IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 8, August 2021

DOI: 10.17148/IARJSET.2021.8844

A ML Framework For DGA-Malware Detection

R.Deepa[1], K.Gagan Kumar[2], G.Prashanth[3]

[1] Assistant Professor - CBIT, Hyderabad, Telangana

[2] UG- Information Technology, CBIT, Hyderabad, Telangana

[3] UG- Information Technology, CBIT, Hyderabad, Telangana

ABSTRACT: Malware has always been a threat to the computer world, but with fast growth in the use of the Internet, malware severely affects the computer world. Malware predictors and detectors are critical tools in defense against malware. The existing malware detectors and predictors have been created, the effectiveness of these detectors and predictors depend upon the techniques being used. This study specifically, addressed the following objectives. Real-time detection of domain names that are generated using the domain generating algorithms (DGA) is a challenging cyber security challenge. DGAs can constantly generate large amounts of domains to evade blacklist detection. Traditional malware control methods, such as blacklisting, are insufficient to handle DGA threats. In-order to solve this problem we decided to use machine learning algorithms to detect DGA domains and compare the performance of these algorithms. In this research project, we first performed feature engineering. Then applied preprocessed data to machine learning models like a random forest, LSTM, logistic regression.

Keywords—Blacklisting, Malware, domain generation algorithm (DGA), machine learning, security, networking

1. INTRODUCTION

As the Internet has become widely distributed, it is very vulnerable to malware hazards. Malware attackers can choose different targets or cyber-physical devices and attack them like mobile devices and connected vehicles. Many of the targets the threat actor attacks are susceptible to malware attacks due to mismanagement issues, poor patching behaviors, and dangerous 0-day attacks. Internet security vendors have provided several strategies to intercept DGA traffic. Traditionally, security providers would first decode the algorithm by applying reverseengineering. Generating a list of domains with potential C2 traffic. Another common strategy is to find similar domain groups by using their statistical properties to determine ifDGA generates a domain. The main disadvantage of traditionalstrategies is the lack of capability to be used for real-time detection and protection. To differentiate DGA domain namesfrom normal domain names, researchers have discovered that DGA-generated domain names contain significant features.Therefore, many studies aim to target blocking those DGA domain names as a defense approach.

The DGA that generates the domain fluxing botnet needs to be known so that thecounter measures are taken Initially we collect the benign and DGA domains and createa dataset. The dataset is splitted into training set, testing set.We have uploaded all the training set, testing set to Google drive for easy usage of them when required. We are provided with folders consisting of the domains with start time end timeand DGA family along with the respective categories.
We will now preprocess the domain data using python libraries. Now we can proceed to build simple models with respect to the algorithms. sklearn allows us to build random forest and logistic regression. Keras allows us to build neural networks effortlessly with a couple of classes and methods,after compiling the network we get the model. The model helps us to classify the domain whether it is DGA or normal domain. Feature extraction functions help us to extractrequired information from the given domain and statistical values are extracted by applying machine learning functions and we build models for respective algorithms and predict the accuracy.

2. LITERATURE SURVEY

Tommy Chin, Kaiqi Xiong, Chengbin Hu, Yi Li. In this paper, they proposed that DGA domains have groups of very significant characters from normal domains. By grouping domains according to their features, the authors applied a machine learning classifier to distinguish DGA domains from normal domains easily. Several machine-learning techniques have been studied to classify malicious codes. They include neural networks, support vector machines (SVM) and boosted classifiers. After applying the above algorithms models are created and accuracy is predicted.
Dingkui Yan, Huilin Zhang. In this paper, they proposeda DGA detection system, called Pontus, which is based upon

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 8, August 2021

DOI: 10.17148/IARJSET.2021.8844

powerful linguistics-based features. The features of Pontusare extracted exclusively from the individual domain name, Pontus has a good classification performance. Their systemis based upon the key insight that benign domain names and mAGDs differ greatly in the linguistic aspect. Benign domain names often represent some specific meanings, such as a brand name, a person’s name. Those domain names usually adhere to the regular linguistic pattern for fluent reading or easy remembering. However, the random-looking mAGDs disobey regular linguistic patterns. Though wordlist based mAGDs follow the regular linguistic pattern, they can be split into 2 or 3 words completely. Sometimes, the 2 or 3 words are separated by hyphens.

Vaclav aprenosil Ibrahin Ghafir. In this paper, they pro- posed a methodology for detecting any connection to or from malicious IP. The detection method is based on a blacklist of malicious IPs. They process the network traffic and match the source and destination IP addresses for each connection with IP blacklist. And the blacklist is automatically updated each day based on different intelligence feeds at once and the detection is in the real time. They have implemented the detection method by using top of Bro , which is a passive, open-source network traffic analyzer.
Since the DGA domain names are usually randomly generated, the lengths of DGA domains are very long. Such a feature can be used to detect DGA domains. That is, shorter DGA domain names are more difficult to be detected. This is because most normal domains tend to be short.
Ahluwalia Et Al[4].proposed a detection model that candynamically detect DGA domains. They apply information theoretic features based on a domain length threshold. Their approach can dynamically detect the DGA domains with any length. Many other studies have been done on DGA detection based on the DGA domain features.
Maetal[5]. proposed a lightweight approach to detect DGA domains based on URLs using both lexical and host-based features. They consider the lexical features of the URL such as length, number of dots, and special characters in the URL path.

3. DATASET COLLECTION

For this research, we have three datasets in total: one datasetfor Benign Domains, Alexa Top 1 Million Sites, which are a combination of good domains; two datasets for DGA Domains Bambenek Consulting provided malicious algorithmically- generated domains and 360 Lab DGA Domains . Combining these three datasets and shuffling them to generate the dataset for the project.Total datasets consist of more than 2 million domains.

4. Results and Discussion

After collecting dataset the following steps were followedas shown in Fig 1.

4.1 Model Creation

Now we can proceed to build a simple Random Forest model, LSTM model. Select random K data points from the training set: Build the decision trees associated with the selected data points (Subsets).Choose the number N for decisiontrees that you want to build. For new data points, find thepredictions of each decision tree, and assign the new data points to the category that wins the majority votes. We diddata preprocessing and we fitted the Random Forest model algorithm to the training set. Predicted the test result and foundout the accuracy of the result.

Keras allows us to build neural networks effortlessly with a couple of classes and methods. The Sequential class initializes a network to which we can add layers and nodes. The add method allows us to add layers of nodes to the initialized network. Keras allows us to build neural networks effortlessly with a couple of classes and methods. The Sequential class initializes a network to which we can add layers and nodes. The add method allows us to add layers of nodes to the initialized network

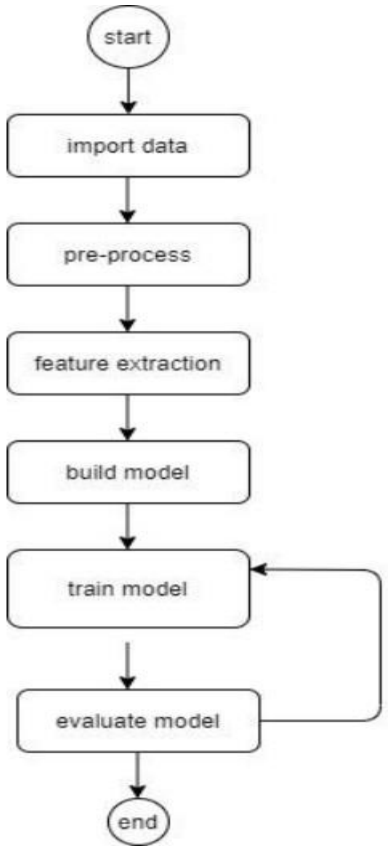


Fig. 1. Flowchart of the procedure followed on the dataset.

4.2 Feature Extraction

We will now extract the features from the training dataset which we have collected. We have merged both the benign domain names and DGA domain names and then made acopy of the original dataset for future use. Approximately we have taken 16 features into consideration. They are as follows domain length, subdomains number, subdomain length mean, has www prefix, contains single character subdomain,has hvltd, contains TLD subdomain, underscore ratio, contains IP address, contains digit, vowel ratio, digit ratio, ratio of repeated characters in a subdomain, ratio of consecutive consonants, ratio of consecutive digits, entropy. The process of reducing the vector dimensions is referred toas feature selection. At the end of this process, we expectthe selected features to outline the relevant information from the initial set so that it can be used instead of initial data without any accuracy loss. The feature extractor is used extract features from the domain names filtered in the first component. Each domain name is considered as a string. Applying features engineering first. Based on our knowledge and reference materials, three kinds of features will be generated: Structural Features; Linguistic Features; Statistical Features. For the first part of feature engineering: From Fig 2. nine structural features are generated. For example,prata.pt, DNL (The length of the domain name) is 8. It only has 1 subdomain, so its NoS value is 1. The length of the subdomain (SLM) is the length of ‘prata’, which equals 5.0.It does not have www Prefix, so its Hwp value is 0. ‘.pt’is a valid top-level domain, so its HVTLD domain is 1. Itdoes not contain a single-character subdomain, so the CSCS value is 0. So does the CTS. The ratio of underscore (UR)for example is 0 also. And it does not have an IP address.

Features	Ex: prata.pt	Ex: tbaxcrnxirtmuusq.eu
DNL (Domain Name Length)	8	19
NoS (Number of Subdomains)	1	1
SLM (Subdomain Length Mean)	5.0	16.0
HwP (Has www Prefix)	0	0
HVTLD (Has a Valid Top Level Domain)	1	1
CSCS (Contains Single-Character Subdomain)	0	0
CTS (Contains Top Level Domain as Subdomain)	0	0
UR (Underscore Ratio)	0.0	0.0
CIPA (Contains IP Address)	0	0

Fig. 2. Structural features

Based on linguistic analysis, three linguistic features are generated from the domain. Whether a domain contains a digit (contains digit), the ratio of the vowel in a domain and the ratio of the digit. The value of these linguistic features can be known from Fig 3.

RRC (The ratio of repeated characters in a subdomain)	0.25	0.33
RCC (The ratio of consecutive consonants)	0.4	0.625
RCD (The ratio of consecutive digits)	0	0
Entropy (The entropy of subdomain)	1.92	3.5

Fig. 3. Linguistic features

There are also 4 statistical features that will be generated. From Fig 4. RRC represents the ratio of repeated characters in a subdomain. RCC represents the ratio of consecutive consonants, RCD represents the ratio of consecutive digits andEntropy means the entropy of subdomain.

contains_digit (Contains digit)	0	0
Vowel_ratio (The ratio of vowel)	0.4	0.25
Digit_ratio (The ratio of vowel)	0.33	0.0

Fig. 4. Statistical features

4.3 Description

Our models will predict the label either 0 or 1 after the models applied on dependent and independent variables. Based on the feature extraction the domain is identified. We createda folder containing the group of multiple folders with thetext formatted files and csv files of dga domains and benign domains. Once after the loading domains from the different text files to csv files. Dga domains and bambeneck domains are concatenate to form single entity of dga domains and later it is concatenate with benign domains to make a single entity.The data is shuffled and copied to another file.The steps to performthe DGA detection is described as follows.

The Machine Learning framework consists of a classification model for preliminary distinguishing. In the classification, machine learning classifiers are used to distinguish DGA domains from normal domains in the set. Each of the features extracted is assigned a weight and the model uses the features from the previous section to calculate probabilities of the binary outcome and also sets a threshold to classify. It then classifies the domain name into one of the two categories (either DGA or normal) based on the calculated probability. In the classification, the extracted features from the feature extractor are used and tested against different machine learning classifiers includes Random Forest, LSTM-Long short term memory, Logistic regression

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

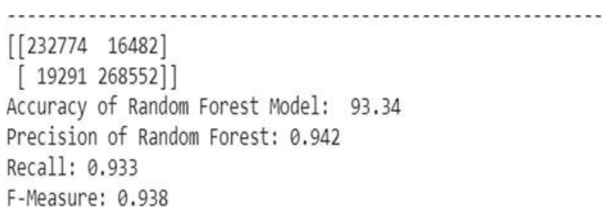
Vol. 8, Issue 8, August 2021

DOI: 10.17148/IARJSET.2021.8844

Random forest (RF): Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.
To create a random forest model, we use a random forest classifier function which classifies and gives the result given domain dga or benign. initially it Select random samples froma given dataset. Construct a decision tree for each sample and get a prediction result from each decision tree. Perform a vote for each predicted result. Select the prediction result with the most votes as the final prediction.
Random forests also offer a good feature selection indicator. Scikit-learn provides an extra variable with the model, which shows the relative importance or contribution of each feature in the prediction. It automatically computes the relevance score of each feature in the training phase. Then it scales the relevance down so that the sum of all scores is 1.
Logistic Regression: The LR is a predictive analysis. LR is used to describe data and to explain the relationship between onedependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.
LSTM: Long short-term memory (LSTM) is an artificialrecurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed forward neuralnetworks, LSTM has feedback connections. It can not only process single data points (such as images), but also entiresequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connectedhandwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).Initially load the data from txt file and csv file extensions and classify the labels according to the domain benign domainshas type and domain family, DGA domains has type ,domainfamily, start time ,end time for dga family.DGA domains arecollected from 2 different stages later combined together lateron combined with benign domains,drop duplicates if there any available,and shuffle data after removing duplicates.Copythe original dataset and perform operation on the copieddata. Feature extraction is applied on the data after applying thepython functions statistical data is extracted save the values to csv file for easy access furtherly, since the extracted features are 17 only 11 are used to build the model which is defaultto build the model for that few columns are dropped whichprovide only less information to classify the domains.The data verified for null values, once the data is sorted the data is ready to use for building the model. First, we separate the columns into dependent and independent variables (or features andlabels). Then we split those variables into a training and test set After splitting, we will train the model on the training set and perform predictions on the test set. After training, check the accuracy using actual and predicted values, precision, recall and f-measure are calculated.
Logistic Regression classifier is used to build the model for logistic regression. Training data is used to build the model andperform predictions on the test set. check the accuracy using actual and predicted values.

Lstm model is built by using five different layers: embedding, dense, lstm, activation, dropout layer and activation function sigmoid and optimizer as rmsprop, model is built by using training set, and accuracy and error rate is calculated. Binary_crossentropy is used to calculate the error rate. theaccuracy of the analysis increases with the num~~b~~er of theepochs.

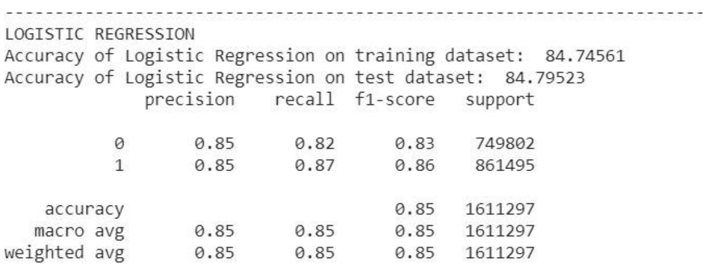
Random Forest Accuracy.



```
------------------------------------------------------------
[[232774  16482]
 [ 19291 268552]]
Accuracy of Random Forest Model:  93.34
Precision of Random Forest: 0.942
Recall: 0.933
F-Measure: 0.938
```

Fig. 5. Random forest accuracy.

Fig 5. gives the information of random forest algorithm which gives accuracy of 93.34%, precision of 0.942, recall of 0.933 and F-Measure of 0.938.



```
-----------------------------------------------------------------------
LOGISTIC REGRESSION
Accuracy of Logistic Regression on training dataset:  84.74561
Accuracy of Logistic Regression on test dataset:  84.79523
              precision    recall  f1-score   support

           0       0.85      0.82      0.83    749802
           1       0.85      0.87      0.86    861495

    accuracy                           0.85   1611297
   macro avg       0.85      0.85      0.85   1611297
weighted avg       0.85      0.85      0.85   1611297
```

Fig. 6. Feature importance.

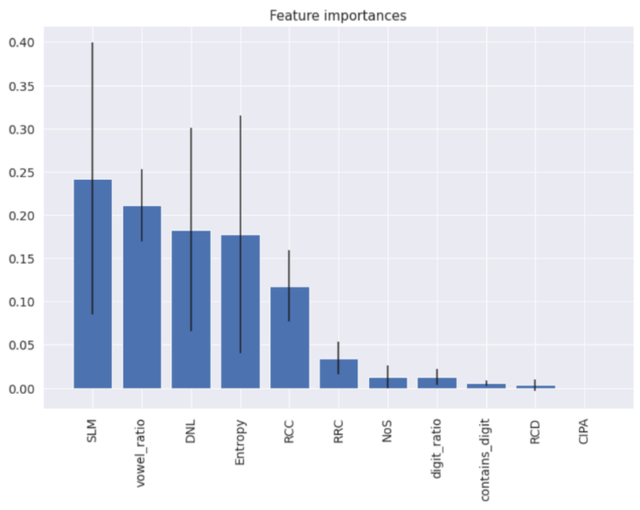
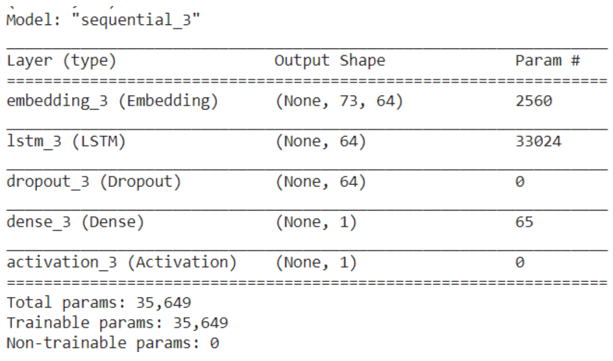


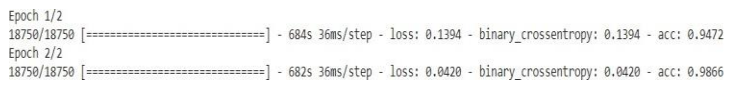
Fig. 7. Logistic Regression Accuracy.

Fig 7. gives the information of logistic regression algorithmwhich gives accuracy of 84.79%, precision of 0.85, recall of 0.87 and F-Measure of 0.85.



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 73, 64)            2560
_________________________________________________________________
lstm_3 (LSTM)                (None, 64)                33024
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 65
_________________________________________________________________
activation_3 (Activation)    (None, 1)                 0
=================================================================
Total params: 35,649
Trainable params: 35,649
Non-trainable params: 0
```

Fig. 8. LSTM model



```
Epoch 1/2
18750/18750 [==============================] - 684s 36ms/step - loss: 0.1394 - binary_crossentropy: 0.1394 - acc: 0.9472
Epoch 2/2
18750/18750 [==============================] - 682s 36ms/step - loss: 0.0420 - binary_crossentropy: 0.0420 - acc: 0.9866
```

Fig. 9. LSTM Accuracy

Fig 9. gives the information of LSTM algorithm which givesaccuracy of 98.66%, loss -0.042, binary crossentropy -0.042.

–

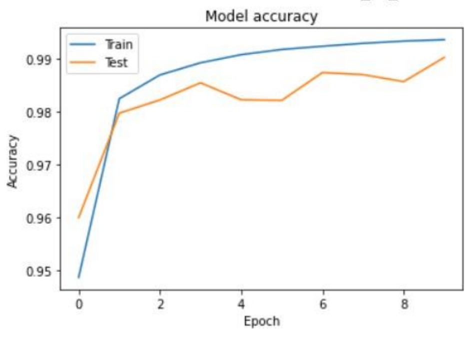


Fig. 10. Plot training and validation accuracy values.

The above graph shows the relation between model accuracyand epoch of LSTM.

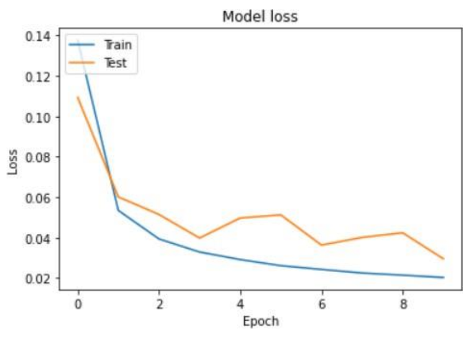
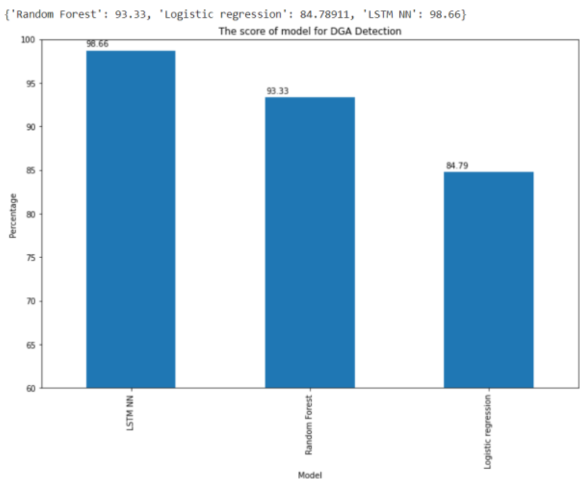


Fig. 11. Plot training and validation loss values

The above graph shows the relation between model loss andepoch of LSTM.



```
{'Random Forest': 93.33, 'Logistic regression': 84.78911, 'LSTM NN': 98.66}
```

Fig. 12. The score of model for DGA Detection

Fig 12. shows the accuracy of Different Algorithms in which LSTM shows highest accuracy.

IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 8, August 2021

DOI: 10.17148/IARJSET.2021.8844

CONCLUSION

Detecting DGAs is a grand challenge in security areas. Blacklisting is good for handling static methods. However, DGAs are usually used by an attacker to communicate with a variety of servers. They are dynamic, so simply using the blacklisting is not sufficient for detecting a DGA. The dynamically changing nature of the malicious domains needs to be addressed with an advanced system capable of detecting the history of the domain. The proposed machine learning framework consists of a feature extractor, a machine learning model is built for classification and prediction model. By summarizing the work we have done, the graph of each model's performance has been generated. It can be known thatthe Long Short-Term Memory neural network and Random Forest model have the best performance in DGA Detection. From the process of building our model, LSTM has the best performance but it will need a lot of time to train. However, for the Random Forest part, it only needs a few times totrain the model and it can classify a domain as soon as possible. As the size of the data increases, the performance of the machine learning model decreases. For that We will implement a deep learning model to handle the enormous amount of data and perform the classification, which has a better performance than the machine learning algorithms. Based on our extensive experiments on the real-world feed, we have shown that the proposed framework can effectively extract domain name features as well as classify, and detect domain names where it belongs.

REFERENCES

1. L. I. U. Wu, L. I. U. Ke, R. E. N. Ping, and D. Hai-Xin," Study and Detection of malware based on behavior," no. 60203044, pp. 39–42, 2011
2. J. Demme et al.," On quality counter on online malware detection feasibility," Proc. Int. Int Symp. Symp. Software. Computer. Architecture, pp. 559–570, 2013
3. D. R. Ellis, K. S. Attwood, J. G. Aiken, and S. D.Tenaglia," A worm identification strategy," WORM' 04-Proc.2004 Activities of the ACM. Rapid Malcode, pp. 43–53, 2004.
4. M. I. Gorelik, N. Abu-Ghazaleh, and D. Ozsoy, C. Donovick. Ponomarev," Malware Adware processors: an active online malware detection system," IEEE 21st Int 2015. Symp.Symp. Strong Perform. Software. Computer. Archit. 2015 HPCA, pp. 651–661, 2015.
5. R. Sharp," Malware Introduction," Netw. Safe. Secure. Laboratory check, pp. 331–363, 2015.
6. J. Blount, Tauritz, and Tauritz, and S. A. Mulder," Adap-tive rule-based malware detection using training classification systems: concept proof," Proc. Int. Int. Software. Computer. Softw. Appl. For instance, pp. 110–115, 2011.
7. K. Xiong, "Multiple priority customer service guarantees in cluster computing," in Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS), May 2009, pp. 1–12.
8. K. Xiong, Resource Optimization and Security for CloudServices. Hoboken, NJ, USA: Wiley, 2014.
9. K. Xiong. (2008). Resource Optimization and Se- curity for Distributed Computing. [Online]. Available: https://repository.lib.ncsu.edu/handle/1840.16/3581
10. M. Berman et al., "GENI: A federated testbed forinnovative network experiments," Comput. Netw., vol. 61, pp.5–23, Mar. 2014.
11. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu,"A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf.Knowl. Discovery Data Mining (KDD). Portland, OR, USA: AAAI Press, 1996, pp. 226–231. [Online]. Available: http://dl.acm.org/citation.cfm?id=3001460.3001507
12. E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," ACM Trans. Database Syst., vol. 42, no.3, pp. 19-1–19-21, Jul. 2017, doi: 10.1145/3068335.
13. 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.
14. K. Xiong and X. Chen, "Ensuring cloud service guar- antees via service level agreement (SLA)-based resource allo- cation," in Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst. Workshops, ICDCS Workshops, Jun./Jul. 2015, pp. 35–41.
15. T. Chin and K. Xiong, "Dynamic generation con- tainment systems (DGCS): A moving target defense ap- proach," in Proc. 3rd Int. Workshop Emerg. Ideas TrendsEng. Cyber-Phys. Syst. (EITEC), Apr. 2016, pp. 11–16, doi: 10.1109/EITEC.2016.7503690.
16. K. Sornalakshmi," Detection of DoS attack and zero day threat with SIEM," in Proc. IEEE Int. Conf. Intell. Comput. Control Syst. (ICICCS), Jun. 2017, pp. 1–7.
17. S. Yadav and A. L. N. Reddy, "Winning with DNS fail-ures: Strategies for faster botnet detection," in Proc. Int. Conf. Secur. Privacy Commun. Syst. Berlin, Germany: Springer,2011, pp. 446–459.
18. S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1663–1677, Oct. 2012.
19. F. Guo, P. Ferrie, and T.-C. Chiueh, "A study of the packer problem and its solutions," in Proc. Int. Workshop Recent Adv. Intrusion Detection. Berlin, Germany: Springer, 2008, pp. 98–115.
20. T. Holz et al., "Measurements and mitigation of peer- to-peer-based botnets: A case study on storm worm," in Proc. LEET, vol. 8, no. 1, 2008, pp. 1–9.
21. Suraj S. Bhoite; Swapnali K. Londhe. "Aspect Based Online Sentiment Analysis Product Review and Feature Using Machine Learning". International Research Journal on Advanced Science Hub, 3, Special Issue 7S, 2021, 54-59.
22. Madhamsetty Charitha; Nagaraj G Cholli. "Big Data Analysis and Management in Healthcare". International Research Journal on Advanced Science Hub, 3, Special Issue 7S, 2021, 42-53.
23. Toomula Srilatha; Jyothi Sree C.. "Survey on Plant Diseases Prediction using Machine learning for better Crop Yield". International Research Journal on Advanced Science Hub, 3, Special Issue 6S, 2021, 1-5.
24. Salma Begum; Sampuma P.. "A Study on growth in Technology and Innovation across the globe in the Field of Education and Business". International Research Journal on Advanced Science Hub, 3, Special Issue 6S, 2021, 148-156.
25. Krithika G K; Karthik S; Kowsalya R; Alfred Daniel J; Sangeetha K. "Driver Alert System Using Deep Learning and Machine Learning". International Research Journal on Advanced Science Hub, 3, Special Issue ICARD-2021 3S, 2021, 120-123. doi: 10.47392/irjash.2021.078