Efficient of EmpowGen using Deep Learning

3rd Dr. Sivaprakash P

Mentor IT

Rathinam College of Arts and Science

Coimbatore, India

p.sivaprakash@inurture.co.in

2nd Er. M Saravana kumar   
Assitant Professor Rathinam College of Arts and Science Coimbatore,India

Saravankumar.m@inurture.co.in

1st Mr. Dinesh Kumar C  
PG Scholar

Rathinam College of Arts and Science

Coimbatore, Indiadineshkarthi196@gmail.com

*Abstract*—Domain generation algorithms (DGAs) are used by attackers to generate a large number of pseudo-random domain names to connect to malicious command and control servers (C&Cs). These domain names are used to evade domain-based security detection and mitigation controls. Reverse engineering of malware samples to discover the DGA algorithm and seed to generate the list of domains is one of the techniques used to detect DGA domains. These domains are subsequently preregistered and sinkholed, or published on security device blacklists to mitigate malicious activity. This technique is time-consuming and can be easily circumvented by attackers and malware authors. Statistical analysis is also used to identify DGA domains over a time window, however many of these techniques need contextual information which is not easily or feasibly obtained. Existing studies have also demonstrated the use of traditional machine learning techniques to detect DGA domains. Our goal was to detect DGA domains on a per domain basis using the domain name only, with no additional information. T DGA classifier that leverages a Long short-term memory-based architecture for the detection of DGA domains without the need for contextual information or manually created features. We compared the performance of different LSTM based architectures by evaluating them against a dataset of 2 million plus domain names. The results indicated little difference in performance metrics among the LSTM architectures.

Keywords — DGA, RNN, LSTM, CNN, C&Cs.

# Introduction

The Internet has become an integral part of our lives, but it is also a source of malware. Malwares gain access through various methods, such as DNS translation, which increases traffic over the DNS and provides access to attackers. Botmasters use botnets to steal information and data from victims, and ransomware attacks are distributed by DoS attacks. To detect malicious access, DGA's blacklisting method is used, but machine learning methods are vulnerable to imbalanced DGAs. To handle imbalanced DGAs, Cost-Sensitive long short-term memory (CS-LSTM) is proposed as a novel method for DGA categorization. DNS is a collective form of database to store domain names and IP addresses. Domain fluxing bots use the Domain generation algorithm to randomly generate thousands of domain names, which are registered by the botnet operator. Bot-nets are a major threat to cyber security applications, as they create software vulnerabilities from backdoor access .

# EXISTING SYSTEM

Domain generation algorithms (DGAs) are typically used by malware authors to generate a large number of domain names that can be used for various malicious purposes, such as command and control (C2) servers or phishing attacks. There are several existing systems that use different techniques to generate these domains. Here are some examples:

Markov Model-Based DGA: This type of DGA system uses a Markov model to generate domain names. Markov models are statistical models that can predict the probability of a sequence of events based on the preceding events. This approach is used to generate random-looking domain names that can evade detection by security tools.

Wordlist-Based DGA: This DGA system uses a predefined list of words to generate domain names. These words are usually related to the malware or the intended target of the attack. The DGA system randomly combines these words to generate domain names.

Time-Based DGA: This type of DGA system uses the current date and time to generate domain names. The system uses a mathematical algorithm to generate a sequence of characters that changes over time. This approach allows the malware to generate new domains on a regular basis, making it difficult for security tools to block them.

Hybrid DGA: This DGA system combines multiple techniques to generate domain names. For example, it may use a Markov model to generate the first part of the domain name and a wordlist to generate the second part. This approach can make the generated domain names even more random-looking and difficult to detect.

Artificial Intelligence-Based DGA: This DGA system uses machine learning techniques, such as deep neural networks, to generate domain names. These models are trained on a large dataset of domain names and can learn to generate new domain names that resemble the patterns in the training data. This approach can create highly sophisticated and realistic domain names that are difficult to distinguish from legitimate ones.

# CHALLENGES IN EXISTING SYSTEM

There are several challenges associated with existing DGA systems:

Detection: One of the main challenges in DGA systems is detecting the generated domains. Malware authors often use techniques to evade detection by security tools, such as generating domain names that resemble legitimate ones or using encryption to hide the communication.

Evolution: Another challenge is the evolution of DGA systems. Malware authors are constantly adapting and evolving their DGAs to evade detection by security tools. This includes changing the algorithms used to generate domain names, the frequency of domain name generation, and the infrastructure used to host the domains.

Diversity: Existing DGA systems often generate a large number of domains, which makes it difficult to block all of them. Moreover, these domains can be hosted on different servers and IP addresses, making it even harder to detect and block them. Legitimate Domains: DGA systems can generate domain names that are similar to legitimate domains, which can result in blocking legitimate traffic. This can cause significant disruptions for users and businesses. False Positives: Security tools may generate false positives by incorrectly identifying legitimate domains as generated by a DGA system. This can result in unnecessary alerts and disruptions. Resource Intensive: Detecting and blocking DGA-generated domains requires significant computational resources and can be expensive for organizations. Overall, the challenges associated with existing DGA systems require continuous development and improvement of detection and mitigation strategies to stay ahead of evolving threats.

# LITERATURE SURVEY

Kührer, m., rossow, c. And holz [1] blacklisting are used in system protection from malwares. In this an emprical approach on a publically available data-set with 15 malware and 4 blacklist content listed data is analyzed. The main aim is to classify blacklist to understand the nature of the listed domains according to corresponding ip addresses. As per experimentation parked domains are identified, graphical approach is determined to list the sinkholes. Antonakakis, manos [2] proposed a technique to identification of domain names without reverse engineering process, work mainly focuses on the nx-domain names. Clustering approach is taken and domains are classified into clusters based on their characters tics features. This detected method is named as pleiades, which is comprised to monitor the traffic flow along the recursive server. Yadav, sandeep [3] explained the dns traffic along the network, analysis of patterns in which domain flux is happening is studied. The distribution pattern of a particular numerical character is taken and mapped to extract the information using bigram techniques. Distance is calculated using kl and edit distance method, the output is mapped to different ipv models to consider the data traffic along the internet service provider. Antonakakis, manos [4] a novel approach called notos a dns dynamic reputation system, these approach helps in calculate score by analyzing the transverse edge of the networks. All the information is extracted from different resoueces of dns names, based on the information clustering is done to identify the behaviour of malicious and legitimated. This method comes up with 96.8\% of true positve rate and with loss of 0.38\% for identifying dns traffics. Bilge, leyla ,[5] a new method exposure is used analysis dns domains at large scale, in this approach 15 fetures are extracted and characterized in basis of dns queries and serves a command to c \& c server for specific threat analysis. Yu, bin, jie pan, jiaming hu, [7] proposed various deep learning algorithm for dga detection. Nearly about 2m domain names is used for evaluation to classify domain names either benign or malicious. Five deep learning architecture such as rnn architecture in which endgame and cmu model used, in cnn architecture nyu and invincea model used, in hybrid rnn/cnn architecture mit model is used. In which all network performs equally well of 97-98\% of accuracy with false positive rate of 0.001. Using hang craft feature random forest classifier achieve 91.57\% with false positive rate of 0.001. Zeng, f., change, s,[8] focused on CNN architecture such as vgg net, alex net, squeeze net, inception, res net used to classify dga names. Transfer learning approach is used to extract features from raw input, using various network about 99.86\% with false positive rate of 0.011 is achieved. Yu, b., gray, d.l., pan, j., de cock, [9] supervised learning approach used for dga detection. Various filtering steps are used to obtain real time data samples from dns servers. A comparison study is done between lstm and cnn network in which lstm performs with high performance rate. Vinayakumar, r., prabaharan poornachandran [18] performed several analyses works on large scale to give a detail of cyber activity on dns server and how to detect malware in early stage. For analysis deep learning methods used by analyzing the dns information at tier-1 internet service provider. It is stated that the developed framework can detect nearly 2million malicious activities in real time and can give real time warning. Mac, h., tran, d., tong, v., nguyen, l.g,[10] investigated various handcrafted features such as hidden markov model, decision tree, svm (support vector machine) and c4.5 used. In feature-based technique entropy and length are consider with equivalent dictionary score followed by n-gram normality score for the domain names. Deep learning network such as cnn-lstm, bi-directional lstm used, from which svm and bidirectional lstm achieve higher classification rate on both binary and multi-class dataset. Woodbridge, j., anderson, h.s., ahuja, [13] focused lstm architecture to classify domain names, this paper helps to understand the data distribution. Using data distribution, a characteristic of dga family is analyzed by taking the sequence and length features for each domain names separately. Using lstm model, the sequence characteristic of the domain names is absorbed and it helps in classifies the domain name according to corresponding classes. Vinayakumar, r., soman, k.p. and poornachandran p [11] collected dns logs from server, these dns log details are used to classify dga families. Author mainly focuses on deep learning architecture and data collection for cyber security application. As per analysis lstm approach gives a higher detection rate on the collected data-set.in [14] this paper author used nearly one million dga domain names from various resources such as open dns, alexa data-set, osnit data which consists of 17 dga malware families. Author considers the effectiveness these data on various in various deep learning architectures at a large-scale approach curtin, ryan r., andrew b. Gardner [15] analyzed and give a solution to measure the complexity of distribution of dga families using a technique called smash-word score. These smash-word score help in identify the resemblance of dga families based on english words, considering the smash- word count machine learning model and deep learning models are built. [17] proposed various feature extraction methods used by various research work using machine learning algorithms. This paper state the state of art of all proposed work and how feature extraction methods contributes in binary and multi-class problems. Mohan, vysakh s., r. Vinayakumar [16] proposed a new approach s.p.o.o.f net, combination of a convolution neural network (cnn) and lstm which is an embedding concept from natural language processing (nlp) is been embedded into cybersecurity use cases.

# V. Methods and proposed

## **Dataset Description**

## The data used to train the classifier is taken from the CSE-CIC-IDS2018 dataset provided by the Canadian Institute for Cybersecurity. It was created by capturing all network traffic during ten days of operation inside a controlled network environment on AWS where realistic background traffic and different attack scenarios were conducted. As a result the dataset contains both benign network traffic as well as captures of the most common network attacks. The dataset is comprised of the raw network captures in pcap format as well as csv files created by using CICFlowMeter-V3 containing 80 statistical features of the individual network flows combined with their corresponding labels. A network flow is defined as an aggregation of interrelated network packets identified by the following properties: Source IP Destination IP Source port Destination port Protocol The dataset contains approximately 16 million individual network flows and covers the following attack scenarios: Brute Force DoS, DDos Heartbleed, Web Attack, Infiltration, Botnet. In this work 3 different methods are obtained for experimental analysis

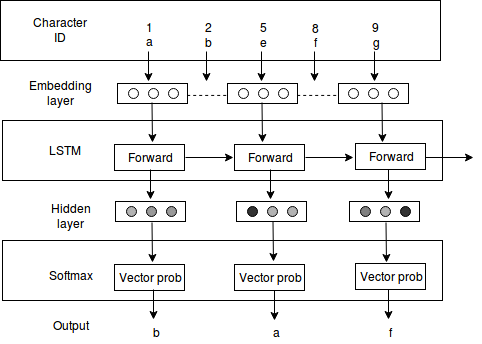
## **Proposed Method**

#### **Method 1:** Standard Long- short term memory

#### **Method 2:** LSTM + Glove embedding

## **Character level encoding**

Proposed architecture consists of input layer followed by embedding layer where character level features are extracted and passed. The output of the embedding layer is possessed with LSTM layer with softmax activation function followed by output layer.



1. Character Level Encoding

##### Domain name representation is typically referred as encoding. The encoding of the domain name consists of two steps. The raw domain names are pr-processed into characters in the first step. In pre-processing , the top-level domains is removed and all characters converted into lower case. Second step involves creating vocabulary with only initial step of using the training data. The parameter value of vocabulary size is based on the symmetry between each class training vectors and number to be learned for the given task. Here only the characters that meet the minimum frequency is selected to limit the size of the vocabulary. Followed by the initial step pf creating vocabulary, each character is assigned to a unique ID and each unique ID is a vector denoting the size if the vocabulary D. Default key value is assigned to the unknown characters, and using look up-table operation, the unique characters are transformed into feature vectors.

## **Glove Embedding**

##### Pre-trained embedding of words is a key element in NLP's deep learning. In mainstream deep learning, the pioneer of word embedding is the famous word2vec. Glove is a pre-trained embedding technique ,In fact, Glove is a much more principled approach to embedding words, which generally offers profound insights into embedding words. The main aim of glove is

### Meaning of a words is captured and mapped into a vector space

### Glove embedding helps in reduction of loss by updating the weights.

The underlying principle behind GloVe can be stated as follows: the co-occurrence ratios between two words in a context are strongly connected to meaning.

This sounds difficult but the idea is really simple. Take the words “ice” and “steam”, for instance. Ice and steam differ in their state but are the same in that they are both forms of water. Therefore, we would expect words related to water (like “water” and “wet”) to appear equally in the context of “ice” and “steam”. In contrast, words like “cold” and “solid” would probably appear near “ice” but would not appear near “steam”.

In GloVe, principled based approach is performed. The first step is to build a co-occurrence matrix. GloVe also takes local context into account by computing the co-occurrence matrix using a fixed window size (words are deemed to co-occur when they appear together within a fixed window). For instance, the sentence

The probabilities shown here are basically just counts of how often the word k appears when the words “ice” and “steam” are in the context, where k refers to the words “solid”, “gas”, “water”, and “fashion”. As you can see, words that are related to the nature of “ice” and “steam” (“solid” and “gas” respectively) occur far more often with their corresponding words that the non-corresponding word. In contrast, words like “water” and “fashion” which are not particularly related to either have a probability ratio near 1. Note that the probability ratios can be computed easily using the co-occurrence matrix. The detail overview of glove embedding.

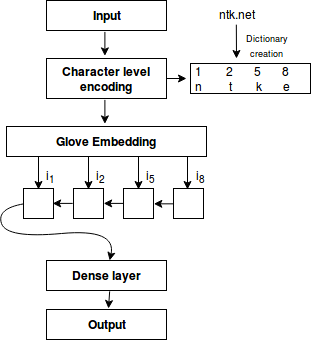


Fig. 2. Glove embedding architecture

# Result And Discusion

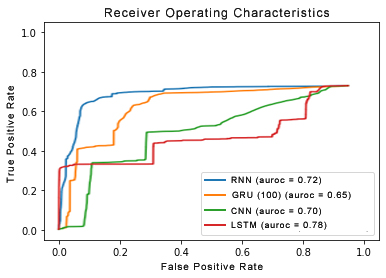
## **Experiment 1**

Deep layer algorithm such as RNN, LSTM, CNN, GRU is used. To evaluate the performance of all algorithms constant network parameter is set for all algorithm such as learning rate as 0.1, batch size as 32, and dropoutas 0.2. In recurrent algorithm RNN, LSTM, methods are adopted, in which softmax is used as a activation function with embedding vector length of 128 and maximum feature of 40, in LSTM and GRU memory cell size is set as 128.

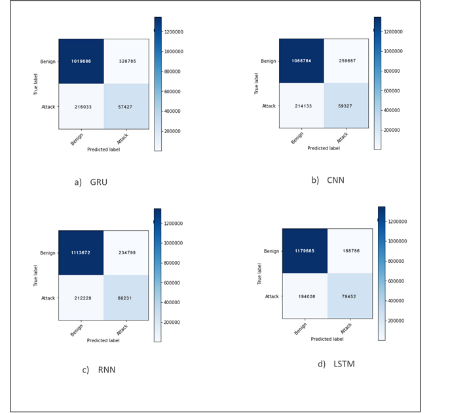
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithms | Accuracy(%) | Precision | Recall | F1-Score |
| GRU | 0.65 | 0.62 | 0.67 | 0.66 |
| CNN | 0.70 | 0.68 | 0.69 | 0.61 |
| RNN | 0.72 | 0.67 | 0.74 | 0.69 |
| LSTM | 0.78 | 0.71 | 0.79 | 0.76 |

Table 4.1: Summary of test results on different deep learning architecture

In convolution architecture 1D convolution is adopted for analysis, with 64 filter of length3. .By evaluating the model performance of each moreover all network are differs with high amount of variation of accuracy from which GRU not performed well compared to other algorithms and LSTM outperformed with 78.2\% accuracy as show in table 1 as per experimentation1.



To get a detailed overview of the correctly predicted and non-predicted data confusion matrix is drawn as show in figure 4.2



## **Experiment 2**

On account of LSTM performance in trial 2 experimentation hyper-parameter tuning is applied with various tuning parameters in LSTM architecture. In trail 2 embed-ding vector length is been kept constant of 50, and various LSTM memory size is used such as 256, 128, 64 and 32 to identify the actual parameter for analysis. It is clearly state that when number of input neuron decreases accuracy increases as per result input neuron with 64 and 32 is performed with good result of 84 and 87%

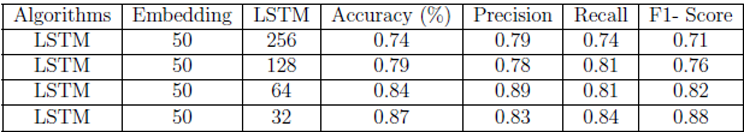
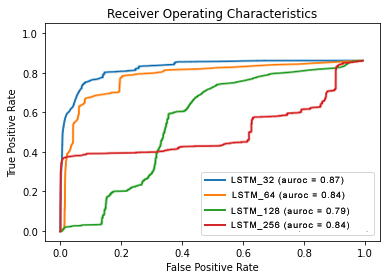
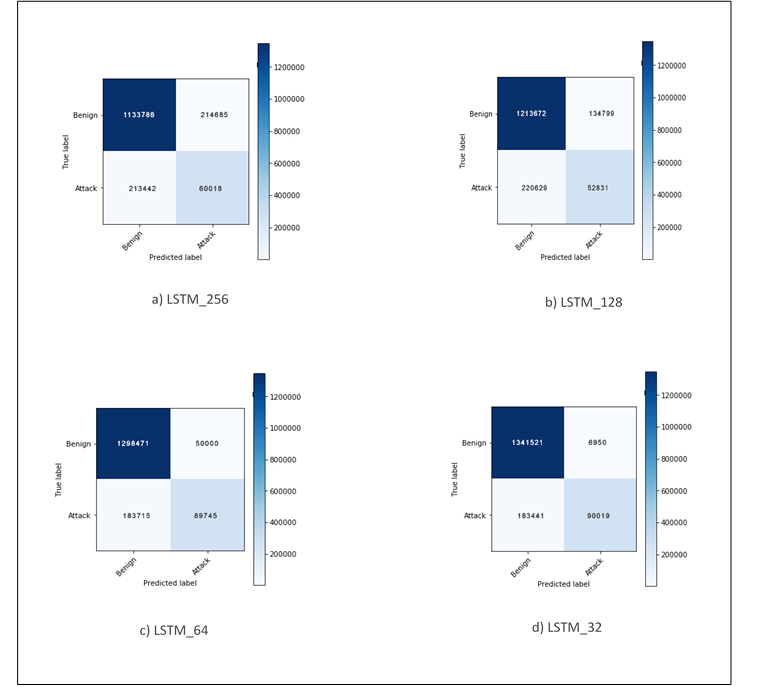


Table 4.2: Summary of test results on LSTM with different LSTM Size



To get a detailed overview of the correctly predicted and non-predicted data confusion matrix is drawn as show in figure



LSTM of size 64 and 32 gives a better result compared to previous experimentation only benign classes are predicted almost of 93 \%. Attack classes are not predicted correctly with high amount of misclassification which in accuracy declination. While analyzing the data it clearly states that there are more amount of trainable parameters are there compared with attack class which result in miss classification

And decrease in accuracy. To overcome this misclassification and for insufficient data glove embedding technique is applied to increase the weights of the attack class. Glove embedding is performed for LSTM of input size 64 and 32.

## **Experiment 3**

On account of LSTM performance in trial 3 experimentation LSTM with glove tuning is applied with various tuning parameters in LSTM architecture various LSTM memory size is used such as 256, 128, 64 and 32 similar to experiment 2 to see how glove embedding performs on different memory size. It is clearly state that when number of input neuron decreases accuracy increases as per result input neuron with 64 and 32 is performed with good result of 93 and 97\% . Compared to experimentation 2 all LSTM size of 256, 128, 64 ,32 performs well .

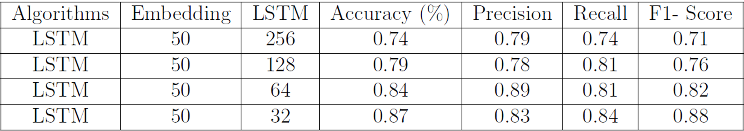
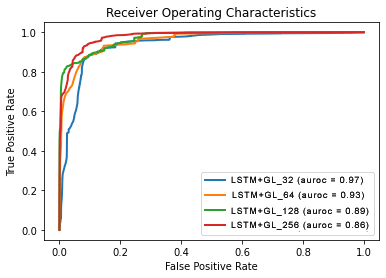
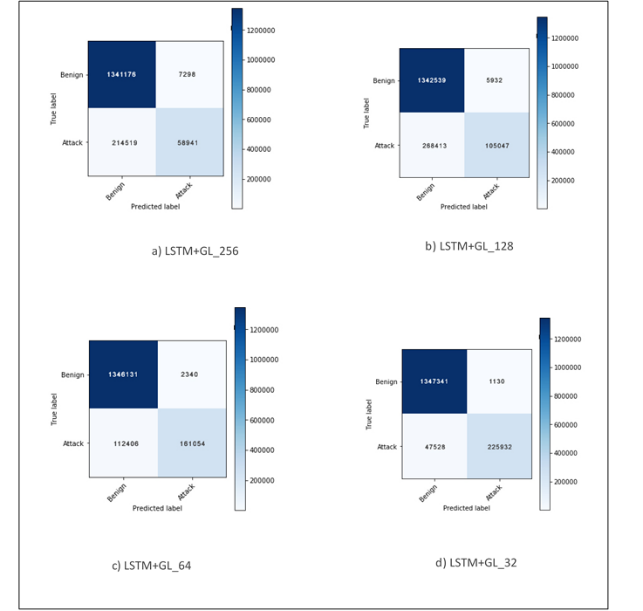


Table 4.3: Summary of test results on LSTM with different LSTM Size 28





# CONCLUSION

This article provided an approach to classifying domains produced by DGA using LSTM networks. LSTMs are advantageous compared to other methods because they use raw domain names as their input and there is no need to manually generate characteristics that are hard to keep and in an adversarial machine learning environment. In addition, DGA dataset posses with class imbalance, to overcome this imbalance of data a new approach Glove-LSTM is applied on DGA Family categorization.

Cost-Sensitive helps in understand the importance of each class and classify the classes to a particular class, Glove embedding method is used to increase the performance of GL-LSTM.

The proposed method performs well and obtained better result in multiclass DGA dataset. Further the proposed method can be used to handle imbalanced dataset.

# References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
8. Zeng, Feng, Shuo Chang, and Xiaochuan Wan. \emph{Classification for DGA-based malicious domain names with deep learning architectures.} In 2017 Second International Conference on Applied Mathematics and information technology, p. 5. 2017.
9. Yu, Bin, Daniel L. Gray, Jie Pan, Martine De Cock, and Anderson CA Nascimento. \emph{Inline DGA detection with deep networks.} In 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 683-692. IEEE, 2017.
10. Mac, H., Tran, D., Tong, V., Nguyen, L.G. and Tran, H.A., 2017, December. \emph{DGA botnet detection using supervised learning methods.} In Proceedings of the Eighth International Symposium on Information and Communication Technology (pp. 211-218). ACM.
11. Vinayakumar, R., Soman, K.P., Poornachandran, P. and Sachin Kumar, S., 2018. Evaluating deep learning approaches to characterize and classify the DGAs at scale. Journal of Intelligent & Fuzzy Systems, 34(3), pp.1265-1276.
12. Lison, Pierre, and Vasileios Mavroeidis. \emph{Automatic detection of malware-generated domains with recurrent neural models.} arXiv preprint arXiv:1709.07102 (2017).
13. Woodbridge, Jonathan, Hyrum S. Anderson, Anjum Ahuja, and Daniel Grant. \emph{Predicting domain generation algorithms with long short-term memory networks.} arXiv preprint arXiv:1611.00791 (2016).
14. Vinayakumar, R., Soman, K.P. and Poornachandran, P., 2018. \emph{Detecting malicious domain names using deep learning approaches at scale.} Journal of Intelligent & Fuzzy Systems, 34(3), pp.1355-1367.
15. Curtin, Ryan R., Andrew B. Gardner, Slawomir Grzonkowski, Alexey Kleymenov, and Alejandro Mosquera. \emph{Detecting DGA domains with recurrent neural networks and side information.} arXiv preprint arXiv:1810.02023 (2018).
16. Mohan, Vysakh S., R. Vinayakumar, K. P. Soman, and Prabaharan Poornachandran. \emph{SPOOF net: syntactic patterns for identification of ominous online factors.} In 2018 IEEE Security and Privacy Workshops (SPW), pp. 258-263. IEEE, 2018.
17. Kwon, Jonghoon, Jehyun Lee, Heejo Lee, and Adrian Perrig. \emph{PsyBoG: A scalable botnet detection method for large-scale DNS traffic.} Computer Networks 97 (2016): 48-73.
18. Vinayakumar, R., Prabaharan Poornachandran, and K. P. Soman. \emph{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, 2018.
19. Sun, Y., Kamel, M.S., Wong, A.K. and Wang, Y., 2007.\emph{ Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition}, 40(12), pp.3358-3378.
20. Chollet, et al. , Keras (2015).
21. Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. \emph{Tensorflow: A system for large-scale machine learning.} In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265-283. 2016.
22. ntonakakis, Manos, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. From throw-away traffic to bots: detecting the rise of DGA-based malware. In Presented as part of the 21st USENIX Security Symposium (USENIX Security 12), pp. 491-506. 2012.
23. Tran, D., Mac, H., Tong, V., Tran, H.A. and Nguyen, L.G., 2018. \emph{A LSTM based framework for handling multiclass imbalance in DGA botnet detection.} Neurocomputing, 275, pp.2401-2413.
24. Geffner, J., 2013. \emph{End-to-end analysis of a domain generating algorithm malware family.} Black Hat USA, 2013.
25. Choudhary, Chhaya, Raaghavi Sivaguru, Mayana Pereira, Bin Yu, Anderson C. Nascimento, and Martine De Cock. \emph{Algorithmically generated domain detection and malware family classification.} In International Symposium on Security in Computing and Communication, pp. 640-655. Springer, Singapore, 2018.
26. Rajalakshmi, R., S. Ramraj, and R. Ramesh Kannan. \emph{Transfer Learning Approach for Identification of Malicious Domain Names.}In International Symposium on Security in Computing and Communication, pp. 656-666. Springer, Singapore, 2018.
27. Bharathi, B., and J. Bhuvana. \emp{Domain Name Detection and Classification Using Deep Neural Networks.}In International Symposium on Security in Computing and Communication, pp. 678-686. Springer, Singapore, 2018.