Machine Learning for Malware Detection in Network Traffic

Ayorinde Henry Omopintemi
University of Bradford
Bradford, United Kingdom
a.h.omopintemi@bradfor.ac.uk

Ibrahim Ghafir
University of Bradford
Bradford, United Kingdom
i.ghafir@bradford.ac.uk

Shadi Eltanani
Oxford Brookes University
Oxford, United Kingdom
seltanani@brookes.ac.uk

Sohag Kabir
University of Bradford
Bradford, United Kingdom
s.kabir2@bradford.ac.uk

Moemedi Lefoane
University of Bradford
Bradford, United Kingdom
m.lefoane@bradford.ac.uk

ABSTRACT

Developing advanced and efficient malware detection systems is becoming significant in light of the growing threat landscape in cybersecurity. This work aims to tackle the enduring problem of identifying malware and protecting digital assets from cyber-attacks. Conventional methods frequently prove ineffective in adjusting to the ever-evolving field of harmful activity. As such, novel approaches that improve precision while simultaneously taking into account the ever-changing landscape of modern cybersecurity problems are needed. To address this problem this research focuses on the detection of malware in network traffic. This work proposes a machine-learning-based approach for malware detection, with particular attention to the Random Forest (RF), Support Vector Machine (SVM), and Adaboost algorithms. In this paper, the model's performance was evaluated using an assessment matrix. Included the Accuracy (AC) for overall performance, Precision (PC) for positive predicted values, Recall Score (RS) for genuine positives, and the F1 Score (SC) for a balanced viewpoint. A performance comparison has been performed and the results reveal that the built model utilizing Adaboost has the best performance. The TPR for the three classifiers performs over 97% and the FPR performs < 4% for each of the classifiers. The created model in this paper has the potential to help organizations or experts anticipate and handle malware. The proposed model can be used to make forecasts and provide management solutions in the network's everyday operational activities.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**.

KEYWORDS

Machine learning, Malware Detection, Intrusion Detection, Malware Analysis, The Adaboost Algorithm, Random Forest, K-Nearest Neighbor Algorithm

ICFNDS '23, December 21–22, 2023, Dubai, United Arab Emirates
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0903-6/23/12
https://doi.org/10.1145/3644713.3644804

ACM Reference Format:
Ayorinde Henry Omopintemi, Ibrahim Ghafir, Shadi Eltanani, Sohag Kabir, and Moemedi Lefoane. 2023. Machine Learning for Malware Detection in Network Traffic. In *The International Conference on Future Networks and Distributed Systems (ICFNDS '23), December 21–22, 2023, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3644713.3644804

1 INTRODUCTION

Numerous electronic equipment has bad experiences alteration by malware in the digital age. Malicious software that is created with the intention of harming a victim is where the name malware originates. Malware can infiltrate networks, infect computers and other smart devices, steal sensitive data, damage vital infrastructure, and more [22]. These programmes include malware such as ransomware, rootkits, worms, spyware, bots, and viruses. According to [21] IT services claims that in only one year, one billion emails were exposed, impacting one in five internet users, and resulting in data breaches that cost organisations, on average, $4.35 million in 2022. The first half of 2022, there were about 236.1 million ransomware assaults worldwide. In 2021, the accounts of one in two internet users in America were compromised. Malware attacks are becoming more complicated over time, despite improvements in detection, proper family class classification, and continual evolution, malware continues to be a serious threat to the internet [19]. Malware has also increased the risk of sophisticated attacks, such as multi-stage attacks [1, 2, 15, 16] and Distributed Denial of Service (DDoS) attacks [8, 9], which have been a serious threat in recent years. To mitigate the risk of cyber attacks, Intrusion Detection Systems (IDSs) have been used to monitor the network traffic [4, 11, 12].

The ability to identify virus in a computer allows the development of malware prevention or anti-malware solutions that include a unique signature to recognise the infection [6]. Depending on why it was created, malware can take on a variety of forms, including ransomware which is intended to extort money, and spyware which is used to spy on people. Human intelligence continues to be a key component in the construction of many tools and approaches, even though several Machine Learning (ML) methods have been developed for anomaly detection in network traffic [14, 17, 24]. Particularly, the hand-crafted characteristics play a significant role in the classic ML-based malware analysis techniques. According to [18] these characteristics describe what computer security professionals view as the most important inherent traits of malware.

However, the feature engineering procedure is time-consuming, and the manually created features that are created are task specific and frequently arbitrary to individual judgement. When it comes to malware detection, traditional ML has found it to be especially effective. Most ML-based solutions, on the other hand, depend substantially on the experience, level of competence, and breadth of the subject knowledge of the security professionals to define the features to characterised malware manually. The study in [20] generate representations of malware that are less reliant on human expertise that is important but unresolved challenge.

This paper proposes a machine-learning-based approach for malware detection, with particular attention to the Random Forest (RF), Support Vector Machine (SVM), and Adaboost algorithms. The remainder of this paper is organised as follows. Section 2 presents the related to malware detection, section 3 describes the proposed methodology, section 4 shows the evaluation results, Section 5 presents a performance analysis of the utilised ML algorithms and Section 6 concludes the paper.

2 RELATED WORK

The malware threat and the hazards associated with online defense are getting harder to defend against in a world where technology is developing at an accelerating rate. A piece of software known as malware, or more generally known as a computer virus, exists solely to enter a person's computer and harm both the machine and documents found. The malware is implemented to provide unauthorised login. This kind of security breach, often known as Trojan and can take over any victim's computer [3]. The ongoing growth of malware and cloud-safety holes raises the query of what method we preserve to defend asset. This research involves several applications, the use of system knowledge built on malware recognition techniques are evaluated as viable fixes [13].

For classification issues like malware detection, system learning has many things to offer. In order to assist restricting and avoiding the harm that these infections wreak, the result of this issue entails applying machine Learning algorithms to identify action and common actions of viruses. The issue with malware identification, according to [5], is the capability to identify malware that is complicated and does not behave in a predictable way. A wide range of viruses must be examined by machine learning models before usage to improve virus detection. This issue demonstrates the need for in-depth and exhaustive testing, particularly in dire circumstances including harmful and destructive malware intrusions. This paper examines the method and also reveals 13 varieties of instances when machine learning applications might be used to benefit other fields. Some of the most common varieties of malware are discussed through this paper.

According to [7] Virus are pieces of code that infect a computer system and keeps reproducing until it corrupts the structure or the document on the computer. Worm: This is a computer program that replicates itself to move between computers on a network connection and infect further systems [23] Trojan Horse is harmful programme that has been disguised to look innocent or legal to convince consumers that it is secure to download. Once downloaded, the programme shows its actual nature and has the potential to do serious harm [10]. A Run time malware programme will either steal data or encrypt it so the user cannot access it. The user will then be coerced by a third party to make a payment or exchange of some kind to get their information back Viruses are designed to infect the computers of an organisation or different or group of people to either corrupt their files or seize control of the system and steal or hold for ransom personal information, such as financial information or identification. For companies that handle a lot of client information, this kind of breach can be very harmful. Because malware is always changing, system knowledge set of rules can be an extremely useful instrument for identifying the arrangements and developments that precede these incidents. Machine learning is used in these circumstances to develop a projected result for incident to happen by using software that employs various algorithms.

The signature-based approach deals with a series of bytes that can be retrieved from software and serve as an identification. It is possible to identify viruses that are communicated from the database through the signature signal. Due to this, the technique frequently fails to counter brand-new or evolving threats. This solution necessitates human involvement in the registration of the signature and is useless against certain kinds of malware. In order to categorise a programme as harmful when a signature-less approach is utilized, some predefined guidelines must be taken into account. A set of suspected elements could be defined like "assembly proven to unusual intention, document permission changed completely," "unnecessary archive and changes made," etc. After that, establish a threshold, and if any programme activates the attributes above the defined threshold, it can be regarded as damaging to the computer. Algorithm must assess the size of the datasets before undergoing training or applying determination functions across the dataset's attributes. Most malware detection methods used by antivirus software producers depend on the use of signature malfunction recognition.

This signature has accurate functionality to detect each malware or virus in a real time environment. to be able to detect new malware programmes. Anomaly detection can find new threats; however, it has a high false alarm rate. Malware investigation is divided into fixed or non-fixed functionality depending on the malware's current condition, To do a static analysis on an executable file, the programme must not be run Finding a source's style and profiling the code flow are two benefits of static analysis. Numerous evaluations indicate that malware is rapidly expanding and harming the vast array of risks. The most recent operating systems all have flaws that allow malware to proliferate, and many Web browsers with built-in virtual private network (VPN) services send highly unsafe data from distant servers. Networks are breached by malicious code used by attackers with inexperienced users through serious cyberthreats.

3 PROPOSED METHODOLOGY

The utilised ML-based approach is illustrated in Figure 1. Stage 1 in the ML process involves gathering data to train a classifier for an expected design and stage 2 employs geographical correlations and expert knowledge to reduce parameters and weight reusability. Reducing predictions and boosting performance, stage three of the process entails fitting extracted features, this enables algorithms to identify patterns and relationships. Stage 4 implements the SVM, RF, and AdaBoost algorithms to train the malware prediction model. These algorithms differentiate between malicious and benign software, this phase enables the trained model to recognize possible malware occurrences that are essential for real-time application. Once the intrusion detection result is present at step 6, the algorithm learns to modify weights as necessary to detect malware or non-malware. Step 7 evaluates the performance of the utilised ML algorithms. The model is developed in Python using Anaconda. Machine learning algorithms Random Forest, K-Nearest Neighbors, and Adaboot Algorithm were utilized for classification. Software utilized in this study included Jupiter Notebook in Python and Anaconda, which was used to extract features from the network traffic. The accuracy, precision, F1 score, and recall score were used as the standard evaluation parameters to assess the performance of each classification algorithm based on each classification and trained algorithm.

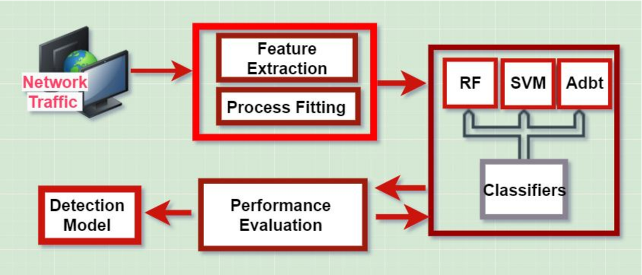

Figure 1: A ML-based Approach for Malware Detection.

A genuine positive is a link that is anticipated to be there when evaluating the link prediction algorithm. An actual missing link that is mistakenly identified as an existing link is known as a false positive. True missing links that are anticipated to be missing links are true negatives an actual present link that is misdiagnosed as a missing link is known as a false negative. A true positive is defined as a true present connection that is an inferred present link while evaluating the interpellator.

4 EVALUATION RESULTS

The malware detection method's performance was evaluated using the experimented benign dataset that was downloaded and tested from Kaggle. 10,593 malware samples were present in the dataset. 4238 are normal samples and 6355 are benign samples. The paper used the GPU version, a minimum of 500GB of RAM, 64-bit Ubuntu 16.04, and Python Tensorflow 1.9. TensorFlow offers a versatile framework for creating machine-learning models with hyperparameters. It uses multi-dimensional arrays to carry out operations. To hasten the categorization process, parallel execution is used. First, we chose 80% of the training data and 20% of the data for testing. Previously, several researchers suggested that an 80%–70% training data ratio is a better option for experimentation. Second, the research examines the tests with two distinct categorization sizes, namely 234x254 for (1=Malware) and 259x229 for (0=Non-Malware). Third, the research used four evaluation criteria for performance evaluation: precision, recall, F1 score, and accuracy. The number of malware samples categorized as false and true, respectively, was indicated by the number of True Positives (TPs) and False Positives (FPs). The number of benign samples labelled as true or false was indicated similarly by the number of True Negatives (TNs) and False Negatives (FNs). TPR (True Positive Rate) and FPR (False Positive Rate) are employed as performance indices in general evaluation standards to improve the detection accuracy of malware.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

The percentage of accurately detecting malware is calculated using equation (1), and the detection rate rises when large percentages are made public. FPR, or the percentage of malware false detection, is calculated using equation (2); lower percentages indicate accurate classification.

4.1 Performance Evaluation of Random Forest

Precision and Recall performance metrics were evaluated as shown in Figures 2 and 3, The performance of the model with 10,593 features shows ROC of 1.00 this indicates good and robust accuracy of the developed mode.

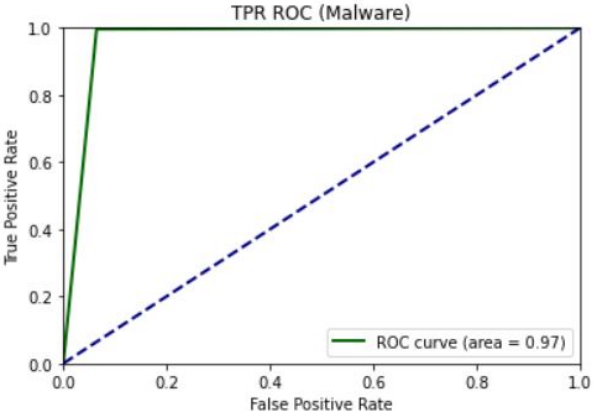

Figure 2: Random Forest Algorithm TPR Vs FPR

The ROC curve is shown in Figure 3 the diagonal line denotes the absence of any models. There is an equal amount of space above and below the diagonal, or 0.5. The best model is regarded as having the highest AUC value. A model's prediction is ideal when the ROC value is 1.0, and it is deemed to be poor when the ROC value is 0.5 or lower.

Figures 4 show the confusion matrix predicted indexes; the True Positive predicted for malware is 6,984, False Positive is 15, False Negative is 279, and True Negative is 3,261. This sums the 10,539 of the total datasets experimented. The use shows the trade-off between precision and recall and relates to how relevant the accuracy of each class is predicted.

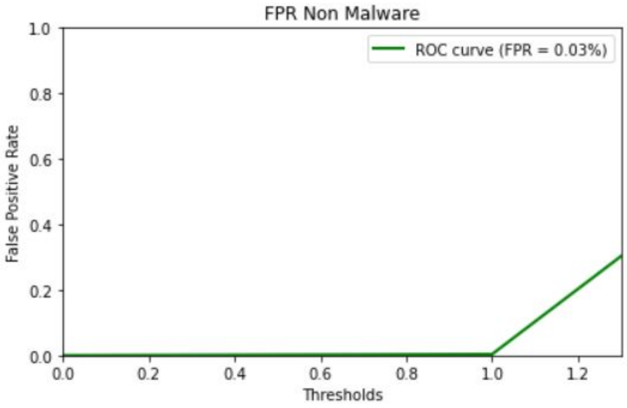

Figure 3: Random Forest Algorithm FPR Vs Threshold

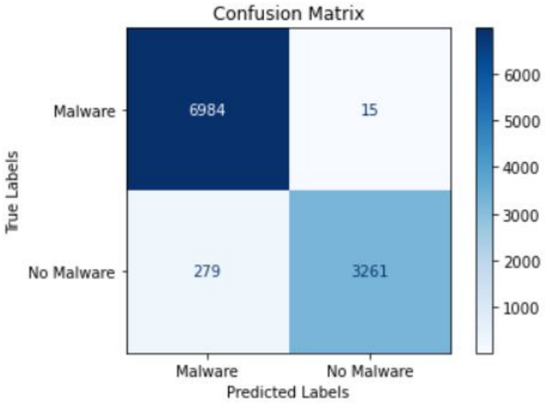

Figure 4: Random Forest Confusion Matrix

4.2 Performance Evaluation of KNN

Performance parameters for precision and recall were assessed as depicted in 5, 6 AUC shows good and reliable accuracy curve of over 98% of the model validation performed on 10,593 features. Hence, the created model indicates good performance.

Figure 7 displays the expected indexes for the confusion matrix; the predicted True Positive for malware is 6,931, False Positive is 68, False Negative is 197, and True Negative is 3343. The 10,539 experimental datasets are added up in this. The application demonstrates the relationship between the recall and precision and relates to the applicability of each class of the KNN model accuracy.

Figure 7 displays the expected indexes for the confusion matrix; the predicted True Positive for malware is 6,931, False Positive is 68, False Negative is 197, and True Negative is 3343. The 10,539 experimental datasets are added up in this. The application demonstrates the relationship between the recall and precision and relates to the applicability of each class of the KNN model accuracy.

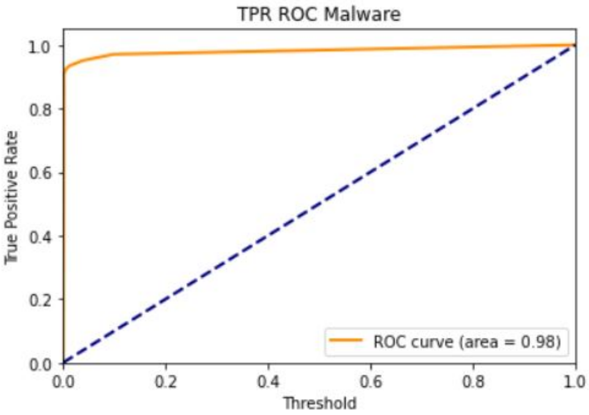

Figure 5: KNN FPR Vs Threshold

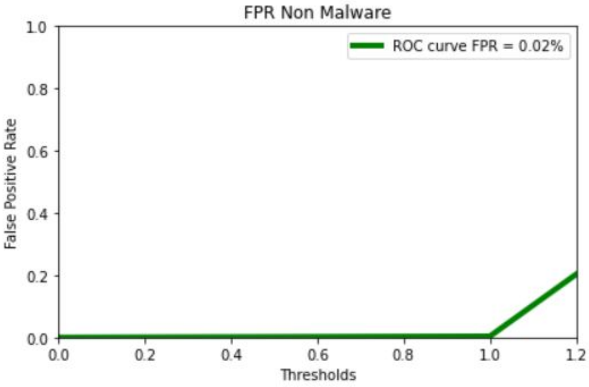

Figure 6: KNN FPR Vs Threshold

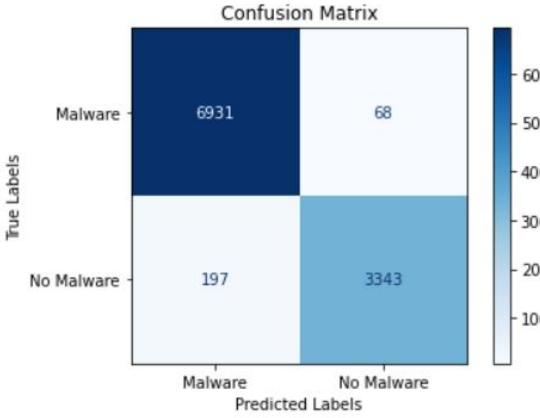

Figure 7: KNN Confusion Matrix

4.3 Performance Evaluation of AdaBoost

The adaBoost classifier shown in Figures 8 and 9 has a True Positive Rate (TPR) of almost 100% and a False Positive Rate (FPR) of almost

0%, indicating that the model correctly identified the particular task. For many classification tasks, the optimum result is a TPR of 100% and an FPR of 0%, which shows that the model is functioning flawlessly with respect to that dataset.

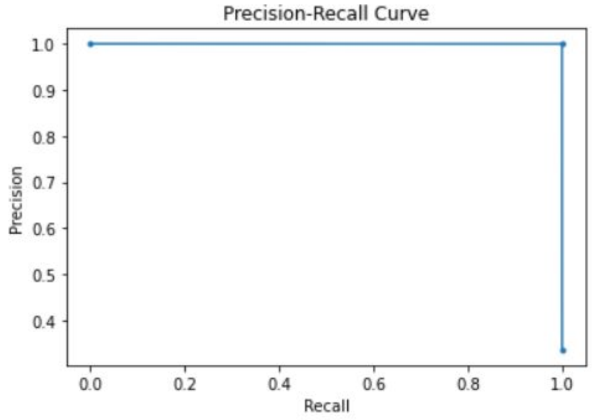

Figure 8: Adaboost Algorithm Precision Vs Recall

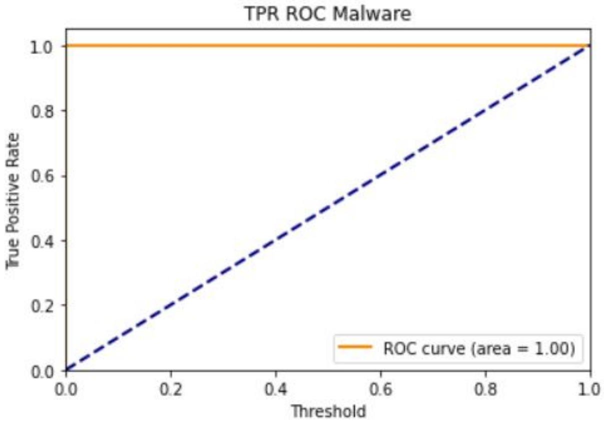

Figure 9: Adaboost Algorithm TPR Vs Threshold

The 100% ROC shows good and reliable accuracy of the created model in the performance of the model with 10,593 features. True Positive Rate (TPR) and False Positive Rate (FPR) testing results are illustrated in Figure 8 of the Adaboost classifiers' confusion matrix. This statement suggests that TPR and FPR are derived from or represented by the confusion matrix, showing 1 as a false alarm rate.

5 A PERFORMANCE COMPARISON OF RF, KNN AND ADABOOST ALGORITHMS

The table compares the three classification techniques with the XGBoost algorithm using a dataset of 10,539 attributes, with 80% being used for training and 20% for testing. Table 5.1 includes a performance summary for various methods. KNearest Neighbour (KNN), and Random Forest perform less to XGBoost Algorithm.

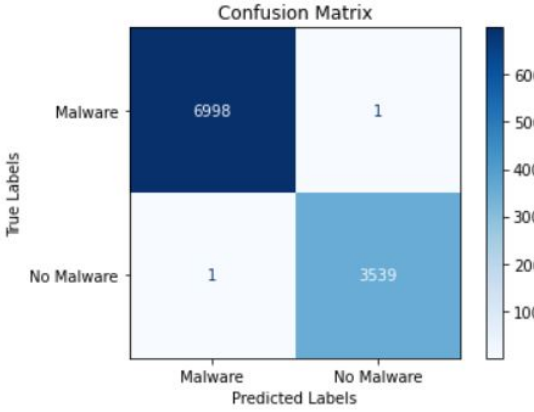

Figure 10: Adaboost Confusion Matrix

Table 1: A Performance Comparison of RF, KNN and Adaboost algorithms

Experiment	Random Forest	KNN Algorithm	Adaboost
Precision	0.97210	0.97485	0.99980
Accuracy	0.97295	0.97494	0.99982
F1 Score	0.97211	0.97488	0.99981
Recall	0.97182	0.97473	0.99981
FPR	0.03	0.02	0.01

The True Positive Rate for Adaboost and KNN classifiers perform the best. While RF classifier achieves the minimum result for the True Positive rate, AdaBoost algorithm achieves the minimum FPR. Overall, the results show that the KNN classifier is the second-best algorithm and that AdaBoost came out as the best algorithm.

6 CONCLUSION

This work presents the effectiveness of different machine-learning classifiers for malware detection in network traffic. The three machine learning algorithms experiment on a total of 10,539 malicious and non-malware data samples. These algorithms' performance is assessed with different evaluation metrics such as footing the model Accuracy, Precision, F1 Score, Recall, AUC, False Positive Rate, and True Positive Rate. The evaluation metrics perform well over 95%, including the True Positive Rate (TPR) under the ROC, while the False Positive Rate (FPR) of each of the algorithms is less than 4%. The Adaboost algorithm has demonstrated the highest performance among all other classifiers. After analysing the experimental data, it can be said that the Adaboost algorithm outperforms all other algorithms in terms of Accuracy, Precision, F1 Score, Recall, AUC False Positive Rate, and True Positive Rate when it comes to detecting malware.

REFERENCES

[1] Francisco J Aparicio-Navarro, Timothy A Chadza, Konstantinos G Kyriakopoulos, Ibrahim Ghafir, Sangarapillai Lambotharan, and Basil AsSadhan. 2019. Addressing multi-stage attacks using expert knowledge and contextual information. In *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 188–194.

[2] Franciso J Aparicio-Navarro, Konstantinos G Kyriakopoulos, Ibrahim Ghafir, Sangarapillai Lambotharan, and Jonathon A Chambers. 2018. Multi-stage attack detection using contextual information. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 1–9.

[3] Omar Bawazeer, Tarek Helmy, and Suheer Al-Hadhrami. 2021. Malware detection using machine learning algorithms based on hardware performance counters: analysis and simulation. In *Journal of Physics: Conference Series*, Vol. 1962. IOP Publishing, 012010.

[4] Andrew Carlin, Mohammad Hammoudeh, and Omar Aldabbas. 2015. Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. *International Journal of Advanced Computer Science and Applications* 6, 6 (2015).

[5] Robertas Damaševičius, Algimantas Venčkauskas, Jevgenijus Toldinas, and Šarūnas Grigaliūnas. 2021. Ensemble-based classification using neural networks and machine learning models for windows pe malware detection. *Electronics* 10, 4 (2021), 485.

[6] Sidney ML de Lima, Heverton K de L Silva, João H da S Luz, Hercília J do N Lima, Samuel L de P Silva, Anna BA de Andrade, and Alisson M da Silva. 2021. Artificial intelligence-based antivirus in order to detect malware preventively. *Progress in Artificial Intelligence* 10, 1 (2021), 1–22.

[7] Sidney ML de Lima, Heverton K de L Silva, João H da S Luz, Hercília J do N Lima, Samuel L de P Silva, Anna BA de Andrade, and Alisson M da Silva. 2021. Artificial intelligence-based antivirus in order to detect malware preventively. *Progress in Artificial Intelligence* 10, 1 (2021), 1–22.

[8] Diab M Diab, Basil AsSadhan, Hamad Binsalleeh, Sangarapillai Lambotharan, Konstantinos G Kyriakopoulos, and Ibrahim Ghafir. 2019. Anomaly detection using dynamic time warping. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE, 193–198.

[9] Diab M Diab, Basil AsSadhan, Hamad Binsalleeh, Sangarapillai Lambotharan, Konstantinos G Kyriakopoulos, and Ibrahim Ghafir. 2021. Denial of service detection using dynamic time warping. *International Journal of Network Management* 31, 6 (2021), e2159.

[10] Ahmed Hashem El Fiky, Ayman Elshenawy, and Mohamed Ashraf Madkour. 2021. Detection of android malware using machine learning. In *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 9–16.

[11] Mohammad Hammoudeh, Gregory Epiphaniou, Sana Belguith, Devrim Unal, Bamidele Adebisi, Thar Baker, ASM Kayes, and Paul Watters. 2020. A service-oriented approach for sensing in the Internet of Things: Intelligent transportation systems and privacy use cases. *IEEE Sensors Journal* 21, 14 (2020), 15753–15761.

[12] Mohammad Hammoudeh and Robert Newman. 2015. Information extraction from sensor networks using the Watershed transform algorithm. *Information Fusion* 22 (2015), 39–49.

[13] Sushil Kumar et al. 2021. MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things. *Future Generation Computer Systems* 125 (2021), 334–351.

[14] Moemedi Lefoane, Ibrahim Ghafir, Sohag Kabir, and Irfan-Ullah Awan. 2021. Machine Learning for Botnet Detection: An Optimized Feature Selection Approach. In *The 5th International Conference on Future Networks & Distributed Systems* (Dubai, United Arab Emirates) *(ICFNDS 2021)*. Association for Computing Machinery, New York, NY, USA, 195–200.

[15] Moemedi Lefoane, Ibrahim Ghafir, Sohag Kabir, and Irfan-Ullah Awan. 2022. Multi-stage Attack Detection: Emerging Challenges for Wireless Networks. In *2022 International Conference on Smart Applications, Communications and Networking (SmartNets)*. 01–05. https://doi.org/10.1109/SmartNets55823.2022.9994027

[16] Moemedi Lefoane, Ibrahim Ghafir, Sohag Kabir, and Irfan-Ullah Awan. 2023. Latent Dirichlet Allocation for the Detection of Multi-Stage Attacks. In *The 24th International Arab Conference on Information Technology*. IEEE, 1–6.

[17] Moemedi Lefoane, Ibrahim Ghafir, Sohag Kabir, and Irfan-Ullah Awan. 2023. Unsupervised Learning for Feature Selection: A Proposed Solution for Botnet Detection in 5G Networks. *IEEE Transactions on Industrial Informatics* 19, 1 (2023), 921–929. https://doi.org/10.1109/TII.2022.3192044

[18] Ayorinde Henry Omopintemi, Promise Irebami Ayansola, and Kehinde Gbemisola 2022 Ogundijo. [n. d.]. Device Synchronization Using a Computerize Face Detection and Recognition System for Cyber security. ([n. d.]).

[19] Sheikh Shah Mohammad Motiur Rahman, Fatama Binta Rafiq, Tapushe Rabaya Toma, Syeda Sumbul Hossain, and Khalid Been Badruzzaman Biplob. 2020. Performance assessment of multiple machine learning classifiers for detecting the phishing URLs. In *Data Engineering and Communication Technology: Proceedings of 3rd ICDECT-2K19*. Springer, 285–296.

[20] Hemant Rathore, Soham Chari, Nishant Verma, Sanjay K Sahay, and Mohit Sewak. 2023. Android Malware Detection Based on Static Analysis and Data Mining Techniques: A Systematic Literature Review. In *International Conference on Broadband Communications, Networks and Systems*. Springer, 51–71.

[21] AAG IT Services. Year. *AAG IT Services*. https://aag-it.com/the-latest-ransomware-statistics/ Accessed: February 7, 2024.

[22] KV Uma and E Sharon Blessie. 2019. Survey on Android malware detection and protection using data mining algorithms. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on*. IEEE, 209–212.

[23] Xinning Wang and Chong Li. 2021. Android malware detection through machine learning on kernel task structures. *Neurocomputing* 435 (2021), 126–150.

[24] Yuan Zhang, Qinghai Yang, Sangarapillai Lambotharan, Konstantinos Kyriakopoulos, Ibrahim Ghafir, and Basil AsSadhan. 2019. Anomaly-based network intrusion detection using SVM. In *2019 11th International conference on wireless communications and signal processing (WCSP)*. IEEE, 1–6.