J. Electrical Systems 20-7s (2024):469-481

Ranjana B Nadagoudar[1],

M Ramakrishna[2]

Algorithmically Generated Domain Names Detection Using Gated Recurrent Unit Deep Learning

JES Journal of Electrical Systems

Abstract

The modern malware increasingly employs domain generation algorithms (DGAs) to evade traditional DNS query detection methods, such as blacklisting or reverse engineering of suspicious domain names. These algorithms generate vast numbers of random domain names to establish communication with Command and Control (C&C) servers, posing significant challenges for detection. Previous research has predominantly relied on classical machine learning algorithms, necessitating manual feature extraction and classification, which is both time-consuming and labour-intensive this paper, we propose a deep learning-based architecture for detecting DGA-generated domain names. Our model utilizes recurrent networks with gated recurrent units (GRUs) for domain name detection. By converting domain names into vectors and employing GRUs, the model autonomously learns features, eliminating the need for manual intervention in feature extraction. Compared to traditional methods, our approach reduces time costs associated with feature extraction. The experimental result demonstrates the effectiveness of our proposed GRU achieving 98% accuracy, 94% recall rate, 93% precision, and an Area Under the Curve (AUC) of 99.6%. The GRUarchitecture outperforms LSTM models in terms of recall rate and accuracy while requiring less computational resources, indicating significant performance enhancement.

Keywords: -Cyber Security, Botnet, Deep Learning, DGA (Domain Generation Algorithm) and DNS (Domain Name System).

1.Introduction

Despite of advances in the information and communication technology, even then it is very much difficult to respond to cyber-attacks using command and control servers. The malware family attempts to establish communication with C&C server to flood the unsolicited malicious activities. A C&C server basically manages the botnet, and which contains a set of infected devices controlling malicious code from remote attackers [2]. The internet protocol address and domain names are being easily blacklisted in a DNS blacklist. Although, this is not a potential from the attacker's point of view because it can be easily detected and blocked by the various security experts. Now a days the there is a much advancement in modern malwares, so identify and detect such characteristics of malwares we need to make use of domain generation algorithms.

The approach used to fight with DGA generated domains is to identify DNS requests and also detects the algorithmically generated domains using ML classification methods. Preliminary work on DGA acquisition may be categorized as real-time detection and retrospective. Real-time detection performs classification using domains without requiring additional status information. However, those strategies are faster and less effective compared to retrospective [1, 2]. In contrast, the retrospective approach does not perform well in real time, since it has to translate massive amounts of speculation into wider domains. These retrospective techniques are computationally expensive.

Nowadays attackers often use DGAs to try and prevent domains from their C&C servers. A DGA consistently generates multiple domains [9]. Since the algorithmically generated domains name is being constantly changing, there is a requirement to identify the dynamic nature of domains that establish communication with command-and-control servers. Further it is required to categorize which DGA domain produces, as DGAs have divergent advantages and different types, such as ransom ware and the Bank Trojan [4]. Furthermore, the classification of these malwares becomes a significant challenge, as the malicious domain and the legitimate domains having an imbalance issue with different events for each class, hence the classification of these domains becomes a challenging problem.

The simplest approach to classify and detect the DGA domains makes use of blacklist method. However, this method is not effective for an oversized variety of latest domains which are constantly generated [5]. Other method used for the DGA is reverse engineering this is mainly used to decode the DGA, it identifies algorithmically generated domain names and later it blocks these domains. However, this method can prevent malicious domains in smaller number and very much time consuming. Therefore, this method is slow and not efficient compared to other methods. The classical machine learning methods are used to combat with these modern malwares. The main disadvantage with Machine Learning is to manually pick the features [2]

In recent days, to resolve the issues of existing DGA domain detection strategies, several deep learning-based techniques have been proposed [6, 7] which will improve performance without requiring any additional knowledge. Compared to existing machine learning methods the deep learning approaches gives better performance for various applications of cyber security. The proposed deep learning-based strategies don't need a long feature extraction method, and as black boxes, it's troublesome for attackers to perform reverse engineering strategies. Deep learning will learn best options by itself. Therefore, it fully avoids the feature engineering. These strategies also can find DGA domains in real time victimization solely the domains without required of further information. Convolution neural networks and recurrent neural networks are largely employed for DGA domain name detection [13]. However, recent advancement in deep learning shows that there are many types of RNNs mainly the classical RNNs has an issue of vanishing and error gradients issue and difficult to process longer sequences. The advanced models of recurrent networks are LSTM and.

[1*]Visvesvaraya Technological University, India, ranjanapriya8@gmail.com

[2] Vemana institute of technology, India, mramakrishna15975@gmail.com

GRU. Literature survey shows that LSTM has been largely studied in detail for DGA domain name detection. A recent study shows that GRU is an enhanced model of LSTM, and the model has achieved significant performances with less computational cost The remaining of the paper is organized as follows. The DGA domain and its related work are presented in Section 2. The background information of domain generation algorithm is explained in Section 3. The Section 4 presents theoretical information of deep learning techniques. In Section 5, the proposed system architecture of deep learning is explained. The Section 6 describes about the dataset. The performance metrics are described in Section 7 and in Section 8, the experimental results are described

2. Related Work

The domains being queried during a network will be acquired through name Service (DNS) servers or an observation server. One will build a system to investigate every queried domain and provides a finding of fact, either benign or DGA generated, that guides additional actions like communication block or rectification. There has been innumerable works wiped out this field by exploitation approaches that involve classical machine learning, rule-based learning, deep learning and even reverse engineering. DGA is one in every of the foremost effective and preferred tools within the attackers' toolbox. It's being employed by a spread of malware families to cover the placement of their command-and-control servers.

The problem of differentiating benign domains from algorithmically generated domains is not new concept and the studies have been carried out in the earlier days. The early attempts to minimize the malware occurrences had insufficient training data to apply classical ML approaches [7]. To reduce the limitations of existing methods such as blacklist and other methods have tried to use statistical features of domain names. Thus, existing approaches and techniques were rather statistical. Yadav et al. [11, 12] presented framework for identifying algorithmically generated domain names used by several recent botnet. The author had applied statistical measures for classifying the domain names as legitimate or DGA generated. The drawback of this method is that it often does not detect the different DGA families. The author Woodbridge et al. [13] presented a method that makes use of LSTM to categorize the DGA generated and legitimate domains. LSTMs having benefits over other approaches as they are not dependent on features and makes use of raw domain names as its input [6]. The experimental results shows that LSTM outperformed as compared with random forest with manually engineered features and logistic regression with bigram features. These approaches can perform real time detection, but they are sensitive to the imbalanced dataset which makes difficult to detect domains from minority families.

The earlier issue was addressed from a different perspective by Anderson et al. [14] uses GAN network to produce the adversarial domain names to deceive the classifier. The domain names generated by GAN were included in the training dataset; it results with improved performance for DGA detection. However, the author did not tried testing of this dataset on different DGA families [8]. The proposed method is not effective for detecting the different families of DGA generated domain names.

Many methods make use of statistical features as required information for domain detection. The author Curtin et al. [15] had developed a ML based model comprise of RNN's which makes use of WHOIS as additional knowledge. The deep learning model shows the significant performance improvement [7]. They have introduced a complexity measure for DGA families called smash word score. This combined model proposed by the author outperforms with existing methods on DGA families with high score [8].

Schiavoni et al. [9] had proposed a system called Phoenix, a mechanism that tells benign and DGA-generated domains apart by making use of IP-based features. It detects DGA generated domains using DNS query information and linguistic-based features. Phoenix has some limitations requires registered domains to function. If the bot master is not yet registered but still it can identify the DGA generated domains [16]. This method is not effective for detecting the unknown or unseen DGA's. The accuracy of this model was of about 94.8%.

Tran et al.[17] had presented a framework for both binary classification as well as multi-class classification. It handles the problem of multiclass imbalance in domain detection. The experimental result shows that LSTM is far better compared with cost insensitive. These observations help to analyse the various features of the LSTM.MI algorithm, with this it can achieve better accuracy and f1-score with respect to HMM. The framework uses LSTM for domain name identification and detection. This model is most suitable for imbalanced kind of data.

Chin et al. [29] had developed a ML based framework for identifying and detecting the domains from DGA. Further they have applied the proposed ML techniques to investigate the DGA-based modern malwares. The proposed model comprises two levels containing the classification as first level operation and clustering method as a second level operation. These methods are to detect and identify the algorithmically generated domains. In this work ML based methods apply DNS blacklist for detecting DGA generated domains.

Mac et al. [17] the author has studied many classifiers such as ML based classifiers and deep learning models. The classical machine learning methods have also been used for DGA classification and detection. The experimental result indicates that the deep learning methods have improved performance for DGA classification compared with classical machine learning approaches.

Qiao et al. [18] the author had implemented algorithmically generated domains classification approach relied on LSTM with attention method. Attention method combined with LSTM can efficiently classify the DGA generated domains. In this paper the author has considered 15 distinguished class labels for DGA classification. The experimental results shows that model which the author have proposed performed significantly well compared with existing machine

learning models. The method considered weights of distinguished characters in distinct positions in algorithmically generated domains and further it had achieved better accuracy than LSTM model.
Yu et al. [31, 32] carried out performance comparison of various deep learning models; the recurrent neural network was used for character-based text classification for detecting DGAs. Their experimental results showed that the CNN model gives significant improved performance over machine learning models, it also gives higher accuracy compare with existing classifier such as random forest.
Another study proposes CNN and LSTM models by extracting numerous features from domains to identify and detect the algorithmically generated domains. Further, the author Yu et al compared various ML models for DGA detection. The proposed model gives an accuracy of 72.89 % during their experimental analysis.
Vinayakumar et al. [19], the author had presented approach called LSTM for classifying DGA generated domains. Experiments were carried out on publically available data which showed that LSTM performed better than other classifiers. The proposed model is sensitive to class imbalance.
Vinayakumar et al [20], the author developed a model that gathers traffic data of DNS at the ISP level. Further it identifies the DGA based domains in real-time. They also used many deep learning models such as LSTM, CNN, CNN-LSTM and RNN for modern botnet detection. These methods have performed well compared to classical ML approaches and also give better classification accuracy rate.
Vinayakumar et al [21], the author had designed and developed scalable architecture called apache spark. The proposed model gathers DNS logs data and performs the analysis. The deep learning techniques are being used to detect and gives alert for suspicious domains.
Feng et al. [16] have proposed various deep learning models to classify the domains as DGA and non-DGA domain names. Further the proposed model performs automatic feature extraction and capable for handling massive scales of data. The model gives good accuracy of performance; therefore, the proposed approach is effective and efficient for classifications of domains.
The existing studies shows that deep learning-based approaches achieved better performances compared to the existing studies. Mainly, LSTM has been largely employed for DGA domain name detection. However, LSTM has a greater number of learnable parameters, and it is computationally expensive. Literature survey shows that GRU is an advanced model of LSTM and following the model has employed to DGA domain name detection and detailed analysis experiments are shown with benchmark dataset such as AmritaDGA.

3. Domain Generation Algorithm

Domain generation algorithm is a program that produces massive scales of domain names. The detection methods are of two classes reactionary and real-time. The reactionary method uses non supervised clustering techniques for detection. The bot contained in the infected device has to establish communication with command-and-control server. These DGA generates massive amounts of random domains but only of them leads to successful domain shown in Fig.1. Recent attempts at detecting the algorithmically generated domain names with deep learning techniques have performed well. These deep learning methods basically make use of recurrent neural networks and LSTM architectures. Though the deep character or word embedding have shown significant impact for detecting dictionary DGA.

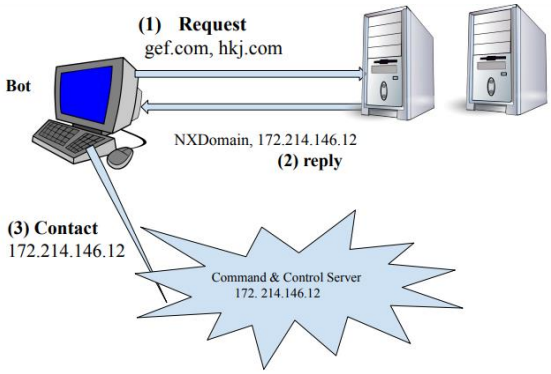

Fig 1: Communication between bot and C2C server.

4. Deep Learning Models

Deep learning is a type of artificial intelligence [6]. It is considered as a subgroup of classical machine learning techniques, unlike the traditional approaches, deep learning may not require feature engineering. Deep learning models have performed well in various applications such as speech processing, computer vision and natural language processing. However, recently it began to use these models in various cyber security applications [33].

4.1Recurrent Neural Network

Recurrent neural network is a powerful type of neural network. Various techniques have been inspired by the recurrent networks. Unlike the traditional feedforward neural network, RNN has an important and significant feature is that it has feedback connections. These networks are very much helpful in modelling the sequence of data. The important characteristic of recurrent network is that it can generate predictive results in the sequential data format that the other techniques can't perform. The recurrent network contains a self-recurrent connection that allowscapturing and transformingof information. A characteristic of recurrent network assists to learn the temporal representation of information. Recurrent network takes the input as x = (x_1, x_2,....,x_t) and which maps to hidden input represented in sequence h = (h_1, h_2, ...,h_t).

$$h_t=\begin{cases} 0 & t=0 \\ tf\,(ht-1, xt) & otherwise \end{cases} \quad (1)$$

Where $t_f$representsaffine transformation of x_t and h_{t-1}. Further the LSTM model was to minimize the gradient issue [32]. The recurrent network RNNcontains a set of gating functions and memory cell. This gating function basically controls the information which is related to memory cell. Various gating functions present in the recurrent network areinput gate, forget gate and output gate. Further the input gate controls the flow of information related to input. To enhance the performance of LSTM, gated recurrent unit was introduced and it has minimalnumber of parameters compared to LSTM and GRU is effectiveand efficient compared to LSTM.

4.2Gated Recurrent Unit (GRU)

Gated recurrent unit is considered as improved model of the recurrent neural network and which is been widely used for machine translation.GRUs was introduced in 2014 by Kyunghyun Cho et al. [1]. GRU uses various gates to control the flow of information. Unlike LSTM the gated recurrent unit does not have a separate cell state (Ct). The gated recurrent unit has only has a hidden state (Ht). This recurrent unit contains the simpler architecture therefore it is faster to train the network. Because of this reason GRU offers performance improvement over LSTM.

To solve the problems related to vanishing or exploding gradients, many variations were developed. One of such variations is the LSTM and most effective variation is the GRU.

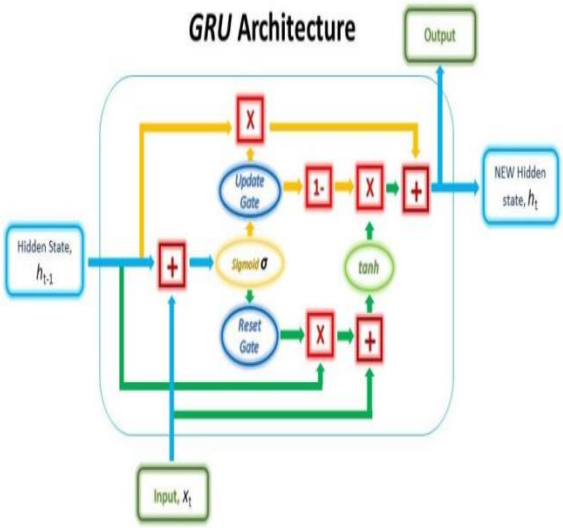

Fig 2**:** The architecture of Gated Recurrent Unit

GRU provides the gatingoperation and flow of information related to gate is mainly controlled by hidden state. To address the issue that arises with RNN. The gated recurrent unitmakesuse of update and reset gatehas two vector entries (0, 1) which can carry out convex combination of operation. Hidden state information should be updated based on the convex operation; it will determine what states needs to be updated whenever needed are. Similarly, the deep network learns to skip the irrelevant information.

The sigmoid function used to transform the values lies in the range [0, 1]. It allows the gate to filter out the between the less-useful and more-useful information in the successive steps.

$$\text{gate reset} = \sigma\,(\text{Winputreset} * \text{xt} + \text{Whiddenreset} * \text{ht} - 1)(2)$$

The gated recurrent network is trained using back-propagation algorithm. Weights presents in the above equation will be updated in such way that the vector will learn to retain only the useful features.

The update gate is evaluatedbyusing the prior hidden state information and current input data. The update gate is mainly responsible for identifying the amount of prior information that needs to be passed along the next state.

$gateupdate=\sigma(W_{input}update*x_t+W_{hidden}update*h_{t-1})(3)$

The importance of the update gate is to help the model to identify how much the prior information stored in the previous hidden state, which needs to be retained for the future.

$h_t = r \odot (1 - gate_{update}) + u(4)$

To improve the capacity of recurrent network and also providing the ease of training model, the gated recurrent unit have been used. Further with help of hidden unit information the gradient problem in the network can be minimized.

5. Proposed Architecture

The proposed method includes the domain names as an input which contains series of characters, later transforms these characters into a series of vectors. In our proposed work we have adopted Keras character level embedding. After translating the character representation into series of vector and in next step series of vectors are provided for deep learning layers. Further processes the series of vectors in the sequential order and at each step it updates the hidden vector state information. Finally, the model will be able to perform the classification task to classify the domains as benign or DGA generated.

The outline of our proposed approach is shown in Fig. 3. The model represents three stages of operation, first one is character encoding here it maps each character into real valued vector and second one is feature representation and finally a classification method to classify the domains. The proposed model is evaluated on binary classification that classifies whether domains are benign or DGA generated. Deep learning models are trained and evaluated on the dataset for DGA detection.

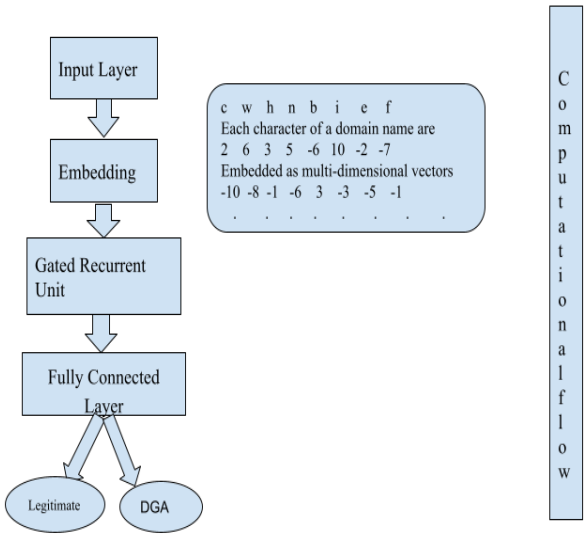

Fig 3: Proposed deep learning approach

5.1 Domain name character embedding: In this paper we have utilized keras for DGA detection. In the initial phase of the operations are going to be assigned and later these weights will learn all the characters in the dataset. Further these embedding layer tries to map each one of character in the dataset to a 128 length of real value represented in the vector. The domain name character level embedding makes use of recurrent neural network to determine numerical value representation by looking at their character level compositions.

5.2 Feature representation (Learning): For representing the features various deep layers have been used such as LSTM, RNN and GRU. These structures willcapture the sequential information. The pattern-matchingapproachalong with the deep learning layer looks effective and efficient compared with regular expression. The regular expression outputs a binary value butthe deep learning models produces a continuous value which in turn represents that how much the pattern is matched.

5.3Recurrent Layers: We have used various recurrent networks such as LSTM, RNN and GRU. Here the number of recurrent unit is set to 128 based on the knowledgeacquired from the selection method called hyper parameter selection. These recurrent network layers mainly represent the sequential information from the embedding layer output. A unit in recurrent neural network makes use of activation function that has value in range [-1, 1]. The gate uses logistic sigmoid function which uses the value in the range [1, 0].

5.4 Classification: In this step we employ classification method to carry out the task of domain classification. It then classifies the domain namesto either legitimate or DGA generated, initially the attributes are captured by deep learning layers and later these features are passed to a fully connected layer. Each neuron in thefully connected layerhas connection to every other neuron in the next layer. The prediction loss of deep learning model is computed using binary cross entropy as follows:

$Loss\ (p, e) = -1/N \sum^{N}_{i=1} [e_i \log p_i + (1- e_i) \log (1-p_i)(5)$

Where p represents predicted probability and e is the vector of expected class label..

In the context of classifying the domain names, the loss of prediction for proposed model is evaluated using categorical cross entropy:

$$\text{Loss}(p, e) = -\sum_x p(x) \log(e(x)) \quad (6)$$

Where p indicates true probability distribution and q represents the predicted probability distribution.

6. Dataset Description

The proposed domain name detection model was evaluated on AmritaDGA dataset for discovering malwares/botnets from the DNS traffic [12]. AmritaDGA is a benchmark dataset publically available for research purpose [1].This database was used in DMD-2018 shared task and after the shared task this database has been used for benchmark purpose by various researchers for DGA detection [7]. Following, in this work, the AmritaDGA database was used for DGA domains detection. Each domain name in the dataset is labelled as legitimate or DGA. The dataset is further divided into training and testing respectively.

All deep learning models are trained using the training dataset. Further the dataset is comprised of training, validation and testing dataset. The training, validation and testing domain name samples are shown in the below Table1.

Table 1: Detailed information of the dataset

	Training	**Validation**	**Testing**
Legitimate	41956	10395	13180
DGA	8650	2257	2635
Total	50606	12652	15815

7. Performance Metrics

We have adopted performance measures to compare the accuracy of variousDGA classification models. Further the various metrics have been usedtodetermine the quality of DGA classification models. AUC, recall, precision, F1 score, and ROC performance evaluation metrics are used to compare the GRU with other deep learning classification techniques.

True Positive: It represents the number of domainsclassified aslegitimate and which is indicated with class 0.

True Negative: This representsthe number of domainsclassified asDGA generated and which is indicated with class 1.

False Positive: It represents the number of domains wrongly classified aslegitimate.

False Negative: It represents the number of domainswrongly classified asDGA generated.

Accuracy: It is defined as the total no of correct predictions made with respect of all the predictions drawn by the model.

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

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

Recall: Measures the completeness of correctly labelled features.

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

F1-score: Defines the harmonic mean between precision and recall measures.

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

Receiver operating characteristic measures the trade-off of the TPR to FPRwhere

$$\text{TPR} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

$$\text{FPR} = \frac{\text{FalsePositive}}{\text{FalsePositive} + \text{TrueNegative}}$$

The ROC curve established with recall and false positive rate. It also shows the capability of the binary classifiers. The ROCcurve also measures the competence of the classifier in differentiating the classes as either DGA and legitimate. The graph is plotted between the two metrics recall and false positive rate.

$$\text{AUC} = \int_0^1 \frac{TP}{TP+FN} \, d \, \frac{FP}{TN+FP}$$

The macro avgand weighted avgare used to average the results over the classes.The Macroavg computes the elements independently and finally it takes the average over all classes. In this paper, the weighted averaging considers as a significant performance indicator.

8. Results and Discussion

The deep learning techniques were implemented using Tensor Flow [36] and Keras [37]. Initially, various test experiments were run to identify various parameters for GRU model. In the proposed GRU model, the first layer is embedding, and it contains embedding length parameter. We run experiments with 32, 64, 128, and 256. The performance with 128 was good compared to others and when we increase 128 to 256, the performance remains same. Thus we decided to set the embedding length as 128. Each character of the domain will be transformed into 128 length vector. Next embedding layer follows GRU layer and again similar experiments were done and 128 units were set to

GRU layer. GRU layer follows the classification or output layer. Also, dropout was added in between the output layer and GRU layer.

The proposed deep learning models are evaluated for categorize the algorithmically generated domains as legitimate or DGA generated. The performance of GRU, LSTM, and RNN is shown in Table 2, 3 and 4 respectively. The performances of deep learning recurrent models were compared with classical approach such as bigram with logistic regression. All the three deep learning recurrent models performed better than the classical approach.
The results of the models are represented in the form of a PRC curve by differentiating two parameters, such as precision and recall. The ROC curve is represented with false positive rate and recall value. Deep learning model outperforms the N-gram derived features with huge size of domain names [38, 39].

The PRC curves for the Bigram baseline model, LSTM, RNN and GRU are presented in Fig4. The performance metrics of various methods are displayed in precision recall curve which shows that the gated recurrent unit results with higher precision and recall value compared with other approaches such as RNN and LSTM, the GRU model gives 93% of precision and 94% of recall value. The green line represented in the figure 4 indicates gated recurrent unit, red line indicates LSTM and blue line indicates the baseline model.

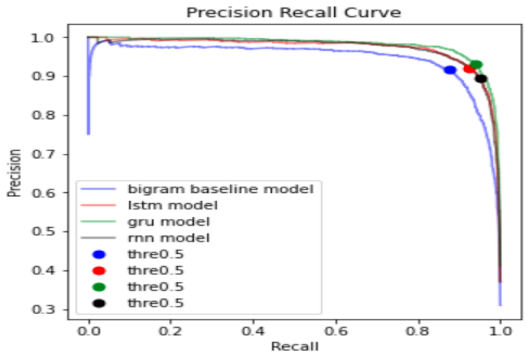

Fig 4: PRC curve of deep learning approaches for classifying the domains as either DGA or legitimate using LR, LSTM, RNN and GRU model.

The ROC curve is represented with false positive rate and recall value shown in Fig. 5. The gated recurrent unit gives the performance with an AUC of 0.996.

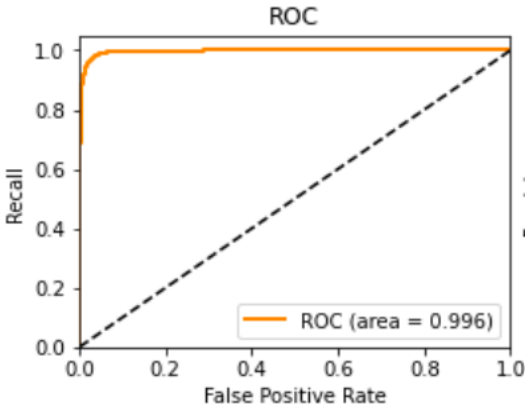

Fig 5: ROC curve

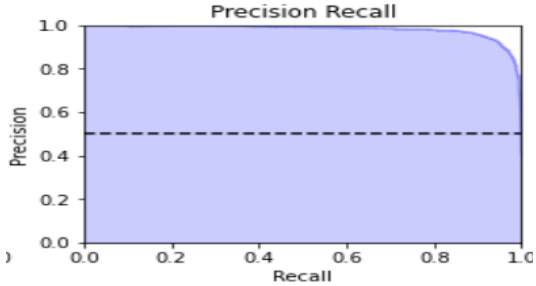

Fig 6: Precision Recall

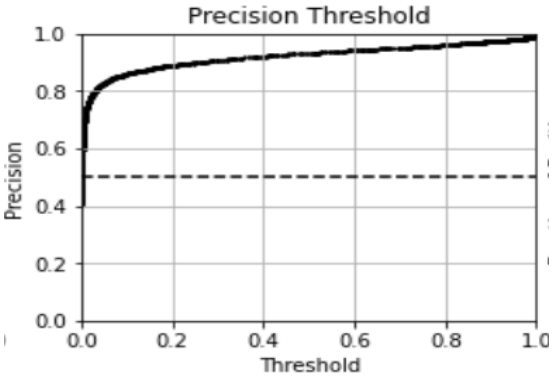

Fig 7: Precision Threshold

The confusion matrix for baseline model with logistic regression in Fig9 shows that 327 legitimate domains are misclassified as DGA generated domains and 207 DGA generated domains are misclassified as legitimate domains, 96.6% of domains are correctly classified and 33.8% of domains are misclassified in the baseline model.

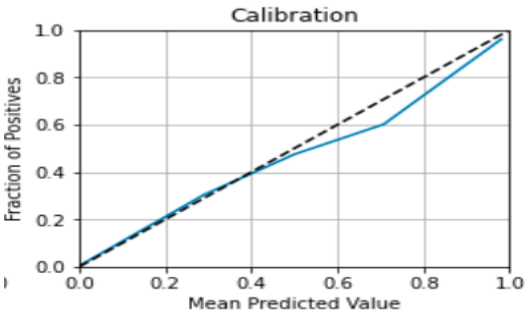

Fig 8:Calibration

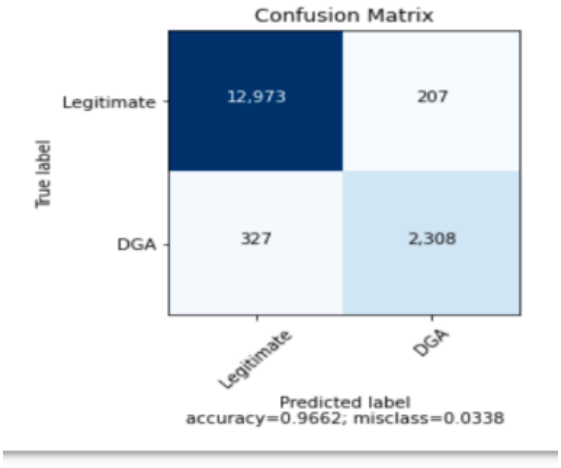

Fig9: Bigram LR representation with confusion matrix.

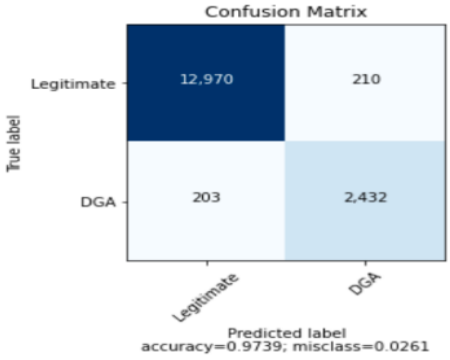

Fig 10: LSTM representation with confusion matrix

The confusion matrix for LSTM model in Fig10 shows that 203 legitimate domains are misclassified as DGA generated and 210 DGA generated domains are misclassified as legitimate domains, 97.3% of domains are correctly classified and 26.1% domains are misclassified in the LSTM model.

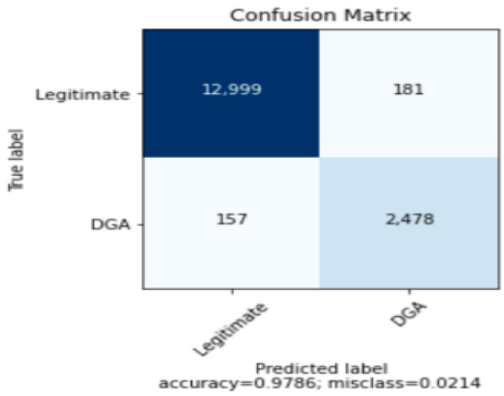

Fig 11: GRU representation with confusion matrix

The confusion matrix for GRU model in Fig11 shows that 157 legitimate domains are misclassified as DGA generated (botnet) and 181 DGA generated domains are misclassified as legitimate domains, 97.8% of domains are correctly classified and misclassification rate as 21.4% of domains are misclassified in the GRU model [39]. The confusion matrix for RNN model in Fig12 shows that 131 legitimate domains are misclassified domains are misclassified as legitimate domains, 97.3% of domains are correctly classified and 26.9% domains are misclassified in the RNN model.

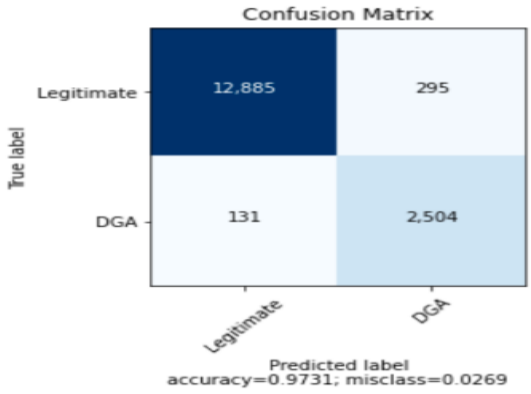

Fig 12: Confusion matrix for RNN

The performance evaluation of logistic regression model is represented with various performance metrics given in Table2, the model gives accuracy of 97%, precision of 92% and f1-score of 90%.

Table 2: Summary of test results of Bigram with LR for binary classification of domain names.

	Precision (%)	recall (%)	f1-score (%)	support
Legitimate	98	98	98	13180
DGA	92	92	90	2635
Accuracy of the model			97	15815
macro average	95	95	95	15815
weighted average	97	97	97	15815

Table 3: Test results of LSTM model for binary classification of domain names.

	precision (%)	recall (%)	f1-score (%)	support
Legitimate	99	98	98	13180

DGA	89	95	92	2635
Accuracy of the model			97	15815
macro average	94	96	95	15815
weighted average	97	97	97	15815

Similarly, the LSTM model gives accuracy of 97%, recall of 92%, macro avg of 95% and weighted avg of 97%. Further the RNN model results with 97% of accuracy, 87% of precision, 95% of recall and finally 92% of f1-score. The proposed model gated recurrent unit performance metrics is shown in table5, the model gives significantly improved accuracy of 98%, macro avg of 96% and weighted avg of 98%.

Table 4: Test results of RNN for binary classification of domain names.

	Precision (%)	recall (%)	f1-score (%)	support
Legitimate	98	98	98	13180
DGA	92	88	90	2635
Accuracy of the model			97	15815
macro average	95	93	94	15815
weighted average	97	97	97	15815

The Table 5 shows experimental results of deep learning approaches to categorize the domains as either legitimate or DGA generated. The proposed detection model gated recurrent unit significantly outperforms in comparison with other deep learning.
models like LSTM and RNN in all measurements, in which our model produces accuracy of 99%, F1-score of 98%, macro avg of 98.1% and weighted avg of 97.34% and provides much lower FPR and FNR. Whereas the LSTM and RNN models produces accuracy of 97%, recall of 94%, precision of 93%, macro avg of 95% and weighted avg of 97%. The baseline model for logistic regression produces accuracy of 97%, f1-score of 90%, precision of 92% and macro avg of 94% and weighted avg of 97%.

Table 5: Test results of GRU for binary classification of domain names.

	precision (%)	recall (%)	f1-score (%)	support
Legitimate	99	99	99	13180
DGA	93	94	94	2635
Accuracy of the model			98	15815
macro average	96	96	96	15815
weighted average	98	98	98	15815

The proposed approach will not make use of feature engineering approaches; however, the recent literature survey shows that the deep learning models are not potential in an adversarial network environment. Thus the performance evaluation of proposed model has to be studies in an adversarial environment and this is considered as important directions towards future work.

Table 6: The performance measures of deep learning models for binary classifiers

	Precision (%)				Recall (%)				F1-score (%)				Support			
	LR	RNN	LSTM	GRU	LR	RNN	LSTM	GRU	LR	RNN	LSTM	GRU	LR	RNN	LSTM	GRU

				DGA	Legitimate
0	92	98	99		
1	89	99	98		
2	92	98	99		
3	93	99	98		
4	88	98	98		
5	95	98	98		
6	92	98	98		
7	94	99	98		
8	93	98	97		
9	92	98	97		
10	93	98	97		
11	94	99	98		
12	2635	13180	15815		
13	2635	13180	15815		
14	2635	13180	15815		
15	2635	13180	15815		

Conclusion

This paper introduces a deep learning-based methodology for accurately classifying algorithmically generated domains as either DGA or legitimate, without the need for manual feature selection. By training at the character level and employing a bigram approach for feature extraction, raw domain names are efficiently converted into numerical values, enhancing the classification process. Our research reveals that traditional models such as logistic regression lack efficiency in classifying DGA-generated domains. In contrast, deep learning models exhibit superior performance by autonomously extracting optimal features directly from raw domain names. Leveraging Keras character embedding, we explore various deep learning architectures including RNN, LSTM, and GRU, highlighting their effectiveness in learning and representing domain features. Notably, our proposed approach eliminates the requirement for feature engineering, streamlining the classification process and demonstrating the efficacy of deep learning methodologies in DGA detection. Among the deep neural networks studied, the GRU model emerges as the most accurate, achieving an impressive accuracy of 99%. Experimental results further confirm the superiority of GRU, with a recall rate of 94%, F1-score of 98%, macro average of 98.1%, and weighted average of 97.34%, surpassing both LSTM and RNN models. Our findings underscore the significance of deep learning in enhancing DGA detection capabilities, contributing to the advancement of cyber security strategies by providing a reliable and efficient means of identifying malicious domains.

References

[1] https://nlp.amrita.edu/DMD2018/

[2] Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q. V., Padannayil, S. K., &Simran, K. (2020). A Visualized Botnet Detection System based Deep Learning for the Internet of Things Networks of Smart Cities. IEEE Transactions on Industry Applications.

[3] Vinayakumar R., Soman K.P., Poornachandran P., Alazab M., Jolfaei A. (2019) DBD: Deep Learning DGA-Based Botnet Detection. In: Alazab M., Tang M. (eds) Deep Learning Applications for Cyber Security. Advanced Sciences and Technologies for Security Applications.Springer.

[4] S. Akarsh, S. Sriram, P. Poornachandran, V. K. Menon and K. P. Soman," Deep Learning Framework for Domain Generation Algorithms Prediction Using Long Short-term Memory," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 666-671.

[5] MamounAlazab and RodericBroadhurst.Spam and criminal activity. Trends and Issuesin Crime and Criminal Justice (Australian Institute of Criminology), (52), 2016.

[6] Brandon Amos, Hamilton Turner, and Jules White.Applying machine learning classifiers to dynamic android malware detection at scale.In 2013 9th international wireless communications and mobile computing conference (IWCMC), pages 1666{1671.IEEE, 2013.

[7] Manuel Egele, TheodoorScholte, EnginKirda, and Christopher Kruegel.A survey on automated dynamic malware-analysis techniques and tools. ACM computing surveys (CSUR), 44(2):6, 2012

[8] Anna L Buczak and ErhanGuven. A survey of data mining and machine learning methodsfor cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18 (2):1153{1176, 2015

[9] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in Proceedings of the Int'l Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 192–211, London, UK, July 2014.

[10] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: detecting the rise of DGA-based malware," in P21st USENIX Security Symposium (USENIX Security 12), pp. 491–506, 2012.
[11] S. Yadav, A. K. K. Reddy, A. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proc. 10th ACM SIGCOMM conference on Internet measurement, pp. 48–61, ACM, 2010.
[12] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," Networking, IEEE/ACM Transactions on, vol. 20, no. 5, pp. 1663–1677, 2012.
[13] J, Woodbridge, H.S. Anderson,A.Ahuja, and D.Grant,(2016). "Pre-dicting domain generation algorithms with long short-term memory networks."[Online]. Available:
[14] Anderson, H. S., Woodbridge, J., &Filar, B. (2016, October). Deep- DGA: Adversarially-tuned domain generation and detection. In Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (pp. 1321).ACM.
[15] Curtin, R. R., Gardner, A. B., Grzonkowski, S., Kleymenov, A., &Mosquera, A. (2018). Detecting DGA domains with recurrent neural networks and side information.arXiv preprint arXiv:1810.02023.
[16] Feng, Z., Shuo, C., &Xiaochuan, W. (2017, December). Classification for DGA-Based Malicious Domain Names with Deep Learning Architectures. In 2017 Second International Conference on Applied Mathematics and information technology (p. 5).
[17] Mac, H., Tran, D., Tong, V., Nguyen, L. G., & Tran, H. A. (2017, December). DGA Botnet Detection Using Supervised Learning Methods.In Proceedings of the Eighth International Symposium on Information and Communication Technology (pp. 211-218).ACM.
[18] Rangaraju, S. (2023, December 1). AI sentry: reinventing cybersecurity through intelligent threat detection. Eph - International Journal of Science and Engineering, *9*(3), 30–35. https://doi.org/10.53555/ephijse.v9i3.211.
[19] Y. Qiao, B. Zhang, W. Zhang, A. K. Sangaiah, and H. Wu, "DGA domain name classification method based on long short-term memory with attention mechanism," Applied Sciences, vol. 9, no. 20, 2019.
[20] Vinayakumar, R., Soman, K. P., Poornachandran, P., & Sachin Kumar, S. (2018). Evaluating deep learning approaches to characterize and classify the DGAs at scale. Journal of Intelligent & Fuzzy Systems, 34(3), 12651276.
[21] Vinayakumar, R., Poornachandran, P., &Soman, K. P. (2018). Scalable Framework for Cyber Threat Situational Awareness Based on Domain Name Systems Data Analysis. In Big Data in Engineering Applications (pp. 113-142). Springer, Singapore.
[22] Vinayakumar, R., Soman, K. P., &Poornachandran, P. (2018). Detecting malicious domain names using deep learning approaches at scale. Journal of Intelligent & Fuzzy Systems, 34(3), 1355-1367.
[23] Lison, P., &Mavroeidis, V. (2017). Automatic Detection of Malware Generated Domains with Recurrent Neural Models. arXiv preprint arXiv:1709.07102.
[24] Vinayakumar, R., Soman, K. P., Poornachandran, P., Mohan, V. S., & Kumar, A. D. (2019). ScaleNet: Scalable and Hybrid Framework for Cyber Threat Situational Awareness Based on DNS, URL, and Email Data Analysis. Journal of Cyber Security and Mobility, 8(2), 189-240.
[25] Yu, B., Gray, D. L., Pan, J., De Cock, M., &Nascimento, A. C. (2017, November). Inline dga detection with deep networks.In Data Mining Workshops (ICDMW), 2017 IEEE International Conference on (pp. 683692).IEEE.
[26] Yu, B., Pan, J., Hu, J., Nascimento, A., & De Cock, M. (2018). Character Level Based Detection of DGA Domain Names.
[27] Guzman Erick, & Fatehi Navid. (2023, December 19). Safeguarding stability: strategies for addressing dynamic system variations in power grid cybersecurity eph - International Journal of Science and Engineering (ISSN: 2454 - 2016); *9*(3): 42–52. https://doi.org/10.53555/ephijse.v9i3.215.

[28] Pierre Lison and VasileiosMavroeidis, "Automatic Detection of Malware-Generated Domains with Recurrent Neural Models" September 2017.
[29] Ren F., Jiang Z., Wang X and Jiang Liu " A DGA domain names detection modelling method based on integrating an attention mechanism and deep neural network" Cybersecurity3,4(2020).
[30] Samuel Schuppen, DominikTeubert and Patrick Herrmann "FANCI: Feature-based Automated NXDomain Classification and Intelligence" August 15-17,2018 27th USENIX Security Symposium.
[31] T.Chin, K.Xiong, C.Hu, and Y. Li, "A machine learning framework for studying domain generation algorithm(DGA)-based malware, " in Proc. SecureComm,2018,pp.433-448.
[32] Vinayakumar R, Soman KP, PrabaharanPoornachandran, MamounAlazab, and Sabu M. Thampi, AmritaDGA: A Comprehensive Data set for Domain Generation Algorithms (DGAs). In Big Data Recommender Systems: Recent Trends and Advances, Institution of Engineering and Technology.
[33] Yu, B., Pan, J., Hu, J., Nascimento, A., & De Cock, M. (2018). Character Level Based Detection of DGA Domain Names.
[34] Yu, B., Gray, D. L., Pan, J., De Cock, M., &Nascimento, A. C. (2017, November). Inline dga detection with deep networks.In Data Mining Workshops (ICDMW), 2017 IEEE International Conference on (pp. 683692).IEEE.
[35] Zago, M., Prez, M. G., &Prez, G. M. (2019). Scalable detection of botnets based on DGA. Soft Computing, 1-21.
[36] Vinayakumar R, Soman KP, Poornachandran P (2017) Applying convolution neural network for network intrusion detection. In: 2017 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 1222–1228.

[37] Vinayakumar R, Soman KP, Poornachandran P (2017) Applying deep learning approaches for network traffic prediction. In 2017 International conference on advances in computing, communications, and informatics (ICACCI). IEEE, pp 2353–23
[38] Tensorflow, https://www.tensorflow.org/
[39] Keras, http://keras.io/