International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019

LSTM Network Based Malicious Domain Name Detection

Gurpreet Singh Josan, Jagroop Kaur

Abstract: ***Detecting malicious domain names attract lot of research in recent years. Researchers tried various text based, network traffic based and combination of these methods to detect malicious names. In this paper, we analyze the possibility of detection malicious names using deep neural network based models. Bidirectional LSTM network has been developed and trained on the dataset. Two tasks were experimented. First task was to identify malicious domain name and second task was to identify the class of domain name. Proposed method is able to perform well on task 1 producing 98.9% accuracy whereas on task 2 it is able to achieve accuracy of 69.7% only.***

Keywords : Botnet, Deep Neural Network, Malicious domain name classification.

I. INTRODUCTION

Among the numerous ways of security threats on internet, malicious domain name is a very basic tool in the hands of attackers. Algorithms have been developed to generate thousands of domain names on just a click of mouse. These domain names can be used by spammers, botnets, and other malwares posing security threats to the network. Command and control servers are generally exploited to force the host to perform variety of malicious task like spamming, phishing, and DDOS etc. Identification of malicious domain names is crucial to plug the security threats.

Automatic malicious domain name identification has numerous applications like stopping the misuse of benign domains, botnet profiling, identification of attack beforehand etc. Numerous approaches have been tried with varied degree of success in the past. Some uses network traffic other study botnet behavior, still other uses machine learning algorithms for classifying domain name as benign or malicious. Experiments based on active and passive analysis of domain names has been done by researchers. Active analysis includes DNS detection and analyses of websites content whereas passive analysis includes rule-based approach, machine learning approaches and graph-based approaches. Advancements in deep neural networks open new dimensions for the researchers. A few attempts have been found in literature with considerable success [15], [16]. We believe that human generated domain names has inherent character sequences which algorithmic or botnet generated domain names lacks. Deep neural network can capture the features of character sequences and may be able to differentiate between benign and malicious domains. Further, the sequence to be generated also depends upon the algorithms generating them. Thus if we train neural network, it will learn the kind of sequences generated by different botnets and may be able to map the malicious domain name to its corresponding botnet family. This paper describes LSTM network based approach for detecting malicious domain name and mapping it to its botnet family. This paper makes following contribution:

Proposed a LSTM based bidirectional neural network model and fine tune its hyper-parameters

- for classifying domain name as benign or malicious.
- for mapping domain name to its botnet family.

The remainder of this paper is organized as follows. Next sections discuss previous work in field followed by description of our model, experimentation and results.

II. RELATED WORK

Identification of malicious domain name is not a new problem. It attracts the attention of researchers toward the end of 2000 decade. Researchers had experimented with various active and passive techniques [02]. Techniques of malicious domain name detection can be classified in two classes- based on text analysis and based on network analysis. Recently some hybrid techniques are also tested by researchers. Text based approaches utilize features of domain name like the length of a given domain, number of dots, number of special characters, frequency of certain characters etc. On the other hand network based methods utilize information obtained from the network like network traffic analysis, timing information registration information etc. Zhang et. al., [11] proposed a textual feature based methods which employs both local and global textual features and reported high precision, recall and f-score. Sharma et. al., [10] uses traffic pattern analysis to identify malicious domains. They proposed a system named BotMAD-Botnet Malicious Activity Detection based on DNS traffic pattern analysis to identify class of botnet family. Zhang et.al., [24] proposed a system BotDigger that can detect DGA based bots using DNS traffic. Evidence like quantity, linguistic and temporal features of domain name are exploited to identify the botnet. Results show that BotDigger detects all the Kraken bots and 99.8% of Conficker bots.

Revised Manuscript Received on August 30, 2019.

* Correspondence Author

Gurpreet Singh Josan*, Department of Computer Science, Punjabi University, Patiala, India. Email: josangurpreet@pbi.ac.in

Jagroop Kaur, Department of Computer Science and Engineering, Punjabi University, Patiala, India. Email: jagroop_80@rediffmail.com

A reputation based on past history of DNS server had been employed by Sharifnya and Abadi, [12] to detect DGA generated domain names. Their system automatically assigns a high negative reputation score to each host that is involved in suspicious bot activities. Suspicious group activity matrix and suspicious failure activity matrix are used to mark the reputation of a host. Host getting high negative score is marked as infected bot. Haddadi et. al., [4] suggest an evolutionary computation technique based on Stateful-SBB to detect malicious botnet. Shi et. al., [14] proposed Extreme Learning Machine (ELM) based technique to detect malware domain names. More than 95% accuracy has been reported by the authors. They also claim fast learning rate of algorithm makes it efficient and effective. Yadav et. al., [22] developed a methodology to detect "domain fluxes" in DNS traffic. They proposed that automatically generated domain names have some patterns that are not found in domain names generated by human beings. They exploit the distribution of alphanumeric characters as well as bigrams. Tian et. al., [13] also used multiple features like textual features, traffic statistics features etc. to classify malicious domains. Detection of malicious domain on cellular network has been explored by Wang and Shirley [21]. Apart from traditional quantitative features of domain names, a word segmentation algorithm was also used to segment the domain names. These segments are used to expand feature set. Their approach is effective as it can detect malicious domain in near real time. Mowbray and Hagen [7] suggested a technique for discovering DGAs from Domain Name Service (DNS) query data. The idea is to use the second-level string lengths in the domain names that they query.

A thorough review of state of the art in field has been provided by Sahoo et.al. [3] and Zhauniarovich et. al. [23] and can be consulted for further insight. Recently researchers experimented with deep neural network based models and conclude that LSTM network shows promising results for identifying malicious domain names [15], [16] and [19].

III. METHODOLOGY

The focus of this paper is on two tasks viz. identification of malicious domain name and mapping of domain name to their botnet family. Clearly, task 1 is a binary classification task where as task 2 is a multi-class classification task. We decided to use deep neural networks for both the task as they are proved to be efficient where input data is large and they are capable of auto learning the inherent features thereby relaxing the feature engineering part. Every domain name is treated as sequence of characters and features of domain name are obtained by passing it into LSTM network. Two models have been employed for both the task. First model is made up of single layer Bidirectional LSTM network whereas second model consist of two layers of Bidirectionl LSTM network. Bidirectional models are capable of capturing features in both forward as well as backward direction (figure 1). Fully connected layer has been used for classification purpose. For using these models we need to preprocess the input data.

A. Preprocessing

We process the input domain name by splitting each name

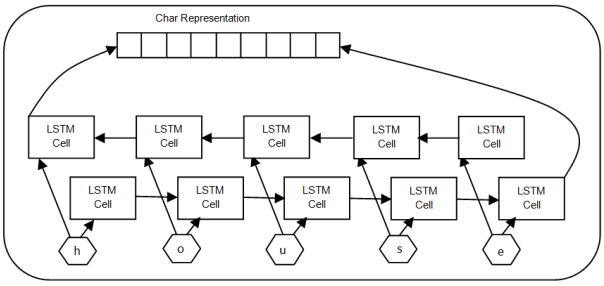


Fig. 1. Bidirectional LSTM network for sequence feature extraction

to make a sequence of characters. As LSTM network expects fixed length input, length of every sequence is fixed to be 196 characters for task 1 and 91 characters for task 2. Shorter sequences are padded by zero. Punctuation marks like hyphen(-), dot(.) etc. plays important role in identifying class of domain name so they are not removed. All punctuation marks are included as separate token. Further, capitalization information is also retained.

B. Character Embedding

Character embeddings are generated for all the characters in input text. We decided to use randomly initialized character embeddings which are updated during learning phase.

C. Models

LSTM networks are proved to give good performance in sequence learning tasks. We experimented with single layer and two layer LSTM networks described in following section.

Single Layer Model: Figure 2 shows the single layer model. Character embedding of 100 dimensions has been used. LSTM layer contain 100 hidden units. Fully connected layer has been used for classification task. The output of LSTM layer is passed through dense layer which takes 200 values as input and produce the probability of classes using softmax activation function. Model has been compiled using adam optimizer with default learning rate. Each sequence is passed through this network for training the model.

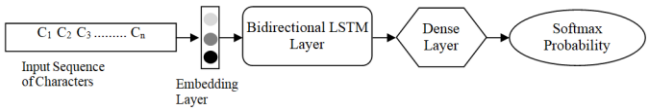


Fig. 2. Single Layer Model

Two Layers Model: Single layer model is further extended by adding one more layer as shown in figure 3. Character embedding also increased to 300 dimensions and each LSTM layer contain 250 hidden units. Fully connected layer has been used for classification task. The output of LSTM layers is passed through dense layer which takes 250 values as input and produces the probability of classes using softmax activation function. Model has been compiled using adam optimizer with default learning rate. Each sequence is passed through this network for training the model.

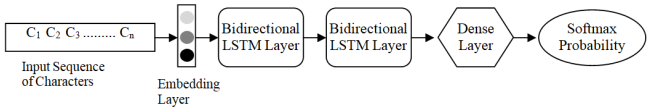
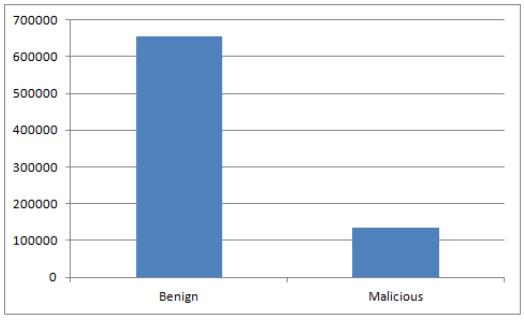


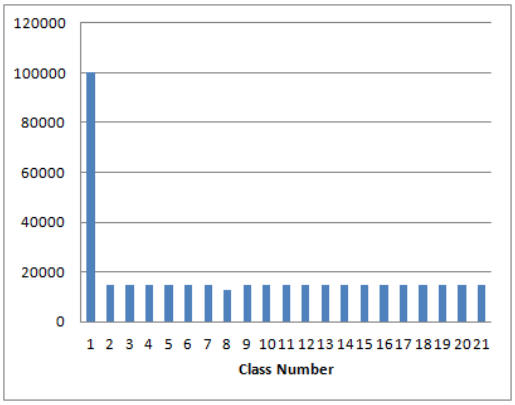
Fig. 3. Two Layer Model.

IV. DATA

Two datasets were collected from public resources. For the first dataset, malicious domain names were collected using publically available DGA algorithms [5], OSINT feeds [6] and netlab-360 [8] whereas legitimate domain names were collected from the Alexa [1] and openDNS [9] The second data set is collected privately within a lab using port mirroring approach. The passive sensors were deployed in an internal network. The detailed experimental setup followed during collection of DNS traffic data set from an internal network is reported in [15], [17], [18], [19], and [20]. For task 2, 20 DGA algorithms were considered. Total samples for task 1 are 790739 and for task 2 are 397777. Figure 4 shows the distribution of sample classes in each data.



a) **Training data for Task 1**



b) Training data for Task 2

Fig.4. Distribution of classes in training data

Task 1 is to identify malicious domain name thus it is a binary class problem. All data was labeled either 0 or 1 where 0 means benign domain name and 1 means malicious domain name. For task 2, the training data consist of domain names from 20 different classes as shown in table-I. For testing purpose, two separate data sets were collected for each task. One type of data set was collected from publically available DGA algorithms. The second type of dataset was collected from the real-time system. Test data statistics are given in table II.

Table I. Class labels for Task 2.

Class	Label	Class	Label
benign	0	pykspa	11
banjori	1	qadars	12
corebot	2	qakbot	13
dircrypt	3	ramdo	14
dnschanger	4	ranbyus	15
fobber	5	simda	16
murofet	6	suppobox	17
necurs	7	symmi	18
newgoz	8	tempedreve	19
padcrypt	9	tinba	20
proslikefan	10		

Table II. Class labels for Task 2

	DGA Algorithm	Real Time System
Task1	2457407	2922
Task2	587112	103200

V. EXPERIMENTS AND RESULTS

We used 90-10% split of the training data to validation split. Both the models were trained on CPU system with batch size of 512. Total 25 epochs were executed with early stopping on no improvement in validation accuracy. Figure 5 shows epoch vs training accuracies in all models.

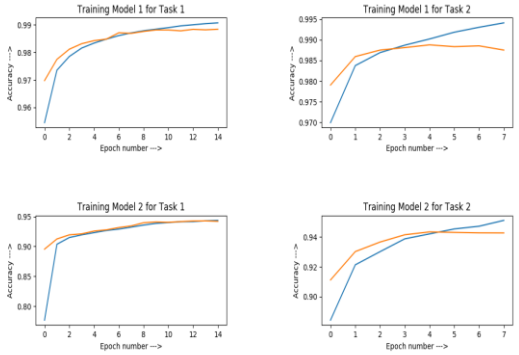


Fig. 5. Epoch vs Training Accuracy

Eight runs were tested using two test sets as described in table 3. Both models had a training accuracy of more than 99%. Validation accuracy of first model is about 98% whereas for second model is 99%. Precision, recall, F1-score and accuracy are reported. Table 4 shows that both models perform comparatively with two layer bidirectional LSTM model having slightly upper hand. For task 1, 98.9% and 71.1% accuracy has been reported by RUN 2 and RUN 4 respectively. For task 2, RUN 6 and RUN 8 are able to achieve 69.7% and 67.9% accuracy respectively. Thus Bidirectional LSTM model is capable of identifying publicly available DGA algorithm generated domain names but needs more tweaking for domains generated from private sources. For multitask classification, despite of high training accuracy, both the models failed on publicly as well as privately collected domain names.

Table III. Test Architecture

Run Number	Description
Run 1	Task 1 test 1 Single layer architecture
Run 2	Task 1 test 1 Two layer architecture
Run 3	Task 1 test 2 Single layer architecture
Run 4	Task 1 test 2 Two layer architecture
Run 5	Task 2 test 1 Single layer architecture
Run 6	Task 2 test 1 Two layer architecture
Run 7	Task 2 test 2 Single layer architecture
Run 8	Task 2 test 2 Two layer architecture

		Precision	recall	f1score	accuracy
Task 1	RUN 1	0.944	0.819	0.877	0.988
	RUN 2	0.947	0.822	0.881	0.989
	RUN 3	0.69	0.999	0.816	0.709
	RUN 4	0.692	0.999	0.818	0.711
Task 2	RUN 5	0.639	0.671	0.622	0.671
	RUN 6	0.689	0.697	0.658	0.697
	RUN 7	0.69	0.674	0.633	0.674
	RUN 8	0.694	0.679	0.636	0.679

VI. CONCLUSION

This article presents our work on identification of malicious domain name using LSTM based deep neural network. The task has been divided further into two parts. First part is a binary classification problem where system needs to classify a domain name as benign or malicious. Second task is a multi-class problem where botnet class of a domain name is needed to be identified. We try to tackle the problems using deep learning techniques. Two models were tested viz. single layer bidirectional LSTM and 2 layers bidirectional LSTM. Bidirectional networks are our choice as we want to capture the character sequence features both in forward as well as backward direction. System was tested on two datasets and performance of 2 layers Bidirectional LSTM model is found to have slightly upper hand than single layer network. While the performance of system on binary classification task is satisfactory, multiclass classification task needs more attention. Extensive experimentation is needed for optimizing multiclass classifier which will further helps in improving the output. The work opens new directions for further research in the field as there is still room for the improvements. As future work, we plan to study the DGA algorithms thoroughly to understand the domain generation process and identify the helpful features for this task. Hybrid methods can be tried for the multiclass problem and is one of the target areas in future work.

REFERENCES

1. Alexa. “Does alexa have a list of its top-ranked websites?” URL https://support.alexa.com/hc/-enus.
2. Bilge, Leyla, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. “Exposure: Apassive dns analysis service to detect and report malicious domains.” ACM Transactions on Information and System Securi, 16, 04 2014.
3. Doyen Sahoo, Chenghao Liu, and Steven C. H. Hoi. “Malicious URL detection using machine learning: A survey.” CoRR, abs/1701.07179, 2017. URL http://arxiv.org/abs/1701.07179.
4. Haddadi, Fariba, H. Gunes Kayacik, A. Nur Zincir-Heywood, and Malcolm I. Heywood. “Malicious automatically generated domain name detection using stateful-sbb.” In Anna I. Esparcia-Alcázar, editor, Applications of Evolutionary Computation, pages529–539, Berlin, Heidelberg, 2013.Springer Berlin Heidelberg. ISBN 978-3-642-37192-9.
5. Johannes Bader. “Domain generation algorithms.” https://github.com/baderj/domain_generation_algorithms, 2015.
6. John Bambenek. “Osint feeds.” URL http://osint.bambenekconsulting.com/feeds/.
7. M. Mowbray and J. Hagen. “Finding domain-generation algorithms by looking at length distribution.”In 2014 IEEE InternationalSymposium on SoftwareReliabilityEngineering Workshops, pages 395–400, Nov 2014. doi: 10.1109/ISSREW.2014.20.
8. Netlabs. “Dga families.” URL https://-data.netlab.360.com/dga/.
9. OpenDNS. “Opendns domain list.” URL https://-umbrella.cisco.com/blog.
10. P. Sharma, S. Kumar, and N. Sharma. “Botmad: Botnet malicious activity detector based on dns traffic analysis.” In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pages 824–830, Oct 2016.
11. Panpan Zhang, Tingwen Liu, Yang Zhang, Jing Ya, Jinqiao Shi, and Yubin Wang. “Domain watcher: Detecting malicious domains based on local and global textual features.” Procedia Computer Science, 108: 2408 – 2412, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.05.204. URL http://www.sciencedirect.com/science/-article/pii/S1877050917307974.InternationalConferenceon Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.
12. R. Sharifnya and M. Abadi. “A novel reputation system to detect dga-based botnets.” In ICCKE 2013, pages 417–423, Oct 2013. doi: 10.1109/ICCKE.2013.6682860.
13. S. Tian, C. Fang, J. Liu, and Z. Lei. “Detecting malicious domains by massive dns traffic data analysis.” In 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), volume 01, pages 130–133, Aug 2016. doi: 10.1109/IHMSC.2016.53.
14. Shi, Yong, GongChen, and Juntao Li. “Malicious domain name detection based on extreme machine learning.” Neural Processing Letters, Jul 2017. ISSN 1573-773X. doi: 10.1007/s11063-017-9666-7. URL https://-doi.org/10.1007/s11063-017-9666-7.
15. Poornachandran Prabaharan Vinayakumar R., Soman K.P. “Evaluating deep learning approaches to characterize and classify malicious url’s.” Journal of intelligent and fuzzy systems, 34: 1333–1343, 2018.
16. Poornachandran Prabaharan Vinayakumar R., Soman K.P. and Sachin Kumar S. ”Evaluating deep learning approaches to characterize and classify the dgas at scale.” Journal of intelligent and fuzzy systems, 34, 2018. doi: 10.3233/JIFS-169423.
17. Prabaharan Poornachandran, Vinayakumar R, Soman KP and Pradeep Menon. ”A deep-dive on machine learning for cybersecurity use cases.” Brij Gupta, Michael Sheng (eds) Machine Learning for Computer and Cyber Security: Principle, Algorithms, and Practices, InPress.
18. Soman K. Vinayakumar R., Poornachandran P. “Scalable framework for cyber threat situational awareness based on domain name systems data analysis.” Big Data in Engineering Applications, page 113-142, 2018.
19. Soman K.P. Vinayakumar R. and Poornachandran Prabaharan. "Detecting malicious domain names using deep learning approaches at scale." Journal of intelligent and fuzzy systems, 34: 1355–1367, 2018.
20. Soman Kp Vysakh S Mohan, Vinayakumar R and Prabaharan Poornachandran. "S.p.o.o.f net: Syntactic patterns for identification of ominous online factors." BioSTAR 2018, In Security and Privacy (SP), InPress.

21. Wei Wang and Kenneth E. Shirley. ”Breaking bad: Detecting malicious domains using word segmentation.” CoRR, abs/1506.04111, 2015. URL http://arxiv.org/abs/1506.04111.
22. Yadav, Sandeep, Ashwath Kumar Krishna Reddy, A.L. Narasimha Reddy, and Supranamaya Ranjan. ”Detecting algorithmically generated malicious domain names.” In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC ’10, pages 48–61, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0483-2. doi: 10.1145/1879141.1879148. URL http://doi.acm.org/10.1145/1879141.1879148.
23. Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. ”A survey on malicious domains detection through DNS data analysis.” CoRR, abs/1805.08426, 2018. URL http://arxiv.org/abs/1805.08426.
24. Zhang, Han, Manaf Gharaibeh, Spiros Thanasoulas, and Christos Papadopoulos. ”Botdigger: Detecting dga bots in a single network.” In Proceedings of the IEEE International Workshop on Traffic Monitoring and Analaysis, pages 16–21, Louvain La Neuve, Belgium, April 2016. IEEE. doi: http://dx.doi.org/10.1109/ICIMP.2010.11. URL http://www.cs.colostate.edu/hanzhang/papers/BotDigger-TMA16.pdf

AUTHORS PROFILE

Gurpreet Singh Josan is working as Associate professor in department of computer science at Punjabi University Patiala. He obtained his Ph.D. in computer Science and Engineering from Punjabi University Patiala in 2009. He has more than 18 years of teaching experience. His area of interest is Natural Language Processing and Machine Learning.

Jagroop Kaur is working as Assistant professor in department of computer engineering at Punjabi University Patiala. She obtained her master degree in computer Science and Engineering from Punjabi University Patiala in 2007. She has more than 18 years of teaching experience. Her area of interest is Natural Language Processing, Machine Learning and Social Media Text Mining.