International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET21858

Machine Learning Techniques for Malware Detection

Harsha A K*[1], Thyagaraja Murthy A[2]
*[1] Post Grad. Student, Department of Electronics and Communication Engineering, JSS Science and Technology University, Mysore, Karnataka, India
[2]Associate Professor, Department of Electronics and Communication Engineering, JSS Science and Technology University, Mysore, Karnataka, India

Article Info
Volume 8, Issue 5
Page Number : 70-76

Publication Issue :
September-October-2021

Article History
Accepted : 03 Sep 2021
Published: 14 Sep 2021

ABSTRACT

The introduction of Transport Layer Security has been one of the most important contributors to the privacy and security of internet communications during the last decade. Malware authors have followed suit, using TLS to hide potentially dangerous network connections. Because of the growing use of encryption and other evasion measures, traditional content-based network traffic categorization is becoming more challenging. In this paper, we provide a malware classification technique that uses packet information and machine learning algorithms to detect malware. We employ the use of classification algorithms such as support vector machine and random forest. We start by eliminating characteristics that are highly correlated. We utilized the Random Forest method to choose only the 10 best characteristics from all the remaining features after eliminating the unnecessary ones. Following the feature selection phase, we employ several classification algorithms and evaluate their performance. Random forest algorithm performed exceptionally well in our experiments resulting in an accuracy score of over 0.99.

Keywords : Malware classification, Encrypted Traffic, Feature Selection, Random Forest, Support Vector Machine

I. INTRODUCTION

Transport Layer Security (TLS) is a cryptographic technology that is increasingly being used to secure web, messaging, and application data transfer on the Internet. TLS is already used to encrypt and encapsulate the contents of HTTPS, the Tor anonymizing network, and virtual private networks based on the OpenVPN protocol, preventing them from being monitored or modified in transit. TLS has been used by malware authors for the same purpose: to prevent defenders from detecting and preventing malware distribution and data theft. There has been a substantial surge in malware that uses TLS to hide its communications in the previous year. As a result, network monitoring devices must be able to detect malicious traffic hidden behind TLS encryption.

Traditional traffic prediction algorithms relied on port numbers and deep packet inspection (DPI). URL analysis and TLS fingerprinting are two more types of traditional traffic analysis. These approaches are ineffective in detecting malicious messages in TLS transmission. In the absence of the payload content, other features such as inter-packet arrival intervals, TCP headers, and flow direction can be utilized to identify the application and data encrypted by TLS. Modern traffic analysis techniques are built on the combination of these features with machine learning algorithms.

We describe a malware categorization approach that employs various machine learning classifiers in this work. To aid with categorization, we utilise a comprehensive feature selection and removal procedure. The amount of features in the benchmark dataset is decreased to improve the performance and efficiency of our classification models.

The remainder of the paper is organised as follows. The second section discusses similar work on encrypted traffic analysis. Section III discusses our research study's recommended approach. We discuss the dataset selected for this research. We also go through the feature selection and removal process. Section IV presents the results of the experiments. In Section V, we summarise our findings and make recommendations for further study.

II. LITERATURE REVIEW

Traditional approaches to traffic classification, such as those based on port numbers and payloads, are usually based on packet content analysis. However, as the general public's security awareness develops, so does the usage of encryption techniques for communication. The packet payloads are now encrypted, making prior methods useless. As a result, how to identify encrypted malware has become a research priority in the field of network security.
M.J de Lucia et al. [1] proposed a method for identifying suspicious communication using convolutional neural networks and a support vector machine Their SVM model outperformed the CNN model, and they used an approach that needed very little feature engineering. Shen et al. [2] provided a systematic technique for improving feature selection for successful encrypted traffic categorization. Yu et al. [3] presented a neural network-based approach for detecting encrypted malicious communications. During their investigation, they discovered that the imbalance in the dataset had an effect on the identification of encrypted traffic. Thaseen et al. [4] proposed a method for detecting network intrusion using machine learning algorithms. Singh et al. [5] presented a comparative research that included the examination of HTTPS traffic and virus identification. They recommended fully investigating TLS information and DNS traffic in order to improve malware detection and analysis.

Priya et al. [6] proposed a system for identifying users' browsers and apps by monitoring network data in real time with unsupervised machine learning algorithms. Hou et al. [7] created a detection method for smart home systems based on edge computing using a support vector machine. When all webpages are from the same source, Shen et al. [8] devised a technique for fingerprinting them based simply on packet length information. They achieved an accuracy of up to 91.6 percent using the k-NN technique. Dong et al. [9] proposed using the k-Nearest Neighbour method, a machine learning classification approach, to categorise video traffic. Conti et al. [10] described a method for detecting user activities by analysing an Android device's encrypted network. Using the dynamic time warping approach, they computed the sequence of data packets and extracted features. A random forest model was used for classification. Wang et al. [11] created a method for measuring the distance between packet sequences in order to achieve website fingerprinting.

III. PROPOSED METHODOLOGY

This section offers an overview of our work. Following the exploratory analysis of the dataset, we did comprehensive and rigorous feature pre-processing, feature selection, and dataset reduction. We used five alternative machine learning classifiers after selecting the needed characteristics, and based on the performance metrics, we summarised which model is best suited for encrypted traffic categorization. We further improved the performance of our models using grid search to tune the hyper parameters. We evaluate classifier performance using a variety of metrics, including AUC, Precision, Accuracy, F1-Score, and Recall.

A. Dataset

We utilised the MTA-KDD'19 Dataset developed by Ivan Letteri et al. [12] to perform this study. This dataset was created by grouping packets with the same source address in each pcap in segments, even if they were not time-stamped. As a result, we may see traffic from a new perspective, one that takes into account the hosts participating in the connection rather than simply the packets themselves. There are 33 features in this data collection. The dataset's creators carefully selected each of these characteristics.

B. Feature Elimination

We investigate the effect of preserving just important information that can increase prediction accuracy and eliminating those that are irrelevant and may harm model performance. We reduce model complexity by decreasing the number of features. This can aid in the prevention of overfitting. We selected a correlation threshold of +0.8 to eliminate strongly linked characteristics from our dataset. There were many groupings of characteristics that were correlated. We eliminated 11 features owing to multicollinearity as a result of this. The highly correlated features are shown in table 1.

TABLE I

HIGHLY CORRELATED FEATURES

S. No	Feature 1	Feature 2	Correlation
1.	UDPoverIP	TCPoverIP	0.985633
2.	UDPoverIP	DNSoverIP	0.877538
3.	AckFlagDist	FlowLEN	0.957587
4.	AckFlagDist	PshFlagDist	0.931598
5.	AckFlagDist	SynFlagDist	0.810665
6.	AvgLen	AvgWinFlow	0.947299
7.	AvgLen	MaxLen	0.858070
8.	AvgLen	FlowLen	0.850807
9.	AvgLen	StdDevLen	0.837223
10.	DeltaTimeFlow	MaxIAT	0.945212
11.	NumCon	NUmIPdst	0.930952
12.	AvgIAT	AvgIATrx	0.875336
13.	FinFlagDist	SynFlagDist	0.827972

We then used the random forest method to rank characteristics in order to focus on the most relevant ones. The Random Forest method ranks a node's Gini impurity using tree-based techniques. The highest impurity is found at the root, whereas the lowest impurity is found at the ends of the branches. A tree may also compute how much of the observed impurity in the tree is decreased by each feature once it has been trained. By trimming the tree, we may generate a subset of the most essential characteristics. Figure 1 depicts the top ten features in our dataset in terms of feature significance as determined by the Random Forest method.

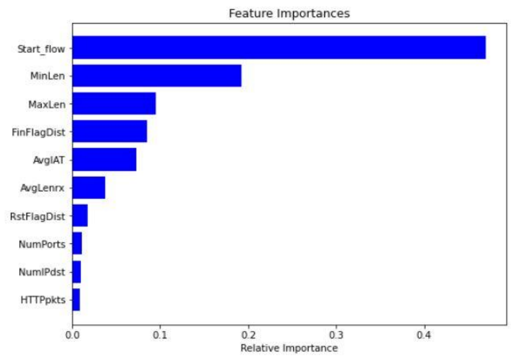

Figure 1: Top 10 features based on feature importance

C. Evaluation Metrics

Multiple evaluation metrics may be used to assess the performance of machine learning classifiers. In our investigation, we employed the following metrics: precision, accuracy, recall, and F1-score. These are based on the confusion matrix, which may be generated by considering the predicted values of the algorithm: true positives, true negatives, false positives, and false negatives.

D. Classifiers Used

1) Decision tree:

The decision tree method is represented as a tree, with nodes representing attributes, branches representing decisions (rules), and leaf nodes representing outcomes. We discovered that our first effort at modelling a decision tree algorithm was prone to overfitting. As a result, hyper parameter adjustment was required. We set max features, minimum samples split, and tree depth to 6, 0.2, and 5, respectively, after hyper parameter tweaking using grid search. This hyper parameter tweaking procedure resulted in a more efficient model. The training set's accuracy was 0.8967, whereas the testing set's accuracy was 0.8874.

2) Naïve Bayes:

This is a classification method based on the Bayes Theorem. It is predicated on predictor independence. A Naive Bayes classifier assumes that the existence of one feature in a class has no impact on the presence of any other feature in the class. These models are simple to build and useful for very large data sets. It is called Naive because it believes that the existence of one feature has nothing to do with the presence of others. When compared to the random forest model, this classifier performed poorly with our dataset. The accuracy of the training set was 0.8236, whereas the accuracy for the testing set was also 0.8212.

3) Random forest:

Random forest is an ensemble learning approach that combines multiple decision trees or weak learners to get more precise and dependable outcomes. A random forest's hyper parameters are nearly equal to those of a decision tree. Random Forest increases the model's unpredictability when 'growing' these trees. The graphs below might assist you in determining the appropriate values for the Random Forest classifier's hyper parameters. N-estimator denotes the number of trees to be built before computing the maximum voting or prediction averages. A larger number of trees improves performance but uses more resources. The maximum number of features, n estimators, minimum sample split, and tree depth were set to 5, 25, 0.2, and 10 correspondingly. The random forest classifier beat the decision tree classifier, and hyper parameter tweaking enhanced the classifier's performance even more. The training set's accuracy was 0.9950, whereas the testing set's was 0.9927.

4) Support vector machine:

The support vector machine algorithm finds a hyperplane in an n-dimensional space, where n denotes the number of distinguishing features between data points. Several such hyperplanes might be utilised to split data point groupings. The aim is to find the plane that has the maximum distance between data points from both classes. Hyperplanes are the boundaries that help in data categorization. Support vectors, on the other hand, are data points that are closest to the hyperplane and have an influence on its direction and location. Grid search was utilised to fine-tune the support vector machine classifier's hyper parameters. Grid search tries all possible dictionary value combinations and evaluates the model for each one using the cross-validation method. The best values for ‘C,' ‘gamma,' and ‘kernel' that resulted were 1000, 0.1, and ‘rbf,' where ‘rbf' stands for the Radial Bias Function kernel. When compared to the random forest model, this classifier did rather well with our dataset. The training set had an accuracy of 0.9247, while the testing set had an accuracy of 0.9264.

IV. RESULTS AND DISCUSSION

This section summarises the results of our experiment. We utilised four machine learning classifiers on the dataset: Random Forest (RF), Support Vector Machine (SVM) with the radial bias function kernel, Naive Bayes classifier, and Decision Tree (DT). Table 2 and 3 summarise the comparative metrics of the algorithms used in this study. The Random Forest classifier outperformed all other models with an accuracy of 0.9927 percent. The SVM classifier provided extremely strong results as well, with performance metric values greater than 0.9. The Nave Bayes model performed the lowest of the five, with comparably low accuracy and precision scores.

TABLE III

FINAL RESULTS (ACCURACY & PRECISION)

S.No	Algorithm	Accuracy	Precision
1	Decision Tree	0.8874	0.9811
2	Naïve Bayes	0.8212	0.7889
3	Random Forest	0.9927	0.9941
4	SVM	0.9264	0.9075

TABLE IIIII

FINAL RESULTS (RECALL & F1-SCORE)

S.No	Algorithm	Accuracy	Precision
1	Decision Tree	0.8018	0.8824
2	Naïve Bayes	0.9021	0.8417
3	Random Forest	0.9921	0.9931
4	SVM	0.9580	0.9321

V. CONCLUSION AND FUTURE SCOPE

As the volume of encrypted communication increases rapidly, detecting malware in traffic which is encrypted has become a difficult and complex challenge. Traditional content-based network traffic categorization is becoming difficult due to the rising use of encryption. In this paper we propose a malware classification technique by using multiple machine learning classifiers. We use an extensive feature selection and elimination process to further help in classification. We removed features based on correlation between them and finally selected only ten best features using the random forest algorithm. The number of features from the dataset is reduced to make our classification models perform better and make it more efficient. From the results we obtained, we conclude that the Random Forest algorithm performed the best compared to the other algorithms used for the classification of data packets as malicious or benign.

In future studies, we plan to use datasets which we will develop ourselves. Future work will also include deep learning models.

VI. REFERENCES

[1]. Lucia MJ, Cotton C. Detection of encrypted malicious metwork traffic using machine learning. IEEE Military Communications Conference 2019 (pp. 1-6). IEEE.

[2]. Shen M, Liu Y, Zhu L, Xu K, Du X, N. Guizani N. Optimizing feature selection for efficient encrypted traffic classification: A systematic approach. IEEE Network. 2020; 34(4):20-27.

[3]. Yu T, Zou F, Li L, Yi P. An encrypted malicious traffic detection system based on neural network. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery 2019 (pp. 62-70). IEEE.

[4]. Thaseen IS, Poorva B, Ushasree PS. Network intrusion detection using machine learning techniques. International Conference on Emerging Trends in Information Technology and Engineering 2020 (pp. 1-7). IEEE.

[5]. Singh AP, Singh M. A comparative review of malware analysis and detection in HTTPs traffic. International Journal of Computing and Digital Systems. 2021; 10(01):111-23.

[6]. Priya A, Nandi S, Goswami RS. An analysis of real-time network traffic for identification of browser and application of user using clustering algorithm. International Conference on Advances in Computing, Communication Control and Networking 2018 (pp. 441-445). IEEE.

[7]. Hou S, Huang X. Use of machine learning in detecting network security of edge computing system. International Conference on Big Data Analytics 2019 (pp. 252-256). IEEE.

[8]. Shen M, Liu Y, Chen S, Zhu L, Zhang Y. Webpage fingerprinting using only packet length information. International Conference on Communications 2019 (pp. 1-6). IEEE.

[9]. Dong Y, Zhao J, Jin J. Novel feature selection and classification of internet video traffic based on a hierarchical scheme. Computer Networks. 2017; 119:102–11.

[10]. Conti M, Mancini LV, Spolaor R, Verde NV. Analyzing Android encrypted network traffic to identify user actions. IEEE Transactions on Information Forensics and Security. 2016; 11(1):114-25.

[11]. Wang T, Cai X, Nithyanand R, Johnson R, Goldberg I. Effective attacks and provable defenses for website fingerprinting. Proceedings of the 23rd USENIX Conference on Security Symposium 2014 (pp. 143–57). ACM.

[12]. Letteri I, Penna G, Vita L, Grifa M. (2020). MTA-KDD'19: A dataset for malware traffic detection. Proceedings of the Fourth Italian Conference on Cyber Security 2020 (pp. 153-65). CEUR-WS.

[13]. S. Feghhi and D. J. Leith, "A Web Traffic Analysis Attack Using Only Timing Information," IEEE Trans. Info. Forensics and Security, vol. 11, no. 8, 2016, pp. 1747–59.

[14]. L. Xiao et al., "Cloud-Based Malware Detection Game for Mobile Devices with Offloading," IEEE Trans. Mobile Computing, vol. 16, no. 10, Oct. 2017, pp. 2742–50.

[15]. M. Shen et al., "Secure SVM Training Over Verticall Partitioned Datasets Using Consortium Blockchain for Vehicular Social Networks," IEEE Trans. Vehic. Tech., 2019, pp. 1–1.

[16]. A. Panchenko et al., "Website Fingerprinting at Internet Scale," Network and Distributed System Security Symp., 2016, pp. 21–24.

[17]. T. Wang et al., "Effective Attacks and Provable Defenses for Website Fingerprinting," Usenix Conf. Security Symp., 2014, pp. 143–57.

[18]. V. F. Taylor et al., "Robust Smartphone App Identification Via Encrypted Network Traffic Analysis," IEEE Trans. Info Forensics and Security, vol. 13, no. 1, 2018, pp. 63–78.

[19]. B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of TLS (without decryption)," Journal of Computer Virology and Hacking Techniques, vol. 14, no. 3, pp. 195–211, Aug. 2018.

[20]. M. Singh, M. Singh, and S. Kaur, "Issues and challenges in DNS based botnet detection: A survey," Computers & Security, vol. 86, pp. 28–52, Sep. 2019.

[21]. McGaughey, D., Semeniuk, T., Smith, R., & Knight, S. (2018, April). A systematic approach of feature selection for encrypted network traffic classification. In 2018 Annual IEEE International Systems Conference (SysCon) (pp. 1-8). IEEE.

[22]. Chang, Y., Li, W., & Yang, Z. (2017, July). Network intrusion detection based on random

forest and support vector machine. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (Vol. 1, pp. 635-638). IEEE.

[23]. Haripriya, L., & Jabbar, M. A. (2018, March). Role of Machine Learning in Intrusion Detection System. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 925-929). IEEE.

Cite this article as :

Harsha A K, Thyagaraja Murthy A, "Machine Learning Techniques for Malware Detection", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 5, pp. 70-76, September-October 2021. Available at doi : https://doi.org/10.32628/IJSRSET21858
Journal URL : https://ijsrset.com/IJSRSET21858