Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/11900
DOI URL: http://dx.doi.org/10.21474/IJAR01/11900

RESEARCH ARTICLE

EMERGING MACHINE LEARNING TECHNIQUES IN MALWARE DETECTION AND ANALYSIS: A COMPARATIVE ANALYSIS

Prithvi Chintha[1] and Dr. Kakelli Anil Kumar[2]

1. SCOPE, Vellore Institute of Technology, Vellore, India.
2. Associate Professor, SCOPE, Vellore Institute of Technology, Vellore, India.

Manuscript Info

Manuscript History
Received: 20 August 2020
Final Accepted: 24 September 2020
Published: October 2020

Key words:-
Malware Detection, Static Analysis, Behavior-Based Analysis, Windows Executables

Abstract

New types of malware with unique characteristics are being created daily in legion. This exponential increase in malwareis creating a threat to the internet. From the past decade, various techniques of malware analysis and malware detection have been developed to prevent the efficacy of malware. However, due to the fast-growing numbers and complexities in malware, it is getting difficult to detect and analyze the malware manually. Because of the inefficiency in manual malware analysis, automated malware detection and analysis would be a better solution. Thus, malware analysis supported by machine learning became a required part of malware analysis. The automation used in learning patterns in malware can help in efficiently identifying the complexities. Malware Analysis with help the Machine learning would be more efficacious in terms of automation and memory usage. In this paper, we conducted a review of emerging various ML (Machine Learning) strategies used so far, in the field of malware analysis, to give a comprehensive view of the existing processes. We systemized them on various aspects like their objectives, machine learning algorithms used, information about the malware, etc. We also highlighted the existing problems in this particular field of study and tried to find multiple ways in which advancements can happen concerning the current trends being used.

Introduction:-

Machine Learning is a process where a computer (machine) assimilates on its own using empirical analysis and tries to recognize patterns in the functionality [1]. This process helps a computer to make its predictions by testing and training datasets. Humans do not manually script their functioning but the machines themselves learn in an automated fashion. Machine learning has various applications in predicting the traffic in our daily lives, recommending products to the customers based on their previous purchases, recognizing images and objects, refining search engine results, filtering malware, etc. It specifically has many applications in the field of cybersecurity such as identifying if an email is malicious or not by recognizing email headers, body data, etc. In addition to that, it is used in spearfishing, watering hole, web shell, ransomware, remote exploitation, etc. To be more specific, it has applications in the field of malware analysis too. Malware is malicious software that can change the functionality of a system or an application without having a permit. Malware is designed to maliciously affect software and systems [2]. There are many types of malware such as worms, viruses, ransomware, spyware, adware,

Corresponding Author:-Prithvi Chintha
Address:-SCOPE, Vellore Institute of Technology, Vellore, India

trojan horses, rootkits, etc. Analyzing this malware is a method used to determine the dawn, impact, and functionality of the malware. It is also about understanding the behavior of malware and find out multiple ways to detect and mitigate it. Majorly two kinds of malware analysis exist which are classified as static malware analysis and dynamic malware analysis. Both of these techniques involve detecting malware manually [3].

Since, the malware is increasing in numbers and complexities, detecting them using an older version of techniques is not giving efficacious results. Machine learning automates the process of detecting the malware which is more efficient, and easy to work. However, the efficacy of the machine learning algorithms generally has an impact on the system state and its performance. A perfect machine learning model to detect malware follows the following conditions. Firstly, a machine model is created and developed using large amounts of data which is known for the training of that particular model [4]. This helps to understand which amongst the various features of the malware are statistically germane for a proper prediction of the best label. In addition to that, the data that we use for training a model should be appropriate to the pragmatic conditions of reality because this shows that choosing the right dataset is a crucial part to make the model successful.

Secondly, we must be able to interpret a trained model. For instance, deep neural networks are considered as black-box models [5]. That is because we cannot understand the process of prediction but just inputs and the outputs. So, in this situation, when a false alarm occurs, we cannot understand if the problem is with the model that was chosen or the dataset. Thirdly, the model that was chosen shall have very low false-positive rates. It occurs when an algorithm mistakes a harmful label with a normal one. The false-positive rates have to be nulled because even one false positive in thousands would still create dreadful consequences for the users of the systems. Finally, a perfect model adapts itself hastily with the counteractions of the malware writers who try to overcome their vulnerability. Malware writers consistently work on eschewing detection and create various malicious files which are very different from what was seen in the training phase. In addition to this, many companies create new innocuous executable files that are different from previously known files. These should also be taken into consideration as they are lacking in the training dataset. These changes can lead to catastrophic consequences like changing the data distributions, increasing problems in detection rate, etc. [6]. Thus, to avoid all these consequences, a model while being developed to detect a malware sample shall be adaptable to the changes made by the active adversaries. In this paper, we present a comprehensive analysis and review of various techniques that are being used currently by machine learning in analyzing different types of malware. Firstly, we will see the methodologies used in behavioral malware detection using machine learning. In this part of the research, various machine learning algorithms will be compared and the best optimal algorithm will be found out. The work can cause further development and also can be used as a reference for later meticulous research in these particular fields of machine learning and malware analysis [7].

Methodology: -

Static-based Malware Analysis

There are various kinds of algorithms used in this particular type of analysis which is based on their methods of functionality such as statistical methods, rule-based method, distance-based method, neural network-based methods, feature selection with the construction process, and open source machine learning tools [8]. While exploring binary files in legion, using frequencies or similarities of the feature values, the statistical features can be captured and analyzed. The methods used in this use various statistical techniques that are used in malware analysis to predict the type of binary executable. The following set of statistical methods are used to extract and process such data [9].

Methods → **Algorithms ↓**	Statistical Methods	Rule Based	Distance Based	Neural Networks	Open Source ML Tools	Feature Selection
Naive Bayes	✓					
Bayesian Networks	✓					
Decision tree		✓				
K-NN			✓			
SVM			✓			
ANN				✓		
Weka					✓	
LIBSVM					✓	

RapidMiner					✓	
Dlib					✓	
Information Gain						✓
CFS						✓

Table 1:- List of Algorithms and methods used.

Naïve Bayes algorithm is a simple classifier which deals with probability. It is derived from the Bayesian theorem with ingenious suppositions correlation independence of various qualities [10]. Bayesian Networks are an acyclic graphical model that is directed by probability. It is also known as Bayesian Belief Networks (BBN) [11]. It manifests dependencies that are conditional using a directed acyclic graph. Networks are used in detecting the updated grasp of the current state when the evidence variables are found. It is used in various cases of information and classification retrieval. Rule-Based methodologies can create invigorating and imprecise rules for various ML Algorithms. The paramount advantage of this method in the classification of malware is that the analytical rules which are operating can be run at the hardware level that results in an increase in speed for decision making [12].

A decision tree algorithm [13] is used for classification and specific detection of the malware. The tree processing involves the processing of an already classified dataset followed by finding an attribute on each step that can divide the sample set into a part of a larger group of related things that has the best IG (information gain) value. Neuro-Fuzzy is a complex model that combines fuzzy logic with neural networks to generate rules which are similar to our rules with the help of artificial neural networks as shown in figure 2. A perceptron is a basic unit in the neural network that computes to detect features in the inputted data [14]. Distance-Based Methods are based on predefined distance measurement used for classification. Data in these methods should be carefully prepared because as the number of features in these models grow, computations complexity grows too. Thus, proper feature selection plays an important part in this process. k-Nearest Neighbors an algorithm [15] that helps for both classification models and regression models. This algorithm doesn't require a separate adept training or preparation of datasets as it is ready precisely after we label the dataset. This method initially uses a trial that is needed for classification and then begins to calculate the distances between the samples [16]. After that, the K nearest neighbors which have the shortest distances are selected and used in the decision-making process. Support Vector Machine [17] is a supervised learning model that is helpful to find a hyperplane in an N-dimensional space that can classify data points distinctly as shown in figure 1. As the number of features increases, the number of dimensions also increases along with the complexity. So, this algorithm manages to subject any dimensional data into an N dimension for simple calculations [18].

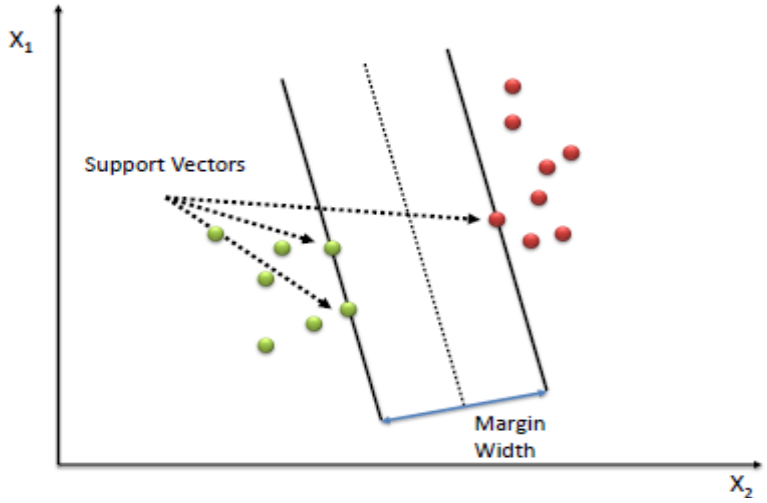

Fig. 1:- Support Vector Machine.

Neural Networks works on the functioning of a perceptron which is the elemental unit of a neural network. A perceptron has a predefined activation function. It reduces the error rate of processing information using backpropagation. ANN has 3 different layers which are the input, hidden, and output layers [19]. Input layers include the features that are been selected for the processing, hidden layers produce activation output which is

dependent on the output of the activation function along with the weighted input of the neurons. Activation function intakes an input value and multiply it with the weights of the corresponding edges and produce a probabilistic output which lies in the range of 0 to 1. The output layer presents the results and is used to interpret them [20].

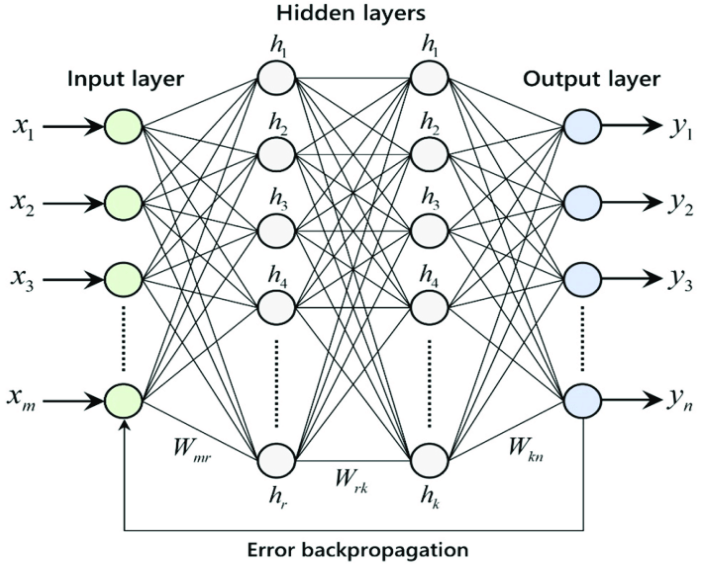

Fig. 2:- Artificial Neural Network.

Open Source ML Tools

Today, many machine learning tools are publicly available as they are being used ubiquitously [21].

1. **Weka:** It is a famous, free, and open-to-all tool used in Machine Learning. It finetunes the parameters used and also help in result analysis [22].
2. **Python Weka wrapper:** It is a package that helps to use Weka using python programming. It uses java bridge which helps in linking Java-based Weka libraries to python [23].
3. **LIBSVM:** It is a publicly available library of Machine Learning scripted in the C++ language. It supports Support Vector Machines that are kernelized for linear, classification, and regression analysis [24].
4. **RapidMiner:** It is a data-mining and Machine Learning tool which has a user-friendly graphical user interface that supports various ML algorithms [25].
5. **Dlib:** It is a free of charge cross-platform toolkit which is programmed in C++. It supports various ML algorithms and along with multithreading to increase efficiency and reduce the time complexity [26].

Feature Selection

In this step, we pass the phase of characteristics extraction and reach the process of feature selection. In this process, a set of redundant and inapposite features are eliminated which do not influence the classification of malware. This is an important step because the characteristics are abundant in number where only very few can be useful to distinguish between benign and malware samples. Correlation-based Feature Subset Selection (CFS) and Information Gain are the very frequently used selection methods [27].

Behavior-based Malware Analysis

A basic overview of the research strategy and methods as depicted below in the following figure 3 for behavioral malware analysis with help of various ML algorithms. Figure 3 shows several steps that are included in this entire process [28].

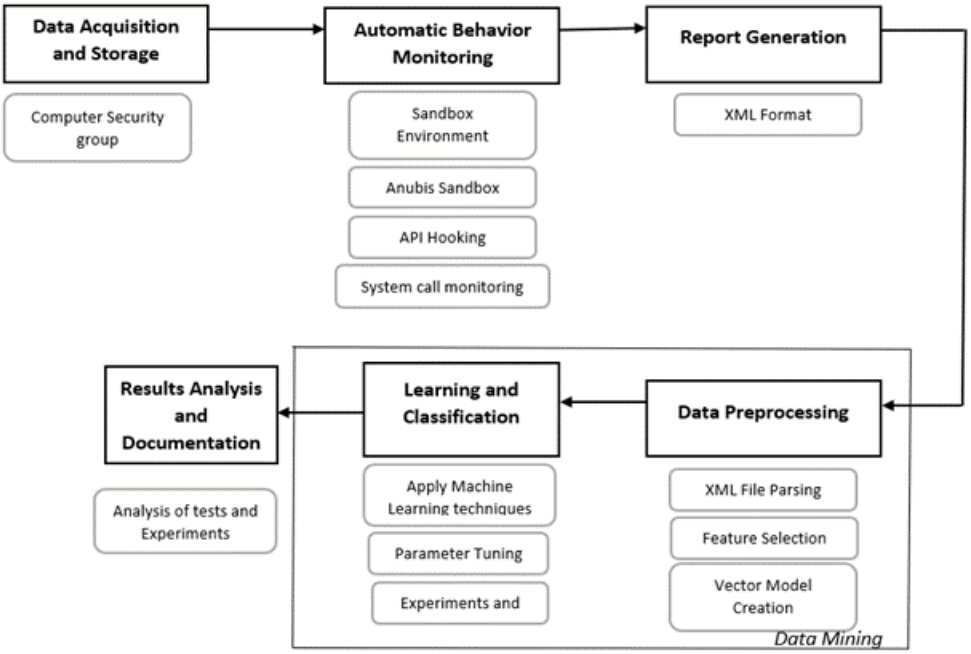

Fig 3:- Behavioral malware analysis using ML algorithms.

Data Acquisition and Storage

The dataset that is being used contains both benign and malware types of datasets. Both of them are found in PE (Portable Executable) file binaries. For instance, two hundred and twenty unique malware samples are acquired along with 250 unique benign software samples [29, 30].

Automatic Behavior Monitoring and Report Generation

In this step, malware samples and benign samples are subjected to dynamic behavior analysis. This is done by putting each sample in an online platform where free dynamic analysis services are provided such as a sandbox environment. This submission of file results in report generation. In this particular, all the reports were downloaded in XML format which will be used in the further process [31].

Data Preprocessing, Learning and Classification

This is the step where data preprocessing occurs. In this step, the germane and paramount attribute values are selected after parsing, through all the XML files followed by the creation of a term dictionary that includes all the previously selected attributes. After that, using term frequency weight and binary weight, each XML file is juxtaposed with the term dictionary by enumerating the presence/absence of each term word. Finally, for each XML report, Spare Vector Models are created along with files having Attribute-Relation File Format (ARFF). In the learning and classification step, Machine Learning techniques are applied to learn and classify the files with ARFF format [32].

Results of the Experiments:-

Tests are executed based on 4 different types of datasets which are Binary-weight vector model with and without feature selection and Term frequency-weight vector model with feature selection. Now, different machine learning techniques like KNN classifier, Naïve Bayes Classifier, Support Vector Machine (SVM), and J48 are used in the above-mentioned process to find out the most optimal algorithm. To measure performance metrics, various statistical measures are calculated for all the ML Algorithms in 4 different types of datasets which are shown below tables 1,2, 3, and 4.

Classifier	TPR	FPR	PPV	Accuracy
kNN	81.7%	8.1%	91.8%	86.5%
Naïve Bayes	58.1%	12.8%	93.2%	65.4%
SVM	90.4%	8.4%	90.4%	91.0%
J48	90.9%	3.8%	95.9%	93.6%

Table. 2:- Performance Metric Results of Binary weight datasets before feature selection.

Classifier	TPR	FPR	PPV	Accuracy
kNN	86.8%	8.8%	90.4%	89.1%
Naïve Bayes	56.8%	22.2%	86.3%	62.8%
SVM	90.5%	7.3%	91.8%	91.7%
J48	95.9%	2.4%	97.3%	96.8%

Table. 3:- Performance Metric Results of term frequency weight datasets before feature selection.

During the process of feature selection using different algorithms like Best First search and Correlation-based Feature Selection, the attributes for the binary-weighted datasets were reduced to 116 attributes from 5191 attributes which are 97.7% reduction. On the other hand, the attributes for the term frequency-weighted datasets were reduced to 11 from 5191 attributes which are 99.7% reduction.

Classifier	TPR	FPR	PPV	Accuracy
kNN	94.3%	8.1%	90.4%	92.9%
Naïve Bayes	94.2%	9.2%	89.0%	92.3%
SVM	94.3%	8.1%	90.4%	92.9%
J48	94.2%	9.2%	89.0%	92.3%
MLP	94.0%	11.2%	86.3%	91.0%

Table. 4:- Performance Metric Results of Binary weight datasets after feature selection.

Classifier	TPR	FPR	PPV	Accuracy
kNN	94.3%	8.1%	90.4%	92.9%
Naïve Bayes	58.7%	19.1%	87.7%	65.4%
SVM	94.1%	10.2%	87.7%	91.7%
J48	94.5%	4.8%	94.5%	94.9%
MLP	94.4%	6.0%	93.2%	94.2%

Table. 5:- Performance Metric Results of term frequency weight datasets after feature selection.

From the observations made using the above tables 1, 2, 3, and 4, the following conclusions were made. The best performance was achieved by the J48 algorithm on both types of datasets. In addition to that, even though feature selection took place and the attributes were reduced, better results in performance were achieved after feature selection.

Malware Detection using ML methodologies for windows executables:-

A detailed methodology contains a proposed malware detection and analysis system specifically used for '.exe' files. It is mainly focused on the detection of the malware which occurs in API calls and followed by the classification of its type as either trojan, worms, viruses, or normal [33]. This process involves the following stages below.

Pre-processing

The input is sequences of viruses, trojans, and worms. In this phase, after extracting the sequence of system function from '.exe' files, the input data is subjected to preprocessing. After that, the dataset attributes are taken and each of them is separated followed by the execution of a sampling process that is used in the categorization of the type of data [34].

Feature extraction
After the first phase, the data is transformed into tokenized data for feature extraction and then unique call sequences like id and call token are identified using the system directory. Using statistical methods like Standard Deviation (SD) and mean, the upper and lower boundary values are calculated. (SD + Mean) gives upper boundary values and (SD - Mean) gives us the lower boundary values.

Rule generation
In this phase, the Rete algorithm is used. After feature extraction, the confidence and support values are predicted by fixing a threshold value. A set of rules are generated using a rete algorithm that controls how the malware detection system functions. Rete algorithm is widely used in the applications of data mining which is used in the comparison of largely collected sets of patterns from humongous data. The algorithm chooses to eschew iterations over facts and rules. The nodes in this process are analyzed in a graph-like structure which is later merged to avoid redundancy during the evaluation of conditions. When veracity is defenestrated from the truth base, it leads to the removal of the association from corresponding rules.

Classification
To classify the type of malware, MDNBS is used to enhance the efficiency of this algorithm, two different temporal and spatial features are combined. This method of multi-dimensional Naïve Bayesian technique uses 1 or more than 1 feature variables to examine the relationship between features and variables of a class. An advantage of the technique is that it extracts various information from different fields, it automatically discovers classes, it has faster training time and most importantly it reduced computational complexity as shown in table 5 [35].

Actual	Predicted	Predicted
	Malware	Normal
Malware	55	4
Normal	7	9

Table 6:- Confusion matrix.

To analyze these results, various performance metrics have been used inthe equations 1, 2, 3 and 4 shown below.
True Positive Rate = True Positive / (True Positive + False Negative) …… (1)
False Positive Rate = False Positive / (True Negative + False Positive) ……(2)
Precision = True Positive/ (True Positive + False Positive) ……(3)
F − Measure = (2 × Recall × Precision) / (Recall × Precision) ……(4)

Conclusion:-

A comprehensive survey is based on various machine learning techniques used in detecting and analyzing malware. The work has presented the review on three types of malware analyses which are different and each focusing on specific methodologies such as static-based malware analysis, behavior-based malware analysis, and malware detection for windows executable files to detect and analyze malware while manual analysis is not providing enough efficacy. Machine learning techniques are used for static analysis; different algorithms are used in various ways for PE32 Windows malware. It was inferred that KNN and C4.5 mostly have better performance results when compared to the other algorithms. SVM and ANN also show good performance on some feature sets. It is thus seen that static analysis using ML techniques are highly efficacious. ML methodologies and techniques that were used in behavioral malware analysis, feature selection was depicted using the Greedy Algorithm (BFS). Feature selection has reduced the cost of performance as the number of features was reduced drastically. After comparing the performance of five various algorithms, it had been inferred that J48 algorithm achieved the best performance. In malware detection and classification for API call sequence, we concluded that MDNBS using the Rete algorithm was the most efficient method proposed. The main advantage of the MDNBS technique is, it reduces computational complexity, time consumption, and enhances detection rate. The results were compares using various metrics of performance such as FPR, TPR, precision, f-measure, and processing time. The review work would provide a deep view and can help the researchers how to use and choose the advanced machine learning algorithms for effective malware detection and analysis.

References:-

1. Saad, S., Briguglio, W., &Elmiligi, H. (2019). The curious case of machine learning in malware detection. arXiv preprint arXiv:1905.07573.
2. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. IEEE Access, vol 6, pp 35365-35381.
3. Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018). On the effectiveness of machine and deep learning for cyber security. In 2018 10th International Conference on Cyber Conflict (CyCon), pp. 371-390. IEEE.
4. Yavanoglu, O., &Aydos, M. (2017). A review on cyber security datasets for machine learning algorithms. In 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp 2186-2193.
5. Chen, H., Fu, C., Zhao, J., &Koushanfar, F. (2019). DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks. In IJCAI, pp. 4658-4664.
6. Navarro, L. C., Navarro, A. K., Grégio, A., Rocha, A., & Dahab, R. (2018). Leveraging ontologies and machine-learning techniques for malware analysis into Android permissions ecosystems. Computers & Security, vol 78, pp 429-453.
7. Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, vol 81, pp 123-147.
8. Shalaginov, A., Banin, S., Dehghantanha, A., & Franke, K. (2018). Machine learning aided static malware analysis: A survey and tutorial. In Cyber Threat Intelligence, Springer, Cham pp. 7-45.
9. Arslan, R. S., Doğru, İ. A., &Barişçi, N. (2019). Permission-based malware detection system for android using machine learning techniques. International Journal of Software Engineering and Knowledge Engineering, vol 29(01), pp 43-61.
10. Raghuraman, C., Suresh, S., Shivshankar, S., &Chapaneri, R. (2020). Static and dynamic malware analysis using machine learning. In First International Conference on Sustainable Technologies for Computational Intelligence, Springer, Singapore, pp. 793-806.
11. Sartea, R., Chalkiadakis, G., Farinelli, A., &Murari, M. (2020). Bayesian Active Malware Analysis. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1206-1214.
12. Mehtab, A., Shahid, W. B., Yaqoob, T., Amjad, M. F., Abbas, H., Afzal, H., & Saqib, M. N. (2020). AdDroid: rule-based machine learning framework for android malware analysis. Mobile Networks and Applications, vol 25(1), pp 180-192.
13. Irshad, A., & Dutta, M. K. (2020). Identification of Windows-Based Malware by Dynamic Analysis Using Machine Learning Algorithm. In Advances in Computational Intelligence and Communication Technology, Springer, Singapore, pp. 207-218.
14. Sethi, K., Chaudhary, S. K., Tripathy, B. K., &Bera, P. (2018). A novel malware analysis framework for malware detection and classification using machine learning approach. In Proceedings of the 19th International Conference on Distributed Computing and Networking, pp. 1-4.
15. Bhodia, N., Prajapati, P., Di Troia, F., & Stamp, M. (2019). Transfer learning for image-based malware classification. arXiv preprint arXiv:1903.11551.
16. Jerlin, M. A., &Marimuthu, K. (2018). A new malware detection system using machine learning techniques for API call sequences. Journal of Applied Security Research, 13(1), pp 45-62.
17. Wadkar, M., Di Troia, F., & Stamp, M. (2020). Detecting malware evolution using support vector machines. Expert Systems with Applications, vol 143, 113022.
18. Mursleen, M., Bist, A. S., & Kishore, J. (2019). A support vector machine water wave optimization algorithm-based prediction model for metamorphic malware detection. International Journal of Recent Technology and Engineering, vol 7, pp 1-8.
19. Shi-qi, L., Bo, N., Ping, J., Sheng-wei, T., Long, Y., & Rui-jin, W. (2019). Deep Learning in Drebin: Android malware Image Texture Median Filter Analysis and Detection. KSII Transactions on Internet and Information Systems (TIIS), vol 13(7), pp 3654-3670.
20. Rami, K., & Desai, V. (2020). Malware Detection Framework Using PCA Based ANN. In International Conference on Computing Science, Communication and Security, Springer, Singapore, pp. 298-313.
21. Talukder, S. (2020). Tools and techniques for malware detection and analysis. arXiv preprint arXiv:2002.06819.
22. Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M., & Frank, E. (2019). Wekadeeplearning4j: A deep learning package for weka based on deeplearning4j. Knowledge-Based Systems, vol 178, pp 48-50.

23. Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á. L., Heredia, I., &Hluchý, L. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review, vol 52(1), pp 77-124.
24. Shaikh, S., Changan, L., Malik, M. R., & Khan, M. A. (2019, December). Software Defect-Prone Classification using Machine Learning: A Virtual Classification Study between LibSVM&LibLinear. In 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), IEEE, pp. 1-6.
25. Kawelah, W. A. A. S., & Abdala, A. S. E. A Comparative Study on Machine Learning Tools Using WEKA and Rapid Miner with Classifier Algorithms C4. 5 and Decision Stump for Network Intrusion Detection. European Academic Research, 7(2).
26. Lashkov, I., Kashevnik, A., Shilov, N., Parfenov, V., &Shabaev, A. (2019). Driver Dangerous State Detection Based on OpenCV &Dlib Libraries Using Mobile Video Processing. In IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), IEEE, pp. 74-79.
27. Ali, A., Shamsuddin, S., & Hassan, S. (2020, July). Feature selection for malicious android applications using Symmetrical Uncert Attribute Eval method. In IOP Conference Series: Materials Science and Engineering, IOP, vol. 884, No. 1, pp 012060.
28. Xiao, F., Lin, Z., Sun, Y., & Ma, Y. (2019). Malware detection based on deep learning of behavior graphs. Mathematical Problems in Engineering, 2019.
29. Chaudhary, S., & Garg, A. (2020, January). A Machine Learning Technique to Detect Behavior Based Malware. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, pp. 655-659.
30. Ugarte-Pedrero, X., Graziano, M., &Balzarotti, D. (2019). A close look at a daily dataset of malware samples. ACM Transactions on Privacy and Security (TOPS), vol 22(1), pp 1-30.
31. Huda, S., Abawajy, J., Al-Rubaie, B., Pan, L., & Hassan, M. M. (2019). Automatic extraction and integration of behavioural indicators of malware for protection of cyber–physical networks. Future Generation Computer Systems, vol 101, pp 1247-1258.
32. oo, S., Kim, S., Kim, S., & Kang, B. B. (2020). AI-HydRa: Advanced Hybrid Approach using Random Forest and Deep Learning for Malware Classification. Information Sciences.
33. Singh, J., & Singh, J. (2020). A survey on machine learning-based malware detection in executable files. Journal of Systems Architecture, 101861.
34. Javid, W. (2019). Machine Learning Aided Static Malware Analysis. International Journal Of Computer Science and Technology, vol 3(2), pp 1-11.
35. Choi, S. Y., Lim, C. G., & Kim, Y. M. (2019). Automated Link Tracing for Classification of Malicious Websites in Malware Distribution Networks. Journal of Information Processing Systems, vol 15(1).