HBRP PUBLICATION

Advancement in Image Processing and Pattern Recognition
Volume 6 Issue 1
DOI: https://doi.org/10.5281/zenodo.7609730

Detection of Malware Using Machine Learning Algorithms

Tahir Naquash[1], Yusuf Ahmad[2], Satyam Singh[3], Shubhanshu Kumar[4], Yash Anjana[5]
[1]Professor, [2,3,4,5]Student
Department of CSE, HKBKCE, Bangalore, India

****Corresponding Author***
***E-mail Id** :- shubhanshubb@gmail.com*

ABSTRACT

Malware is becoming a major cybersecurity threat with increasing frequency every day. There are several ways to classify the new malware based on signatures or code present. Traditional approaches are not very effective against newly emerging Malware- samples. More and more antivirus software offers protection against malware, but zero-day attacks have yet to be achieved. We use machine learning algorithms to improve the mechanism and accordingly provide excellent experimental results. To do Traditional signature approaches also fail, but the newmalware does. This document defines malware and malware types as an overview, also defines new mechanisms that use machine learning algorithms, effective and efficient methods in classifying malware detection, and buildson existing research on malware detection. to introduce. Machine Learning Algorithms describes the main challenges faced in malware detection classification.

Keywords: *Malware, Malware Analysis, Static Analysis, Dynamic Analysis, Classification, Machine learning.*

INTRODUCTION

Malware word defined by Malicious Software. Malware is malicious code that compromises a user's system or computer and intentionally damages her computer by an attacker. Malware, viruses, trojans, backdoors, rootkits, ransomware, worms, botnets, spyware, adware, keyloggers, and variants of the form, such as, access the internet every day. According to AV Test Institute's studies [1],350,000 new malicious code and potentially unwanted applications are registered every day. Each malicious code is categorized into in terms of its behavior and is properly stored by these agencies, and malware statistics show that 847.34 million malicious codes were detected and recorded in 2018. and has been registered. Part of a past malware attack was a macro that Melissa embedded in a Word file. When the user opens it, the macro runs and resends the virus to her first 50 people in the user's address book. Designed by David L. Smith in 1999. Similarly, there have been several malware attacks in the past, including the My doom worm in 2004, Stuxnet in 2010, and Wannacry in 2017. This paper introduced the literature work of previous papers on malware detection classification using machine learning algorithms. Section 2 discusses malware analysis and types of malware analysis. Section 3 is all about his paper on classifying malware detection using machine learning algorithms.

MALWARE ANALYSIS

An analysis of the behavior, functionality, and impactof the Malware sample on your system. Defined as malware analysis. Signature-based, Behavior-based, and Memory-based Malware Analysis Analysis of Sample Malware Variants in Different Ways

Advancement in Image Processing and Pattern Recognition
Volume 6 Issue 1
DOI: https://doi.org/10.5281/zenodo.7609730

Types of Malware Analysis

Basically, there are generally three types of malware analysis, namely static malware analysis, dynamic malware analysis, and memory malware analysis.Static Malware Analysis.

Static Malware Analysis

Detecting or investigating malicious code without executing it is called static malware analysis. Signature-based malware analysis. Static Malware Analysis extracts static features such as metadata strings, code and import libraries and uses them in the feature extraction phase of feature selection or machine classification. In most cases, the input file types for malware static analysis should be exe, dll, documents, assembly code, bytecode, etc. From these file types static characteristics are extracted as output. Tools used for static analysis of malware include PEiD, ssdeep, pafish, Yara, strings, IDA Pro [2], OllyDbg [3], LordPE [4], and OllyDump[5].

Dynamic Malware Analysis

Malware analysis is the process of identifying the actions and features of malicious code while it is active. System calls, file activity, process activity, and network activity are examples of dynamic features. The dynamic features of harmful code are extracted using the dynamic malware analysis tool. CWSandbox [6], Anubis [7], Comodo Automated Analysis, and ThreatTrack are some of the available tools. Process Explorer [12], Process Monitor [13], Capture BAT [14] (for registry monitoring and file system monitoring), RegShot [15] (for system change detection for network monitoring), Wireshark [16] (for network monitoring), Process Hacker [17], and others are monitoring tools for dynamic malware analysis.

Analysis of Memory Malware

Memory malware analysis is the process of evaluating malicious code that has already been executed. Shared libraries, active processes, hook detection, network connections, links to rootkits, hidden artefacts, and code injection are some elements of memory analysis. Volatility, pin tools, Valgrind, and other memory analysis tools are available.

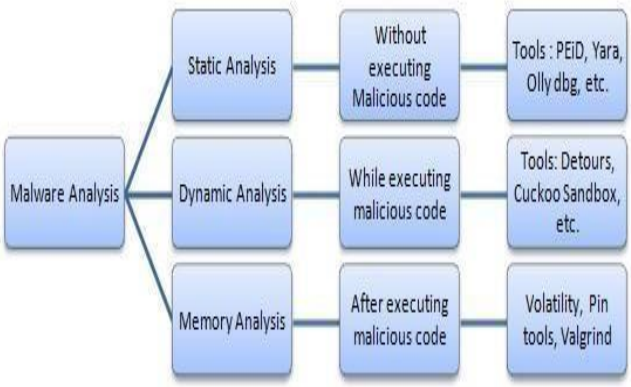

Fig. 1. *Types of Malware Analysis*

Detection of Malware Using ML Algorithms

Malware is categorised using a variety of machine learning techniques, including: The literature on machine learning methods for malware analysis is discussed here. A proposed architecture for malware detection was made by Gupta et al. in 2018 [18]. In this design, a dataset consisting his 200,000 files—0.5 million clean files and

DOI: https://doi.org/10.5281/zenodo.7609730

his 0.15 million malware samples—is first prepared. The malware samples were gathered from a variety of sites, including VXheaven, Notink, and VirusShare. After gathering malware samples, the Cuckoo sandbox is used to automatically analyse the acquired data set for malware. Python scripts are included in this sandbox to analyse the behaviour of any malware that is currently operating. The output is presented in JSON (JavaScript Object Notation) format. Apache Spark uses Python to extract static and dynamic functions from JSON reports. Following feature extraction, a 10-fold cross-validation is used to apply the classification model. On the basis of the parameters true positive rate (TPR), false positive rate (FPR), accuracy, false negative rate (FNR), and accuracy, a classification model is applied to the dataset. According to experimental findings, Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) all predict an accuracy of 89.
13% accuracy, 94.03% SVM, and 98.88% Random Forest. The best accuracy is provided by random forest, with low FPR and FNR. A methodology for performing data preparation and categorization is proposed by Cho et al. in 2016 [19]. (malware similarity). Malware samples are decreased in size during the preprocessing stage as far as they are classified into connected malware families. The classification procedure must take into account similarity calculation, sequence alignment, sequence refining, and behavioural monitoring. A high accuracy of 87% was achieved using 150 malware samples divided into 10 different malware variant families. Classification was conducted five times.combustion, etc. Self-organizing maps are used in 2018 [20] to distinguish between malicious and benign files, which minimises the overfitting processwhile training instances. Data is gathered using the VirusTotal API. Multiple classifiers are employed in this architecture for classification and are put to the test under various conditions. This method employs the Random Forest, BayesNet, MLP, and Support Vector Machine classifier models. First, all malware samples are assessed using 10-fold cross-validation, and Random Forest provides the best result with 98%accuracy. However, due to an overfitting issue, the accuracy of Random Forest drops by 12% when the Random Forest classifier is used on other datasets.
They provide a new Self-Organizing Feature Map (SOFM) technique, a classifier that uses the ANN approach, to solve this issue. The experimental findings reveal a 7% increase in accuracy over the prior model. Adapted from Razak et al. 21] conducted a bibliographic analysis study on his 4000 publications published between 2005 and 2015 usingrecords from the North American, Asian, and other continents found in the ISI Web of Science database. It introduces research and discusses research activity. Using the search term "virus," I located his 4,546 records at a Web of Science database spanning journals, books, book sections, and patents with 2,158 records of non-English articles, including the KCI-Korean Journal Database, Derwent Innovation Index,and SCiELO. Results are gathered from prestigious journals, highly referenced publications, subject areas, productivity, keyword counts, organisations, and authors. They conducted an analysis and came to the conclusion that academic research on malware is more widely published in North Americain the production of malware papers, whereas Asia places him second. A concise review of malware and malware analysis is given in Ray et al. 2016 [22]. They listed several malware varieties, such as worms, Trojan horses, viruses, spyware, backdoors, and rootkits. The two different kinds of malware analysis—static malware analysis and dynamic malware analysis—were also

DOI: https://doi.org/10.5281/zenodo.7609730

covered. If not, malware is dynamically and statically assessed when it is executed. Various techniques used in dynamic malware analysis include function call monitoring, function parameter analysis, information flow tracing, and command tracking. Anubis, CWSandbox, and Norman Sandbox are tools for malware analysis.

They demonstrated the superiority of dynamic malware analysis as a malware analysis technique. Research on malware analysis using machine learning algorithms is presented in Lodz et al. 2017 [23]. We'll start by introducing the analysis target, malware functions, and the machine learning algorithm's verification task for processing the malware analysis function. Then, we had a discussion about malware dataset-related issues and displayed research on datasets from different sources. The third area of research examines novel methods for the economics of malware analysis, with an emphasis on performance indicators including precision, speed of execution, and financial cost. For misclassifiers, Al-Ahmadi et al. 2018 [24] suggests a novel method. This is a three-step procedure. Pretreatment is the initial step, followed by extraction. Families of malware variants serve as the input for this step and are sourced from network traffic. Prior to moving on to flow coding, these variant families are rebuilt in network flow, where later flow coding extraction is used as input. At the start of the sequence extraction process is the second stage, profile extraction. Following n-flow mining of malware, binary similarity, Levenshtein distance, cosine similarity,and interflow distance to the similarity family are used to compare the similarity of flow values. The network behaviour obtained from the second stage is used to extract profiles. The third phase is developing and training a model for the function of profile extraction. For model classification and training, machine learning techniques KNN and random forests are employed. Finally, Malclassifier has a high accuracy rate of 95.5% when classifying malware families (F value). (2017) Khan et al[25] introduction of a malware detection framework (unwanted signatures). Within this framework, malware detection methods are analysed locally and remotely. To determine whether a file is good or bad, signatures are utilised. Various antivirus programmes are used for remote analysis to examine dangerous executables and APIs. Anti- virtual machine, anti-debugger, URL analysis, string analysis, and pack analysis are all used in local analysis. (2018) Ronen et al. [26] offered a widely used benchmark dataset from the Microsoft Malware Classification Challenge that was entered into the Kaggle competition and acknowledged by numerous malware researchers. Over 20,000 malware instances are included in the dataset, which consists of 9 separate malware families and 0.5 terabytes of data. in bytecode and referenced in about 50 academic works. A study on malware detection utilising data mining methods, with a focus on intelligent malware detection methods, was published by leaves et al. in 2017 [27]. They cite his two stages—feature extraction and classification/clustering—as crucial steps in the study and detection of malware.

They discussed the issues and difficulties associated with malware detection when employing data mining techniques, as well as their study results from 2011 to 2016. The implementation and design of sandboxes, feature extractors, and classifiers are presented in kings. 2017 [28]. His three phases—collector, extractor, and classifier—represent the bulk of their job. The PinFWSandbox module, which is part of the Collector, has a static analyzer and dynamic execution that record information from dynamic and log files and send it to the extraction stage. Static feature extraction, dynamic

DOI: https://doi.org/10.5281/zenodo.7609730

statement feature extraction, and feature extraction as a system are all performed by the extractor. The classifier then integrates all of its classifier model actions. B. Results using single-model classifiers, system-call classifiers, dynamic immediate classifiers, and dynamic opcode classifiers produce results that are approximately 96% better for f1 evaluation. Using clustering methods, Pai et al. 2017 [29] provides a malware categorization system. Opcode sequences and their scores are used to extract static characteristics. Using techniques for hidden Markov model clustering, expectation maximisation, and K-means, these static properties are used to categorise malware. Expectation maximisation is a clustering approach that yields incredibly precise results. This method of classifying malware makes use of machine learning clustering algorithms. The methodology presented by Gupta et al. in 2016 [30] leverages the Windows API callsequence.

Among 2000 malware samples, five malware classes—Worm, Trojan-Downloader, Trojan-Spy, Trojan-Dropper, and Backdoor—are categorised using an API call sequence and fuzzy hashing. In order to analyse and categorise instances of unknown malware to cluster according to their families, Liu et al. [31] offer a method. The clustering approach of nearest neighbours was used to categorise the malware instances. 86.7% of unknown malware is successfully categorised as new malware instances according to the experimental result, which shows 98.9% accuracy for known malware instances. An overview of malware analysis and detection methods with various malware families is provided by Makandar et al. in 2017 [32]. Malware is detected and analysed using a variety of methodologies or procedures, one of which is to visualise the malware as an image, such as a grayscale image. A summary of various ongoing works on the visualisation of malware families is provided. Automatic malware classification based on network activity or malware behaviour is suggested by Nari et al. in 2013 [33]. Network traces (pcap files) are used to create behaviour trees, which are then used to classify data using various machine learning methods. Finally, we discovered that when comparing several antiviral programmes, the J48 classifier produced more accurate findings. An automated methodology for categorising unknown malware samples using neural networks is provided by Cosmidis et al. 2017 [34]. From the malimg dataset, features are extracted using feature engineering. Use the perceptron, decision tree, closest centroid, stochastic gradient, multilayer perceptron, and random forest algorithms to categorise unknown malware. Testing time is also taken into consideration as a parameter, and outcomes are improved with random forest on average. 2014 Gundotra, etc. [35] provides an integrated framework to identify unidentified malware using the static and dynamic properties of the integrated feature set to produce improved classifications of malware samples. With the help of four classifiers (multilayer perceptron, IB 1, decision tree, and random forest), the feature-extracted data set is assessed and categorised. According to experimental findings, random forests detected unknown malware and categories with a 99.58% accuracy rate. Earlier before Gandotra et al. [35], Islam et 2013 [36] established the framework, which resembles an integrated framework of static and dynamic functions and makes use of a classifier to perform classification and categorise unknown malware. Gundotra et al. 2014 [37] investigated the use of machine learning in malware investigation with a focus on categorization. A malware family classification using artificial neural networks was proposed by Makandar et al. in 2015 [38]. A grayscale image of the malware binary is created, then it is scaled. The scaled image is subjected to sub-band

DOI: https://doi.org/10.5281/zenodo.7609730

filtering in order to separate itsfeatures and create a feature vector. To extract texture features, the Gabor wavelet and GIST descriptors are employed. The collected features were classified using a feed-forward-back-propagation neural network, and experimental results showed an accuracy of 96.35% for identifying unknown malware. Support vector machines (SVMs) were employed by Kruczkowski et al. in 2014 [39] to categorise malware samples using cross-validation, single exclusion, and random sampling (R S). The accuracy of the three validation methods is improvedby random sampling by a factor of 94.98%. jet al natara reported a novel method for categorising malware samples in 2011 [40].

Malware binaries are transformed into 8-bit vectors, which are thentransformed into grayscale images. For classification,feature vectors and image textures are employed. For feature extraction, gabor wavelets and GIST descriptors are employed. According to test results, the achieves 98% accuracy. Heaven. 2009 [41] efficiently presents malware classification using the printed strings found in the malware executable. On the retrieved features, five classifiers are used: Naive Bayes, Support Vector Machine, IB1, Random Forest, and Decision Tree. The AdaBoostM1 metaclassifier enhances the performance of all classifiers. The experimental findings of WEKA, Random Forest, and IB1 reveal that this method produces better outcomes with an average accuracy of 97%. The n-gram method is used by Kamas et al. 2015 [42] to classify malware. Principal component analysis (PCA) can be used to choose features from data sets for better real-time outcomes. Neural networks (NN), decision trees (J48), support vector machines (SVM), and naive bayes (NB) classifiers are used in classification. Experimental results show that his SVMs [43] deliver automatic malware identification based on behavioural records with 97%accuracy. To extract information from the behavioural logs of malware samples, we use a sandbox environment, Qemu for emulation, and Wine for simulation. To enhance performance, four classifiers—Naive Bayes, Random Forest, J48, and SMO—were applied to the retrieved features. The accuracy of a random forest classifier is good (96.2%). A method for malware classification using useful data features was proposed by Phosphorus et al. in 2015 [44] employing feature selection and feature extraction techniques. The MG TF-IDF method of features chosen precisely from a subset of the dataset is used to execute feature selection on the dataset. The feature extraction method PCA, which reduces the dimensionality of the data set and produces reliable results, is used to analyse these precise features. To increase precision and performance, a support vector machine classifier is applied to the retrieved features. In 2015 [45], Dami et al. suggested a method for categorising and grouping malware samples. Data is gathered from the Cuckoo sandbox environment and pre-processed with the WEKA tool using different classifiers. The LMT classifier outperforms the other five machine learning classifiers in accuracy, scoring 98.3%. To achieve better results, classified malware samples are clustered using the k-means clustering technique. Schulz and others. A data mining strategy for identifying unknown harmful samples was first described in 2001 [46]. LibBFD is used to extract static functionality from the PE executable. The collected characteristics are subjected to the three data mining classifiers Naive Bayes, RIPPER, and Multi- Classifier System.

The average accuracy of the multi- naive Bayesian string is high, coming in at 97.76%. The available research, the tools utilised in the study, the machine learning algorithms employed, the online resources

Advancement in Image Processing and Pattern Recognition
Volume 6 Issue 1
DOI: https://doi.org/10.5281/zenodo.7609730

used to gather the datasets, the parameters taken into consideration to achieve the aim, and what is shown in detail are all summarised in Table 1 below. Table 1 displays the proposed future work they have.

TABLE 1: *SURVEYED PAPERS EXISTING LITERATURE WORK*

Paper	*Tools Used*	*Algorithms Implemented*	*Dataset Sources*	*Result*	*Future Work*
[18]	*Apache Spark, MLlib, Cuckoo Sandbox, Oracle VM VirtualBox*	*Naive Bayes, Random Forest, Support Vector Machine*	*VX Heaven, Nothink, Virushare.*	*Random Forest provides the highest accuracy of the three classifiers tested: Naive Bayes, Support Vector Machine, and Random Forest.*	*Oversee the Dimensionality Reduction process for a big collection of malware samples.*
[19]	*Cuckoo Sandbox*	*Sequence alignment*	*VX Heaven*	*The overall execution time is decreased from 91% to 99%, and the average accuracy of all malware families is 87%.*	*The pair wise sequence alignment approach will enhance the functionality of the suggested system for classifying malware samples and reducing execution time.*

Advancement in Image Processing and Pattern Recognition
Volume 6 Issue 1
DOI: https://doi.org/10.5281/zenodo.7609730

[21]	*Cuckoo Sandbox*	*Random Forest, N, MLP, Support Vector achine, Self rganizing Feature Map*	*Virus total API.*	*Performance increased by 7.24% to 25.68% when compared to the previous model*	*Increasing the sample and granularity of data for other models will perform better?*
[24]					
[28]					
[29]					

DOI: https://doi.org/10.5281/zenodo.7609730

[30]	*ssdeep*	*indows API call sequence and fuzzy hashing .*	*VxVault, VxHeaven, VirusSign*	*Five malware classes are successfully classified using windows API call sequence*	*Text pattern matching techniques are used to classify the malware*	
[20]						
				and fuzzy hashing based classification.		
[31]	---	*Shared nearest neighbour (SNN).*	*Kingsoft, ESET NOD32, and Anubis .*	*Gives 98.9% accuracy of known malware and 86.7% ofaccuracy for unknown malware.*	---	
[33]	*WEKA*	*48, C4.5*	*Communication Research Center Canada (CRC).*	*Better accuracy result between Anti-Virus programs*	----	
[34]	----	*ision tree, nearest centroid, rceptron, tochastic gradient Multilayer Perceptron, Random Forest*	*Malim g dataset.*	*Random Forest gives better accuracy results.*	---	
[35]	*WEKA*	*MLP, DT, IB1, Random Forest*	*Univesity of California*	*Random Forest gives high accuracy results of 99.58%*	----	

DOI: https://doi.org/10.5281/zenodo.7609730

DIALOGUE

Although categorization using machine learning approaches was successful according to prior research on malware detection, significant questions remain. An assault that takes place on the same day as no fresh malware is known as a zero-day attack. All malware researchers have this as their primary objective. AVTest study indicates that there are millions of new virus additions daily. There are still some problems and difficulties with malware detection that need to be addressed. One issue is that when reality is included, practical or manual confirmation of classification results becomes more challenging. Enhancing technological advancements for more active learning is another subject. Machine learning, ensemble learning, deep learning, and other recent developments are anticipated. Moderntechnology is required to execute a zero-day assault. Large data collections need to be handled carefully. These cutting-edge methods are necessary for dimensionality reduction.

CONCLUSION

The available research on malware analysis using different machine learning techniques is summarised in this publication. The studies' tools, machine learning algorithms, sources from which the source datasets were gathered, factors taken into consideration to reach the objective, and corresponding experiments are all summarised in Table 1 along with other relevant literature. describes the outcome. in the future The works are arranged in a table. The discussion made it evident that both small and large datasets can benefit greatly from machine learning algorithms for categorising and clustering malware samples.

REFERENCES

1. AV-TEST (2018, November 28). The Independent ITSecurity Institute, MalwareStatistics [Online]
2. IDAPro. (2018, November 28).
3. OllyDbg. (2018, November 28). [Online].
4. LordPE. (2018, November 28).
5. OllyDump. (2018, November 28). [Online].
6. Willems, C., Holz, T. and Freiling, F. (2007) Toward Automated Dynamic MalwareAnalysis Using Cwsandbox.
7. Anubis. (2018, November28).[Online]
8. Bayer, U., Kruegel, C. and Kirda, E. (2006) TTAnalyze: A Tool for Analyzing Malware. Proceedings of the 15th European Institute for Computer Antivirus Research Annual Conference.
9. Norman Sandbox. (2018, November 28). [Online].
10. Dinaburg, A., Royal, P., Sharif, M. and Lee, W. (2008) Ether: Malware Analysis via Hardware Virtualization Extensions. Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS'08,Alexandria, 27-31 October 2008, 51-62.
11. ThreatExpert. (2018, November 28).
12. Process Explorer. (2014). [Online].
13. Process Monitor. (2014).
14. Capture BAT. (2018, November 28). [Online].
15. Regshot. (2018, November 28). [Online].
16. Wireshark. (2018, November 28). [Online].
17. Process Hacker replace. (2018, November 28).[Online].
18. Gupta, D., & Rani, R. (2018). Big Data Framework for ZeroDay Malware Detection. Cybernetics and Systems, 49(2), 103-121.
19. Cho, I. K., Kim, T. G., Shim, Y. J., Ryu, M., &Im, E. G. (2016). Malware Analysis and Classification Using Sequence Alignments. Intelligent Automation & Soft Computing, 22(3), 371-377.
20. Burnap, P., French, R., Turner, F., &

DOI: https://doi.org/10.5281/zenodo.7609730

Jones, K. (2018). Malware classification using self organising feature maps and machine activity data. computers & security, 73, 399-410.
21. Deepak, N. R., & Balaji, S. (2016, April). Uplink Channel Performance and Implementation of Software for Image Communication in 4G Network. In *Computer Science On-line Conference* (pp. 105-115). Springer, Cham.
22. Thiagarajan, R., Balajivijayan, V., Krishnamoorthy, R., & Mohan, I. (2022). A robust, scalable, and energy-efficient routing strategy for UWSN using a Novel Vector-based Forwarding routing protocol. *Journal of Circuits, Systems and Computers*.
23. NR, D., GK, S., & Kumar Pareek, D. (2022). A Framework for Food recognition and predicting its Nutritional value through Convolution neural network.
24. Thanuja, N., & Deepak, N. R. (2021, April). A convenient machine learning model for cyber security. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 284-290). IEEE.
25. Shanmugam, P., Venkateswarulu, B., Dharmadurai, R., Ranganathan, T., Indiran, M., & Nanjappan, M. (2022). Electro search optimization based long short- term memory network for mobile malware detection. *Concurrency and Computation: Practice and Experience*, *34*(19), e7044.
26. Deepak, N. R., GK, S., & Bhagappa (2021, Nov). The Smart Sailing Robot for Navigational Investigation is Used to Explore all the Details on the Zone of the Water Pura. Indian Journal of Signal Processing (IJSP), 1(4).
27. Deepak, N. R., & Thanuja, N. Smart City for Future: Design of Data Acquisition Method using Threshold Concept Technique.
28. Kiran, M. P., & Deepak, N. R. (2021, May). Crop prediction based on influencing parameters for different states in india-the data mining approach. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1785-1791). IEEE.
29. Deepak, N. R., & Balaji, S. (2015, December). Performance analysis of MIMO-based transmission techniques for image quality in 4G wireless network. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1-5). IEEE.
30. Ab Razak, M. F., Anuar, N. B., Salleh, R., & Firdaus, A. (2016). The rise of
31. ―malware‖: Bibliometric analysis of malware study. Journal of Network and Computer Applications, 75, 58-76.
32. Ray, A., & Nath, A. (2016). Introduction to Malware and Malware Analysis: A brief overview. International Journal
33. Ucci, D., Aniello, L., & Baldoni, R. (2017). Survey on the usage of machine learning techniques for malware analysis. arXiv preprint arXiv:1710.08189.
34. AlAhmadi, B. A., & Martinovic, I. (2018, May). MalClassifier: Malware family classification using network flow sequence behaviour. In APWG Symposium on Electronic Crime Research (eCrime), 2018 (pp. 1-13). IEEE.
35. Khan, M. H., & Khan, I. R. (2017). Malware Detection and Analysis. International Journal of Advanced Research in Computer Science, 8(5).
36. Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., & Ahmadi, M. (2018). Microsoft Malware Classification Challenge. arXiv preprint arXiv:1802.10135.
37. Ye, Y., Li, T., Adjeroh, D., & Iyengar,

DOI: https://doi.org/10.5281/zenodo.7609730

S. S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3), 41.
38. Wang, C., Ding, J., Guo, T., & Cui, B. (2017, November). A Malware Detection Method Based on Sandbox, Binary Instrumentation and Multidimensional Feature Extraction. In International Conference on Broadband and Wireless Computing, Communication and Applications (pp. 427-438). Springer, Cham.
39. Pai, S., Di Troia, F., Visaggio, C. A., Austin, T. H., & Stamp, M. (2017). Clustering for malware classification. Journal of Computer Virology and Hacking Techniques, 13(2), 95- 107.
40. Gupta, S., Sharma, H., & Kaur, S. (2016, December). Malware Characterization Using Windows API Call Sequences. In International Conference on Security, Privacy, and Applied
41. Cryptography Engineering (pp. 271-280). Springer, Cham.
42. Liu, L., Wang, B. S., Yu, B., & Zhong, Q. X. (2017). Automatic malware classification and new malware detection using machine learning. Frontiers of Information Technology& Electronic Engineering, 18(9), 1336-1347.
43. Makandar, A., & Patrot, A. (2015). Overview of malware analysis and detection. In IJCA proceedings on national conference on knowledge, innovation in technology and engineering, NCKITE (Vol. 1, pp. 35-40).
44. Nari, S., & Ghorbani, A. A. (2013, January). Automated malware classification based on network behavior. In 2013 International Conference on Computing, Networking and Communications (ICNC) (pp. 642-647). IEEE.
45. Kosmidis, K., & Kalloniatis, C. (2017, September). Machine Learning and Images for Malware Detection and Classification. In Proceedings of the 21st Pan-Hellenic Conference on Informatics (p. 5). ACM.
46. Gandotra, E., Bansal, D., & Sofat, S. (2014, September). Integrated framework for classification of malwares. In Proceedings of the 7th International Conference on Security of Information and Networks (p. 417). ACM.

Cite as : Tahir Naquash, Yusuf Ahmad, Satyam Singh, Shubhanshu Kumar, & Yash Anjana. (2023). Detection of Malware Using Machine Learning Algorithms. Advancement in Image Processing and Pattern Recognition, 6(1), 1–12. https://doi.org/10.5281/zenodo.7609730