Exploring and comparing various machine and deep learning technique algorithms to detect domain generation algorithms of malicious variants

Anoop Reddy Thatipalli, Preetham Aravamudu, K. Kartheek, Aju Dennisan
Department of Information Security, Vellore Institute of Technology University, Vellore, India

Article Info

Article history:

Received Mar 5, 2021
Revised May 3, 2022
Accepted Jun 7, 2022

Keywords:

Botnet detection
Deep learning
Domain generation algorithm detection
Machine learning
Malicious domain names

ABSTRACT

Domain generation algorithm (DGA) is used as the main source of script in different groups of malwares, which generates the domain names of points and will further be used for command-and-control servers. The security measures usually identify the malware but the domain name algorithms will be updating themselves in order to avoid the less efficient older security detection methods. The reason being the older detection methods does not use either the machine learning or deep learning algorithms to detect the DGAs. Thus, the impact of incorporating the machine learning and deep learning techniques to detect the DGA is well discussed. As a result, they can create a huge number of domains to avoid debar and henceforth, block the hackers and zombie systems with the older methods itself. The main purpose of this research work is to compare and analyse by implementing various machine learning algorithms that suits the respective dataset yielding better results. In this research paper, the obtained dataset is pre-processed and the respective data is processed by different machine learning algorithms such as random forest (RF), support vector machine (SVM), Naive Bayes classifier, H20 AutoML, convolutional neural network (CNN), long short-term memory neural network (LSTM) for the classification. It is observed and understood that the LSTM provides a better classification efficiency of 98% and the H20 AutoML method giving the least efficiency of 75%.

Corresponding Author:

Preetham Aravamudu
Department of Information Security, Vellore Institute of Technology University
VIT, Vellore Campus, Tiruvalam Rd, Katpadi, Vellore, Tamil Nadu 632014, India
Email: preetham.a2019@vit.ac.in

1. INTRODUCTION

The internet is widely used and it has high standard security strategy team to identify the domain generation algorithm (DGA) traffic through older methods. Also, the security team will be providing a huge list of documents as to generate the list of domains for potential C2 traffic. Then the method they follow for finding the domain groups of the DGA algorithms are using more statistical properties of the DGA. The main drawback of the older methods is not used for protecting the system from recent domains and more on time detection. In this work, a technique to detect randomly generated domains using machine learning algorithm model [1] such as support vector machine (SVM), AutoML: H20, Naïve Bayes classifier and random forest (RF), is being presented. Machine learning algorithms such as supervised learning algorithms, namely random forests (RFs) for decision making, SVM to process the labelled dataset predict the optimal hyper plane, thereby categorizing data. The classification is based on the structural, linguistic and statistical features of the respective domains. The second stage drawback with the machine learning algorithm is the "handcrafted features" which have derived variables, covariates, features being predictable by intruders and their time complexities in real time detection. Henceforth, to overcome this drawback, "learned-features" implementation is made using deep learning algorithms, to achieve better performances supported by deep learning algorithms such as long short-term memory neural network (LSTM) and convolutional neural network (CNN). As the second phase of the work, the dataset would be measured for the efficiency metrics with standard parameters amongst the entire set of proposed algorithms. Final phase of the work describes the solution to a better scheme of algorithms that could be used to detect the malicious variant.

2. LITERATURE SURVEY

The related works of the identification of DGA botnets that have been attempted using different technologies have been discussed in this section. The purpose remains where the use of recent technologies detects the pseudo random domain names tries to connect to the command and connect (C2) server. The work of Vinayakumar *et al*. [2] have given insights on how to deal with DGA Botnets using deep learning and machine learning algorithms, which alternates the idea of blacklisting the domain names which is a non real-time statistical machine learning approaches. Deep learning methods, resembling classical machine learning methods, suggested in their work leverages detection on per domain bases, where feature engineering is not used and circumventing is not possible.

Woodbridge *et al*. [3] have solutions from domain name system (DNS) query blacklisting is that and real time detection, with DGA classifiers and leverage the long short-term memory (LSTM). The work provides an in-depth analysis of the classifier functional interpretability at each layer. The data training set remains the key for the performance metrics of the detection, where best results of classification are deployed at the easiest possible.

Zhou *et al*. [4], proposed a general system to detect the DGA with a new model with high coverage. This helps to understand the algorithms used in high range accuracy detection. The word level and character level analysis done using deep learning algorithm (convolution neural network). Results of the paper concludes that the work to categorize the domains into two or more classifier.

Sharifnya and Abadi [5] planned a DGA grounded botnet detection algorithm by grouping the DNS queries of the host and also try to test it. the calculation helps in the understanding the possibilities of the hosts be a Botnet. Zhang *et al.* [6] used the NXDomain traffic clustering, classification of string features, and other methods that are frequently used such as number and alphabet domain classifier. The neural network has the entropy, bigram and length detections. it is layered neural network approach achieves 94% experimental results.

Understanding the algorithms with the help of their research-based works was essential in the project. Breiman [7] had contributed for the RF as a combination of prediction trees, where each tree of the algorithm works independently and with unique random vector samples. Contribution for the classification of malware, RF algorithms are trained with different data set and are unique in its training and correlates distinctively, that earns better results.

Ren *et al*. [8] related Naïve Bayes classification to uncertain data without much trained dataset. The results have been shown that the prediction is far more unique than the theoretical approaches. Yeo *et al.* [9] have successfully achieved a high accuracy in their malware detection, where they have used CNN, SVM, RF, multi-layered perceptron (MLP). The high range of accuracy was achieved only due to the use of 35 features extracted from the packet supervision, rather than focusing on the IPs and the ports. The overall survey works are more insightful as the work of Idika and Mathur [10] suggests techniques, samples and have also proposed a classification method, which were created after understanding the short-comings of the signature-based, specification based and anomaly based detection methods. The work finally suggested that commercial-off-the-shelf (COTS) malware detector is easier to obfuscate.

3. METHODS

Domain name generation algorithm (DGA) is a botnet malware that is responsible for a continuous communication between the intruder and the bots. The practical challenges faced are majorly on the false positives of the malware distribution that certainly reduces the accuracy of intrusion detection and several limitations. Domain generation have been intercepted with different techniques, where the challenge lies with the real-time detection and security. Lack of real-time security is the disadvantages of the existing algorithms.

Hence, its approach with the methods of machine learning and deep learning concepts uses automation NXDomain [11] classification and intelligence. It uses two supervised learning algorithms such as RFs and SVM. These two said algorithms normally utilizes the structural and self-structural features for detecting the domain data. When the ancient methods are used in finding the malicious codes, hackers just change the custom code to bypass the security strategy model. This is the reason, deep learning and neural network approaches are brought in and considered and therefore it acts like a firewall so that it is very hard for the hackers to discharge this. All the learning algorithms will be using three different datasets out of those two datasets will be malicious and one as the group of good and bad domains as shown in Table 1.

Table 1. DGA family classification

DGA_Family	Domain	Type
none	prat.pt	Normal
banjori	bxjofordlinnetavox.com	DGA
emotet	tbaccrnxirtmuusq.eu	DGA
rovnix	fbo6fssycmvf16nb47.net	DGA
none	giftcardsinfo1.icu	Normal

3.1. Data representation

Data is represented through data domains that have been used as non-structural data. It is not like structural data for which it does not have any rules and regulations. In this paper work, the machine learning and deep learning techniques for analyzing the data are discussed where these two different approaches utilize the old-styled machine learning methods. Here, the respective algorithm transforms the data to a complete structural data and thus the deep learning uses the same uni-structural data for the brain processing methods. This processing method is known as the artificial neural networks that processes different steps of dataset.

3.2. Feature engineering

Machine learning is used for the attribute domain and that is not sufficient in this case. It needs more definite feature sets for which it requires the knowledge and the respective references for further processing. The features are mainly classified as: structural features shown in Table 2, linguistic features shown in Table 3, statistical features shown in Table 4.

Table 2. Structural features

Structures	Ex: Hamata.pt	Ex: husjnshdj.eu
Domain names	7	20
NoS	1	2
Length of mean	3.0	9.0
prefixes	0	0

Table 3. Linguistic features

Structures	Ex: huskak.pt	Ex.ppposft.eu
Number of digits	0	0
Ratio of vowels	0.4	0.25
Digits ratio and vowels	0.33	0.0

Table 4. Statistical features

Structures	Ex.hshsa.pt	Ex.jdbsjhss.eu
Frequent letters in names ratio	0.3	0.4
Successive letters ratio	0.5	0.725
Uninterrupted digits ratio	0	0
Change in the names	2.34	3.6

3.3. DGA detector system

The proposed DGA Botnet detector system is the model that is a hybrid culmination of the selective machine learning and deep learning algorithms described in the upcoming sections. The overall model Figure 1 comprises of these algorithms as a system, where the algorithms are trained using the similar dataset and the results are correlated: this is based on its accuracy and performance in the detection process. DGA detector system as shown in Figure 1.

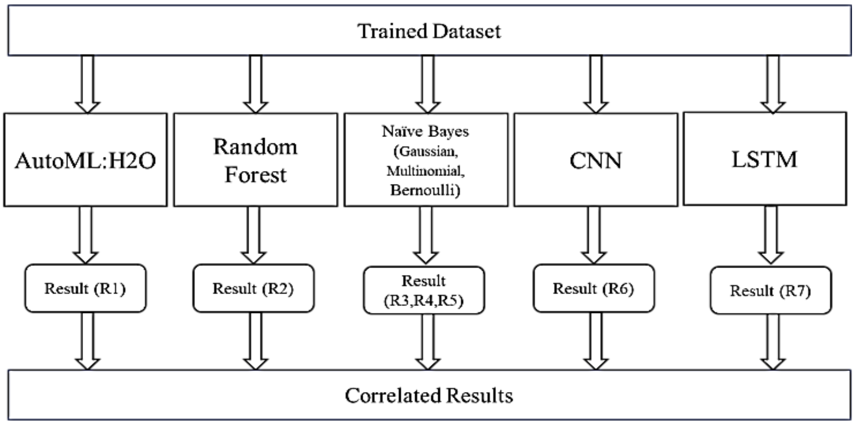


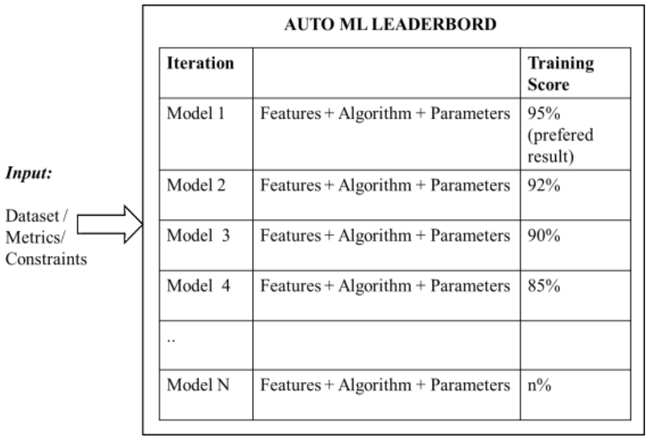
Figure 1. DGA detector system

3.3.1. AUTOML: H20

In foremost productions, AutoML is most important functionalities which automated type of algorithms which produce one of the best models. The biggest advantage of machine learning auto H20 is needed for finding the best dataset model. Figure 2 is shown AutoML model.

3.3.2. Random forest (RF)

RF is used for making a decision in random data sets which uses lot of decision trees to guess the result and then it starts the innermost decision trees for voting and selection. Figure 3 refers the RF model. Independent decision trees are care- fully designed based on the domain attributes, of which the majority of the decision is predicted as the result of the system, as a whole.



AUTO ML LEADERBORD

Iteration		Training Score
Model 1	Features + Algorithm + Parameters	95% (prefered result)
Model 2	Features + Algorithm + Parameters	92%
Model 3	Features + Algorithm + Parameters	90%
Model 4	Features + Algorithm + Parameters	85%
..		
Model N	Features + Algorithm + Parameters	n%

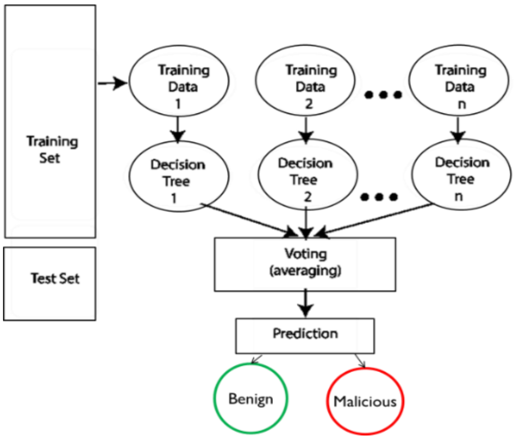


Figure 2. AutoML model

Figure 3. RF model

3.3.3. Naïve Bayes classifier

Naïve Bayes is used for the classification of the different classes to a single class. The malware filters of Naive Bayes are based on the benign or malicious is based on the categorically inputted attributes. The (1) depicts the model’s outcome.

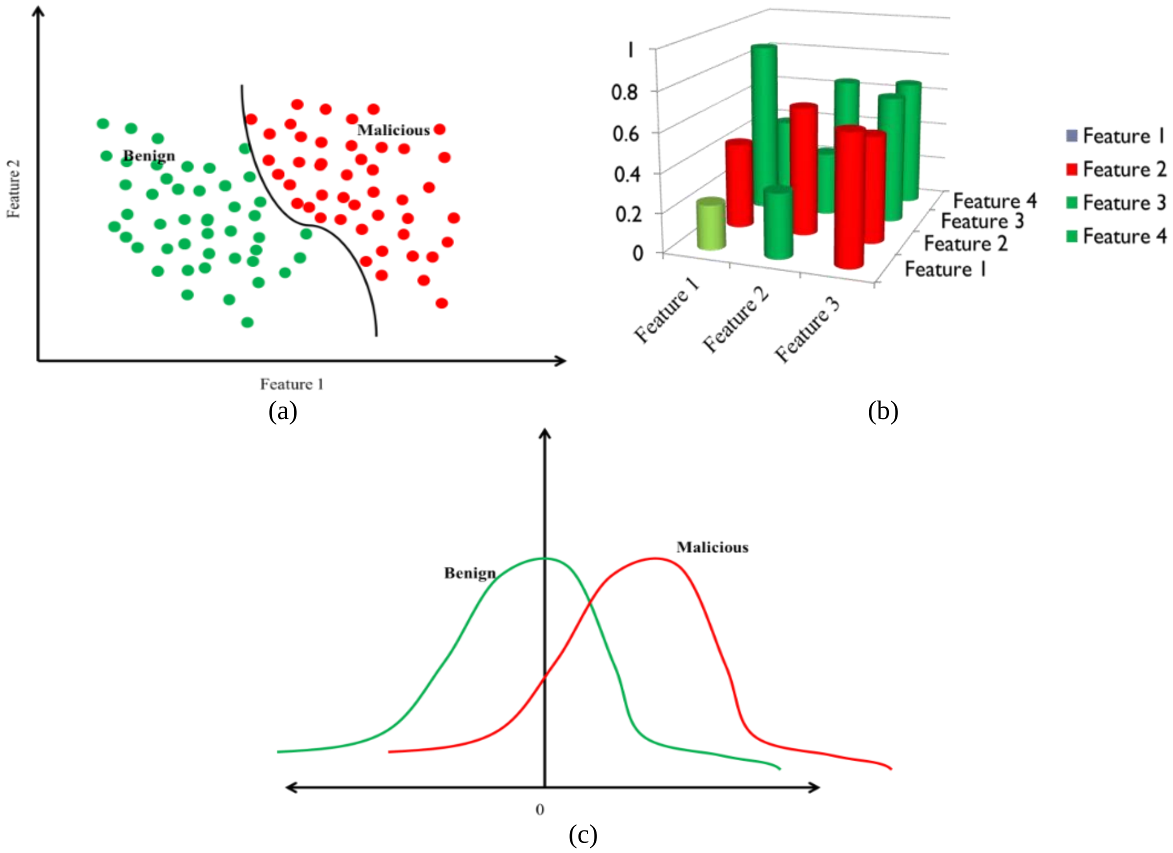
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{1}$$

were,
P(c|x) is the posterior probability of class (c, target) give predictor (x, attributes)
P(c) is the prior probability of class.
P(x|c) is the likelihood which is the predictors's probability of the given class.

P(x) is the predictor probability at the first.

This classifier adopts that the existence of a specific feature in a session is unconnected to the occurrence of any other feature. In this classifier three different models are proposed: Gaussian, Multinomial, and Bernoulli. Bernoulli Naïve Bayes model assumes the features of the domain to have only two possible value, and hence discrete prediction of benign or malicious is made out. The Figure 4(a) represents the model, where the line of curve between benign and malicious, is obtained and a clear distinction between them is made based on the train and test datasets.

Multinomial Naïve Bayes model assumes the features of the domain to have discrete set of value. The labelling of data based on more than one feature is studied and the probability of those features to be benign or malicious is predicted as shown in Figure 4(b). Gaussian Naïve Bayes model normalizes the test and train dataset results and differentiates the benign and malicious in a distinctive way, given a continuous range of values to the feature and the possibilities of being the malicious domain as described in the Figure 4(c). The features are studied as continuous data in this model.



(a) (b)

(c)

Figure 4. Naïve Bayes classifier model (a) Gaussian, (b) Multinomial, and (c) Bernoulli

3.3.4. Convolutional neural network (CNN)

CNN is proposed neural network model which implements the text classifier methodology for decrease the upfitting by increase in the intake data and failure layer elimination and totaling the regularities. CNN model as shown in Figure 5.

3.3.5. Long short-term memory neural network

LSTM is proposed neural network model as shown in Figure 6, which are used for making guesses, dispensation and categorizing the input data. The features are studied as continuous data in this model. The overall architecture in Figure 7 gives better performance than the current one in the test database, we can test its actual impact on the application by having sample predictions for a small fraction of our application end users. While observing performance, the detector system increases the rate of test users gradually with the new model in the hope that nothing will break. If the new dataset yields better result, the trained database will be updated by always returning the prediction of the new model.

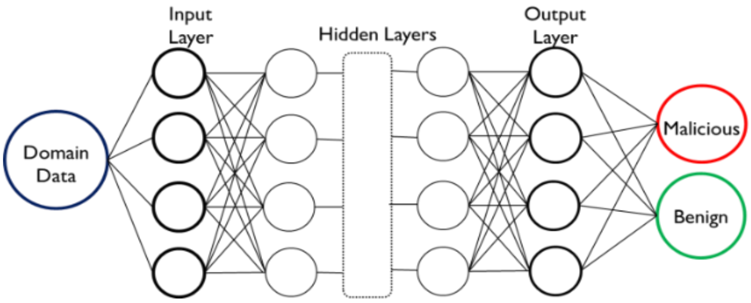


Figure 5. CNN model

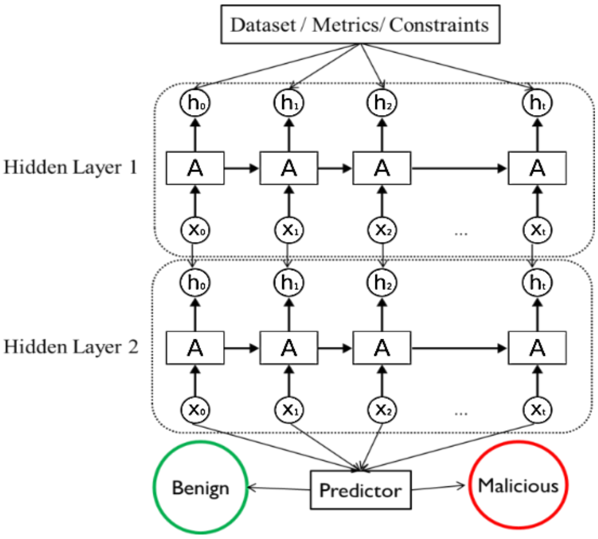


Figure 6. LSTM model

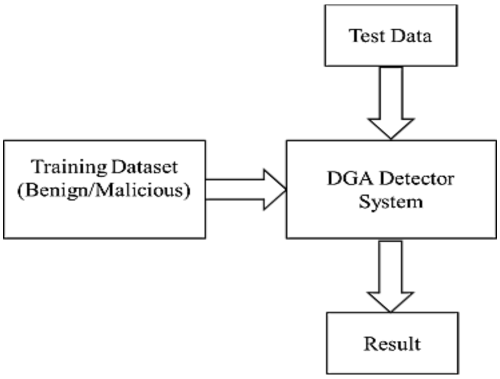


Figure 7. The overall architecture

4. IMPLEMENTATION

The detection system for the DGA Botnets is more towards the performance of the deep learning and advanced machine learning algorithms used. The performance metrics are measured and represented as data graph.

4.1. Random forest (RF)

RF which is a classification of decision trees which is seen for forecasting the result It get around 92.45% accuracy and test dataset gives us 91.95%. Different features such as vowel ratio in the domain names, digit ratio, Figure 8 describes the feature importance levels of different domain keyword. in the graph the X axis is the normalised frequency across different features.

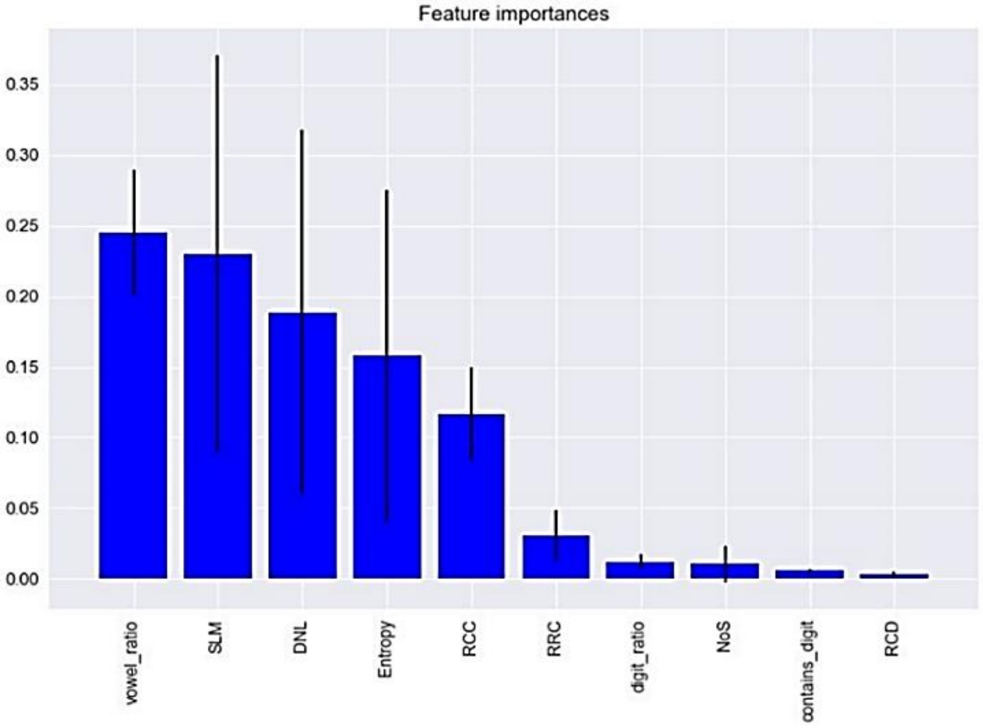
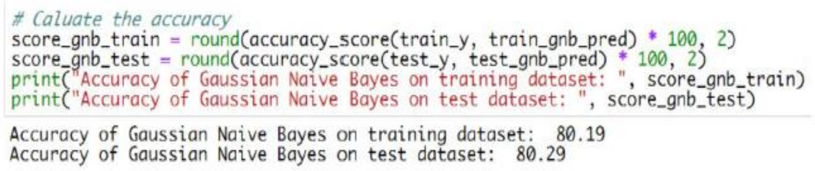


Figure 8. Accuracy of RF classifier

4.2. Naïve Bayes classifier

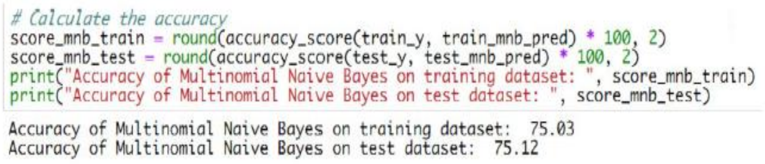
In the implementation, three type of Naïve Bayes models are analysed, namely the Gaussian Naïve Bayes model, multi-nominal Naïve Bayes model and Bernoulli Naïve Bayes model. Figures 9-11 represent the accuracy results of Gaussian, multinomial and Bernoulli Naïve Bayes models respectively.



```
# Caluate the accuracy
score_gnb_train = round(accuracy_score(train_y, train_gnb_pred) * 100, 2)
score_gnb_test = round(accuracy_score(test_y, test_gnb_pred) * 100, 2)
print("Accuracy of Gaussian Naive Bayes on training dataset: ", score_gnb_train)
print("Accuracy of Gaussian Naive Bayes on test dataset: ", score_gnb_test)

Accuracy of Gaussian Naive Bayes on training dataset:  80.19
Accuracy of Gaussian Naive Bayes on test dataset:  80.29
```

Figure 9. Accuracy of Gaussian Naïve Bayes model



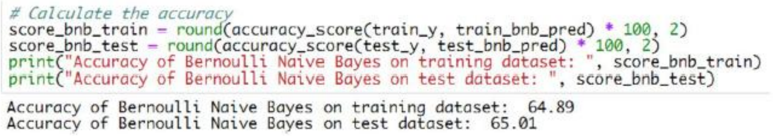
```
# Calculate the accuracy
score_mnb_train = round(accuracy_score(train_y, train_mnb_pred) * 100, 2)
score_mnb_test = round(accuracy_score(test_y, test_mnb_pred) * 100, 2)
print("Accuracy of Multinomial Naive Bayes on training dataset: ", score_mnb_train)
print("Accuracy of Multinomial Naive Bayes on test dataset: ", score_mnb_test)

Accuracy of Multinomial Naive Bayes on training dataset:  75.03
Accuracy of Multinomial Naive Bayes on test dataset:  75.12
```

Figure 10. Accuracy of multinomial Naïve Bayes model

4.3. H20 (Industry based AutoML)

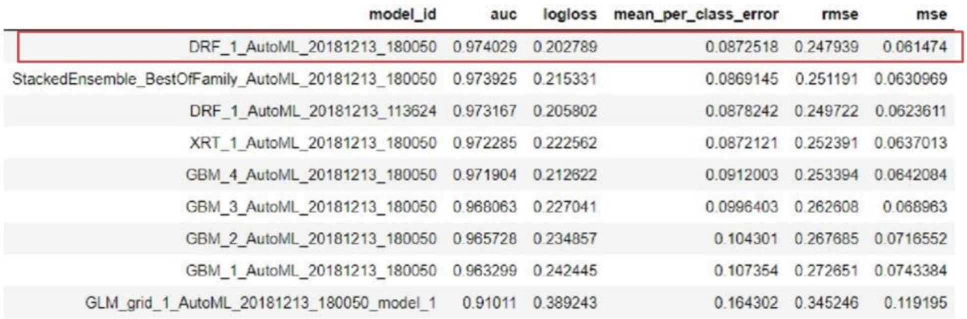
Here It use H20 which is an industry based AutoML landscapes that runs through the algorithms and their guidelines to generate a leading model. Which uses lot of models and compares the models for finding the best suitable models of the dataset, which performs 75 percentage accurately every time. Figure 12 represents the accuracy table of all the algorithms fed to AutoML, the measure of maximum accuracy is determined by the value of AuC, were DRF AutoML outperforms with an Auc=.974029.



```
# Calculate the accuracy
score_bnb_train = round(accuracy_score(train_y, train_bnb_pred) * 100, 2)
score_bnb_test = round(accuracy_score(test_y, test_bnb_pred) * 100, 2)
print("Accuracy of Bernoulli Naive Bayes on training dataset: ", score_bnb_train)
print("Accuracy of Bernoulli Naive Bayes on test dataset: ", score_bnb_test)

Accuracy of Bernoulli Naive Bayes on training dataset:  64.89
Accuracy of Bernoulli Naive Bayes on test dataset:  65.01
```

Figure 11. Accuracy of Bernoulli Naïve Bayes model



model_id	auc	logloss	mean_per_class_error	rmse	mse
DRF_1_AutoML_20181213_180050	0.974029	0.202789	0.0872518	0.247939	0.061474
StackedEnsemble_BestOfFamily_AutoML_20181213_180050	0.973925	0.215331	0.0869145	0.251191	0.0630969
DRF_1_AutoML_20181213_113624	0.973167	0.205802	0.0878242	0.249722	0.0623611
XRT_1_AutoML_20181213_180050	0.972285	0.222562	0.0872121	0.252391	0.0637013
GBM_4_AutoML_20181213_180050	0.971904	0.212622	0.0912003	0.253394	0.0642084
GBM_3_AutoML_20181213_180050	0.968063	0.227041	0.0996403	0.262608	0.068963
GBM_2_AutoML_20181213_180050	0.965728	0.234857	0.104301	0.267685	0.0716552
GBM_1_AutoML_20181213_180050	0.963299	0.242445	0.107354	0.272651	0.0743384
GLM_grid_1_AutoML_20181213_180050_model_1	0.91011	0.389243	0.164302	0.345246	0.119195

Figure 12. Accuracy measure of H20

4.4. Convolutional neural network (CNN)

It used many CNN models here with lot of unique structure and configurations. The best CNN model It got is 1D and it should us accuracy around 80% with data testing. Figure 13 describes the accuracy measure of CNN, wherein the accuracy of two results, namely training dataset and test dataset are comparatively plotted, where X axis represents the percentage of accuracy for each epoch (along Y axis).

4.5. Long short-term memory neural network (LSTM)

LSTM is classified and processed and also used for making predictions of the layer-by-layer approach accuracy what It got is around 98%. Figure 14 represents the accuracy measure of LSTM, wherein the accuracy of training dataset and test dataset are comparatively plotted, where X axis represents the percentage of accuracy for each Epoch (along Y axis).

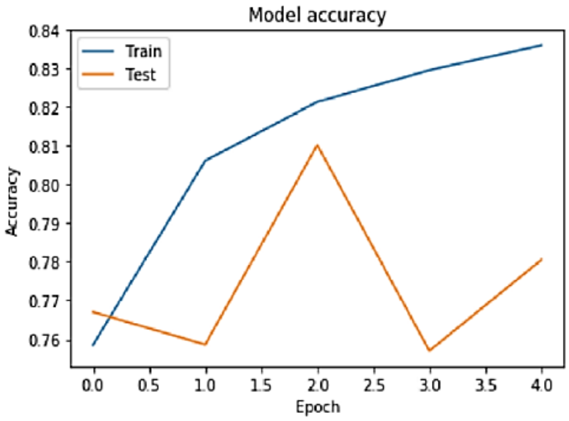


Figure 13. Accuracy measure of CNN

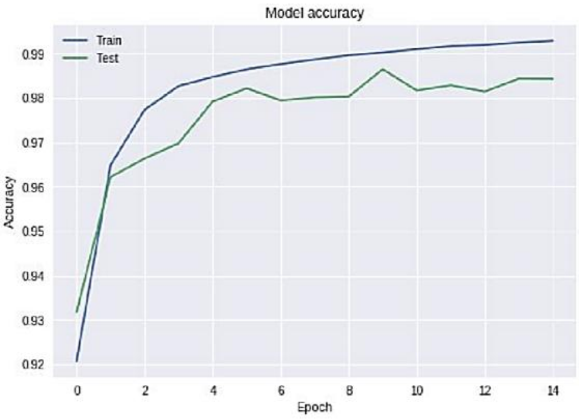


Figure 14. Accuracy measure of LSTM

Figure 15 tabulates the score of each model, wherein by working and doing comparison of all the models accuracy percentage. It is shown that both the AutoMl and LSTM will have best accuracy rate in DGA Detection but H20 requires high time to train the data set. The future scope of the work in detection of DGA would be reducing the training data set, and increasing the false positives rate beyond the designed model. This could be achieved by the use of most advanced and recent deep learning algorithms that could categorize the domains efficiently. This work would significantly impact the upcoming future works on real time DGA botnet detection.

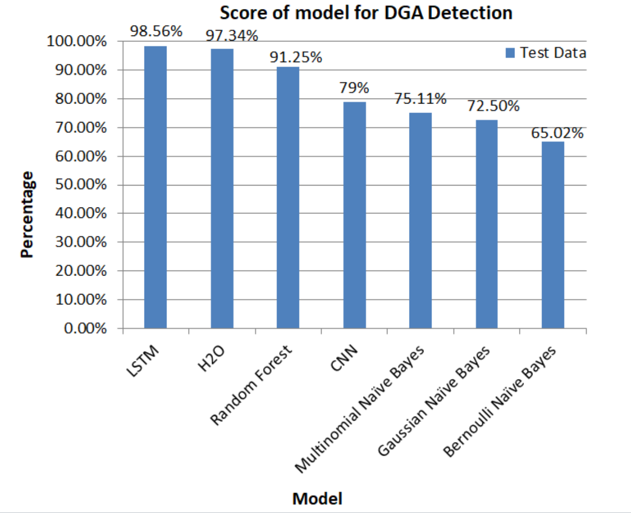


Figure 15. Accuracy comparison of the used algorithms

5. CONCLUSION

The work presents an approach to classify DGA generated domains using deep learning that has a technical advantage as they are unsupervised learning real-time classifiers and featureless. Therefore, there is no need to generate features manually, instead the features are self-extracted during the training. The LSTM, AutoML, RF, CNN, Naive Bayes are the selected algorithms for the work. The DGA families have concatenated the words randomly from the dictionaries, which had to be trained as the dataset. Analysis of the functional interpretability is worked on each layer of the classifier; these layers are different algorithm. RF makes the primary decision using the decision tree of malicious domain, then the Naïve Bayes classifier on secondary decision making. LSTM and CNN consolidation of the results of the other decision trees is made in the implementation. Thus, the experimentation results show that open-source dataset has tested the performance results with 90% false positive rates.

REFERENCES

[1] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, “Data mining methods for detection of new malicious executables,” *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 38–49, 2001, doi: 10.1109/secpri.2001.924286.

[2] R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, S. Akarsh, and M. Elhoseny, “Improved DGA domain names detection and categorization using deep learning architectures with classical machine learning algorithms,” in *Advanced Sciences and Technologies for Security Applications*, Springer, Cham, 2019, pp. 161–192.

[3] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, “Predicting Domain Generation Algorithms with Long Short-Term Memory Networks,” Nov. 2016.

[4] S. Zhou, L. Lin, J. Yuan, F. Wang, Z. Ling, and J. Cui, “CNN-based DGA detection with high coverage,” *2019 IEEE International Conference on Intelligence and Security Informatics, ISI 2019*, pp. 62–67, Jul. 2019, doi: 10.1109/ISI.2019.8823200.

[5] R. Sharifnya and M. Abadi, “A novel reputation system to detect DGA-based botnets,” *Proceedings of the 3rd International Conference on Computer and Knowledge Engineering, ICCKE 2013*, pp. 417–423, 2013, doi: 10.1109/ICCKE.2013.6682860.

[6] Y. Zhang, Y. Zhang, and J. Xiao, “Detecting the {DGA}-Based Malicious Domain Names,” Springer Berlin Heidelberg, 2014, pp. 130–137.

[7] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[8] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, “Naive bayes classification of uncertain data,” *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 944–949, 2009, doi: 10.1109/ICDM.2009.90.

[9] M. Yeo *et al.*, “Flow-based malware detection using convolutional neural network,” *International Conference on Information*

Networking, vol. 2018-Janua, pp. 910–913, Apr. 2018, doi: 10.1109/ICOIN.2018.8343255.

[10] N. Idika and A. P. Mathur, “A Survey of Malware Detection Techniques,” 2007.

[11] D. Anderson, T. Frivold, and A. Valdes, “Next-generation intrusion detection expert system (NIDES): A summary,” no. SRI--CSL--95--07, 1995, [Online]. Available: https://www.cerias.purdue.edu/apps/reports_and_papers/view/974.

BIOGRAPHIES OF AUTHORS

Anoop Reddy Thattipalli received the M. Tech degree in information Security Engineering from the Vellore Institute of Technology (VIT), India, in 2021 and a B. Tech degree in Computer science engineering from Christu Jyothi Institute of Technology, India, in 2018. He has worked as full stack software developer and Security researcher. His current research interests include Secured Blockchain Technology and Security in Data science. He can be contacted at email: anoopreddy.thattipalli2019@vit.ac.in.

Preetham Aravamudu received the M. Tech degree in information Security Engineering from the Vellore Institute of Technology (VIT), India, in 2021 and a B. Tech degree in Electronics and Communication engineering from Pondicherry University, India, in 2017. His current research interests include automotive cybersecurity, Security Data science and threat modelling. He can be contacted at email: preetham.a2019@vit.ac.in.

Aju Dennisan received the M. Tech. Degree in Information Technology engineering from the Manonmaniam Sundaranar University, India, in 2004 and a Ph. D. degree in Computer Science Engineering from Vellore Institute of Technology, India, in 2016. He currently works as an Associate professor at School of Computer Science and Engineering, Department of Information Security, Vellore Institute of Technology, India. His current research interests include the cyber forensics and information security using data science. He is an active member ISTE (Indian society for Technical Education) and CSI (Computer society of India). He holds a multi-disciplinary field of expertise in guiding young researchers and has worked extensively in the areas of cryptography, image processing, remote sensing, machine learning, Deep learning, Artificial intelligence and Wireless sensor networks, and has contributed to more than 26 journals across different national and international publications. He can be contacted at email: daju@vit.ac.in.