org/10.1007/s42979-021-00507-w

[11] Hwang C., Kim H., Lee H., and Lee T., "Effective DGA-Domain Detection and Classification with TextCNN and Additional Features," *Electronics*, vol. 9, pp. 1-18, 2020. https://doi.org/10.3390/electronics9071070

[12] Keyword Research, Competitive Analysis, and Website Ranking, Alexa, https://www.alexa.com/, Last Visited, 2025.

[13] Liang J., Shuhui C., Wei Z., Shuang Z., and Ziling W., "HAGDetector: Heterogeneous DGA Domain Name Detection Model," *Computers and Security*, vol. 120, pp. 102803, 2022.

https://doi.org/10.1016/j.cose.2022.102803

[14] Merchan E., Brizuela R., and Carvajal S., "Comparing BERT Against Traditional Machine Learning Models in Text Classification," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 4, pp. 352-356, 2023. https://doi.org/10.47852/bonviewJCCE3202838

[15] Nadagoudar R. and Ramakrishna M., "Algorithmically Generated Domain Names Detection Using Gated Recurrent Unit Deep Learning," *Journal of Electrical Systems*, vol. 20, no. 7, pp. 469-481, 2024. https://doi.org/10.52783/jes.3342

[16] Nadagoudar R. and Ramakrishna M., "DGA Domain Name Detection and Classification Using Deep Learning Models," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 7, pp. 306-315, 2024. https://dx.doi.org/10.14569/IJACSA.2024.0150730

[17] Namgung J., Son S., and Moon Y., "Efficient Deep Learning Models for DGA Domain Detection," *Security and Communication Networks*, vol. 2021, pp. 1-15, 2021. https://doi.org/10.1155/2021/8887881

[18] Shahzad H., Sattar A., and Skandaraniyam J., "DGA Domain Detection Using Deep Learning," *in Proceedings of the IEEE 5th International Conference on Cryptography, Security and Privacy*, Zhuhai, pp. 139-143, 2021. https://doi.org/10.1109/CSP51677.2021.9357591

[19] Singh J. and Banerjee R., "A Study on Single and Multi-Layer Perceptron Neural Network," *in Proceedings of the Third International Conference on Computing Methodologies and Communication*, Erode, pp. 35-40, 2019. https://doi.org/10.1109/ICCMC.2019.8819775

[20] Soleymani A. and Arabgol F., "A Novel Approach for Detecting DGA-Based Botnets in DNS Queries Using Machine Learning Techniques," *Journal of Computer Networks and Communications*, vol. 2021, pp. 1-13, 2021. https://doi.org/10.1155/2021/4767388

[21] Sood A. and Zeadally S., "A Taxonomy of Domain-Generation Algorithms," *IEEE Security and Privacy*, vol. 14, pp. 46-53, 2016. https://doi.org/10.1109/MSP.2016.76

[22] Stampar M. and Fertalj K., "Applied Machine Learning in Recognition of DGA Domain Names," *Computer Science and Information Systems*, vol. 19, no. 1, pp. 205-227, 2022. https://doi.org/10.2298/CSIS210104046S

[23] Suen C., "N-Gram Statistics for Natural Language Understanding and Text Processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 164-172, 1979. https://doi.org/10.1109/TPAMI.1979.4766902

[24] Sun X. and Liu Z., "Domain Generation Algorithms Detection with Feature Extraction and Domain Center Construction," *Plos One*, vol. 18, pp. 1-25, 2023. https://doi.org/10.1371/journal.pone.0279866

[25] Tang J., Guan Y., Zhao S., Wang H., and Chen Y., "DGA Domain Detection Based on Transformer and Rapid Selective Kernel Network," *Electronics*, vol. 13, no. 24, pp. 1-16, 2023. https://www.mdpi.com/2079-9292/13/24/4982#

[26] Thakur K., Alqahtani H., and Kumar G., "An Intelligent Algorithmically Generated Domain Detection System," *Computers and Electrical Engineering*, vol. 92, pp. 107129, 2021. https://doi.org/10.1016/j.compeleceng.2021.107129

[27] Tran D., Mac H., Van T., Tran H., and Giang N., "A LSTM based Framework for Handling Multiclass Imbalance in DGA Botnet Detection," *Neurocomputing*, vol. 275, pp. 2401-2413, 2018. https://doi.org/10.1016/j.neucom.2017.11.018

[28] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., and et al., "Attention Is All You Need," *in Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, pp. 6000-6010, 2017. https://dl.acm.org/doi/10.5555/3295222.3295349

[29] Vij P., Nikam S., and Bhatia A., "Detection of Algorithmically Generated Domain Names Uing LSTM," *in Proceedings of the International Conference on COMmunication Systems and NETworkS*, Bengaluru, 2020. https://doi.org/10.1109/COMSNETS48256.2020.9027342

[30] Vranken H. and Alizadeh H., "Detection of DGA-Generated Domain Names with TF-IDF," *Electronics*, vol. 11, no. 3, pp. 1-28, 2022. https://doi.org/10.3390/electronics11030414

[31] Wang C., Nulty P., and Lillis D., "A Comparative Study on Word Embeddings in Deep Learning for Text Classification," *in Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, Seoul Republic, pp. 37-46, 2020. https://doi.org/10.1145/3443279.3443304

[32] Wang Y., Pan R., Wang Z., and Li L., "A Classification Method Based on CNN-BiLSTM for Difficult Detecting DGA Domain Name," *in Proceedings of the IEEE 13th International Conference on Electronics Information and Emergency Communication*, Beijing, pp. 17-21, 2023. https://doi.org/10.1109/ICEIEC58029.2023.10200702

[33] Wang Z., Jia Z., and Zhang B., "A Detection Scheme for DGA Domain Names Based on SVM," *in Proceedings of the International Conference on Mathematics, Modelling, Simulation and Algorithms*, Chengdu, pp. 257-263, 2018. https://doi.org/10.2991/mmsa-

18.2018.58
[34] Wong A., Detecting Domain-Generation Algorithm Based Fully-Qualified Domain Names with Shannon Entropy, Technical Report, 2023. file:///C:/Users/acit2k/Downloads/2304.07943v1.pdf
[35] Xie Z., Sato I., and Sugiyama M., "Stable Weight Decay Regularization," *arXiv Preprint*, vol. abs/2011-11152, pp. 1-18, 2020. https://openreview.net/forum?id=YzgAOeA67xX
[36] Zhao D., Li H., Sun X., and Tang Y., "Detecting DGA-based Botnets Through Effective Phonics-Based Features," *Future Generation Computer Systems*, vol. 143, pp. 105-117, 2023. https://doi.org/10.1016/j.future.2023.01.027
[37] Zhao K., Guo W., Qin F., and Wang X., "D3-SACNN: DGA Domain Detection with Self-Attention Convolutional Network," *IEEE Access*, vol. 10, pp. 69250-69263, 2021. https://doi.org/10.1109/ACCESS.2021.3127913
[38] Zhou Z. and Zhu L., "Research on Domain Generation Algorithms and their Detection," *International Journal of Science*, vol. 10, no. 11. pp. 63-69, 2023. http://www.ijscience.org/download/IJS-10-11-63-69.pdf

Suhad Malayshi received a Bachelor's degree in Communication Engineering from AN-Najah National University in 2015. She has seven years of experience in several positions in information technology, she is completing her Master degree in Computer Science at Arab American University. Suhad is honing her skills in Machine Learning, Deep Learning, and Data Analysis using Python. She has a good experience in python, cloud operation and DevOps.

Ahmad Hasasneh earned his B.Sc. in Computer Systems Engineering from Palestine Polytechnic University in 2005 and his M.Sc. in Computer Graphics and Programming from the University of Hull in 2006. He taught at Hebron University before receiving a Ph.D. in artificial intelligence and machine learning from Paris University in 2012. He has since held academic and leadership roles at Hebron University, PTUK, and Palestine Ahliya University, where he helped establish programs in multimedia and smart systems engineering. Since 2024, he has been an associate professor and department head at the Arab American University. His research focuses on Machine Learning Applications in Medical Diagnostics, with active projects under the Palestinian German and Palestinian-Quebec Science Bridges, and collaborations with institutions in Germany, Canada, UAE, and Portugal. He has published widely in international journals and conferences.