International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

Malicious Detection and Prediction using Machine Learning

Himanshi Sonparote[1], Parikshit Arekar[2], Roshani Madankar[3], Shreyas Gosavi[4], Rasika Badre[5]

[1]Student, Computer Science & Engineering Department, PRMIT&R, Badnera
[2]Student, Computer Science & Engineering Department, PRMIT&R, Badnera
[3]Student, Computer Science & Engineering Department, PRMIT&R, Badnera
[4]Student, Computer Science & Engineering Department, PRMIT&R, Badnera
[5]Assistant Professor, Computer Science & Engineering Department, PRMIT&R, Badnera.

Abstract - In the domain of computer security, there is now considerable research into machine learning-based malware detection and prediction. Algorithms for machine learning have showed promise. outcomes in the identification and prediction of harmful software existence in computer systems. The goal of this strategy is to provide precise and effective tools for identifying and avoiding malware attacks. The term "malware" describes harmful software, such as viruses, worms, Trojan horses, ransomware, and spyware, that is intended to harm computer systems. Attacks by malware can cause large monetary losses, privacy breaches, and reputational harm. Therefore, it is essential to provide reliable and effective techniques for identifying and avoiding malware attacks.
Algorithms for machine learning have demonstrated promise in the detection and forecasting of dangerous software. Computer learning is a branch of artificial intelligence that deals with teaching algorithms to discover patterns in data and generate predictions using those patterns. The method includes giving a machine learning algorithm a sizable dataset of known malware samples, which subsequently learns to recognize common patterns and traits associated with malware. The trained algorithm may then be employed to determine if new and unexplored data contains malware.

Key Words: Technological innovation; manipulation thread; KNN; SVM; DT; Cyber security; Cyber-attack; suspicious activity; Cyber threat.

1. INTRODUCTION

Malware is malicious software that compromises a system's security, integrity, and functioning without the user's knowledge in order to carry out the attacker's negative intentions. Malware comes in a variety of forms, including viruses, worms, and Trojan horses, rootkits, backdoors, botnets, spyware, and adware. Antivirus software uses signature-matching algorithms to identify known risks in order to detect and stop malware from being executed. A signature-based database is used by the anti-virus software to find malware. The anti-virus program does a file scan, creates a signature, and then verifies if the signature is present in the database. The file under evaluation is malware if there is a match. Although the virus is appropriately classified by this method, it cannot identify fresh or unknown malware because its signature won't be present in the database. Additionally, attackers can employ a variety of methods, including obfuscation, polymorphism, and encryption, to get through firewalls, gateways, and antivirus systems even when using recognized malware. Dead code insertion, register reassignment, subroutine reordering, instruction replacement, code transposition, and code integration are a few of the obfuscation methods that are frequently utilized. Static analysis is used to identify patterns and extract information such as texts, n-grammes, byte sequences, opcodes, and call graphs. The assembly instructions are created by reverse engineering the Windows executable using disassembler tools. Additionally, protected code (placed in system memory) is retrieved, and memory dumper tools are used to examine packaged executables, which are otherwise difficult to deconstruct. For analysis, the executable must first be unpacked and perhaps encrypted. Static analysis is a difficult alternative since techniques like obfuscation, encryption, polymorphism, and metamorphism might prevent the reverse compilation process from working. Additionally, during the binary compilation of the source code, certain information is lost, such as variables or data structure size. Statistical analysis is used to get beyond these restrictions since it is less vulnerable to obfuscation techniques.
According to statistical analysis, malicious code is run in a virtual or controlled environment. Monitoring is done for actions like function calls, function parameters, information flow, instruction traces, etc. The length of execution, transmitted network traffic, and file system modifications are just a few examples of additional runtime data that may be recorded. Additionally, some malware exhibits distinct behaviors in a virtual environment compared to a physical one, making it more difficult to identify. Furthermore, under specific circumstances, such as a system date, the malicious behaviors may be activated. Therefore, malware that is aware of execution circumstances and the computer environment may readily elude statistical analysis approaches. As the files are run in statistical analysis, their behavior is recorded in a feature vector space that captures their pattern. Although static methodologies can be utilized for analysis, a variety of machine-learning algorithms have been employed to automate the malware analysis and classification phases in order to reduce the number of samples requiring close human inspection. In order to classify unknown malware into the appropriate families, machine learning techniques (such as clustering and classification) are employed to analyze the patterns discovered through static and/or dynamic analysis.
An issue with categorization is figuring out if a file is malware or not. The virus is categorized using a variety of machine-learning methods, including Naive Bayes, Decision Trees, Support Vector Machines, KNNs, etc. The dataset in this

machine learning technique typically consists of the files, and the label denotes whether the file is malicious or not.
To extract characteristics from the lowest level to the highest level, additional layers are added to the machine learning approach. Each layer in this situation recognizes a certain kind of characteristic and passes it on to the next layer. These lower-level traits are then combined to create higher-level features in the following layer, and so on. The last layer in the model may finally classify whether the file is malignant or benign since these features are sort of aggregated as they are passed. Deep learning allows the model to self-extract features, as opposed to machine learning, where the feature set must be given to the network. In this, we analyze the dataset using machine learning and compare the outcomes.
The best feature extraction approach, the best feature representation technique, and the most precise algorithm that can differentiate the malware families with the lowest error rate must all be found. The detection of whether the file is harmful and the categorization of the file into the malware family will both be measured for accuracy. The reliability of the results will also be evaluated with reference to the scoring system currently used in confusion matrixes, and the decision of which method performs better will be made.

2. Literature Survey

Many researchers are also attempting to develop new ways. are more efficient than the existing methods, and shows better segmented result. Some of the most recent works are as follows: "Detection of Malware Using Machine Learning" A detecting system based on several customized perceptron algorithms was Dragos Gavrilut's goal. He attained an accuracy of 69.90% for various methods. It should be noted that the algorithms with the highest accuracy also produced the falsest positives; the algorithm with the highest accuracy created 48 false positives. The most "balanced" algorithm has a 92.01% accuracy rate, adequate accuracy, and a low false-positive rate. 2009; Gavrilut et al.
In order to examine and quantify the detection accuracy of the ML classifier that employed static analysis to extract features based on PE information, Nur (2019) tested three ML classifiers. We collectively taught machine learning algorithms to distinguish between harmful and beneficial content. The most accurate classifier we looked at, the DT machine learning approach, achieved 89% accuracy. In order to obtain the maximum detection accuracy and the most accurate representation of malware, this experiment showed the possibility of static analysis based on PE information and selected important data elements.
Chowdhury (2018) suggested an effective malware detection method based on machine learning classification. We investigated if altering a few factors may improve the accuracy with which malware is categorized. Our technique combined N-gram and API call capabilities. The usefulness and reliability of our proposed approach were proven by experimental assessment. Future research will concentrate on combining a large number of characteristics to improve detection precision while reducing false positives. As can be observed, each study yielded a distinct set of results. From this, we may conclude that no uniform approach for detection or feature representation has yet been developed. The correctness of each individual scenario is determined on the malware families employed and the actual implementation. Malicious programs and their risks, sometimes known as "malware," grew more prevalent and complex as the Internet evolved. Because of its quick spread over the Internet, malware authors now have access to a wide range of malware production tools. Malware's reach and complexity expands by the day. This research focused on analyzing and quantifying classifier performance in order to have a better understanding of how machine learning works. It was suggested that ML systems be taught and tested to assess whether a file is dangerous. The experimental results demonstrated that the decision tree technique is superior for data classification, with 97.0 percent accuracy. These findings demonstrated that the PE library was compatible with static analysis and that concentrating on a few features might enhance malware identification and characterization. The key advantage is that malicious software is less likely to be installed by accident since users may validate a file before opening it.

3. Algorithms

❖ Naïve Bayes:

The classification machine learning algorithm that uses the Bayes Theorem is called Naive Bayes. Both binary and multi-class classification issues may be solved with it. The concept of considering each characteristic separately is central to the argument. Without consideration to correlations, the Naive Bayes technique assesses the likelihood of each characteristic separately and bases its prediction on the Bayes Theorem.
A type of order calculation known as the naive Bayes hypothesis may be used to solve problems involving both two and more classes. Due to the fact that it is based on the Bayes hypothesis, this hypothesis is given that name. Probabilities frequently converse with innocent Bayes. The data in this model is saved. as probabilities for an informed model.

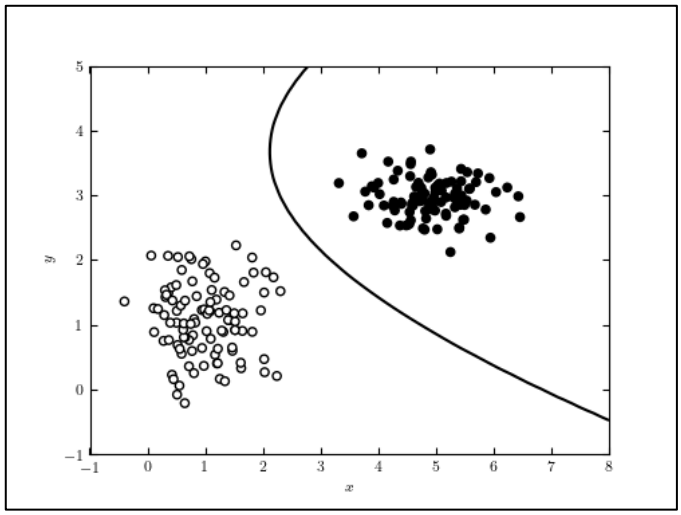

Fig -3.1: Naïve Bayes Classifier

❖ Decision Tree:

A type of directed learning computation called a decision tree uses an information structure to solve a problem. The leaf hub is referred to in this instance as the class mark, while the internal hubs of the tree refer to the attributes. The full dataset is initially taken into account as the root, the distinct element esteems are liked, and the persistent qualities are first turned into discrete qualities before being used to construct the model. The characteristics are then requested as the inner hub or root using quantifiable methods.

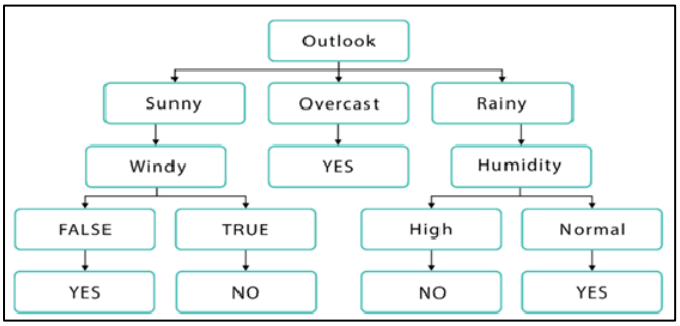

Fig -3.2: Decision Tree Example

❖ **KNN:**

One of the most straightforward but accurate machine learning techniques is K-Nearest Neighbors (KNN). The KNN method is non-parametric, which means it makes no assumptions about the data structure. Non-parametric algorithms are a useful answer for such situations since in real-world problems, data seldom complies with the basic theoretical assumptions. There is no need for learning because the complete training set is encoded in the KNN model representation, which is as straightforward as the dataset.

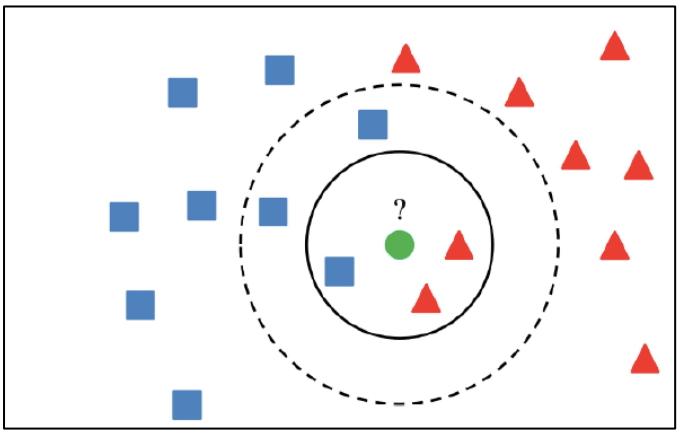

Fig -3.3: KNN Example

❖ **SVM:**

Another machine learning approach that is typically applied to classification issues is Support Vector Machines (SVM). The central concept depends on identifying a hyperplane that the separation between the support vector and the hyperplane. Optimally divides the classes. The points closest to the hyperplane that, if removed, would shift the hyperplane's location are referred to as "support vectors." Margin is the distance between the support vector and the hyperplane.

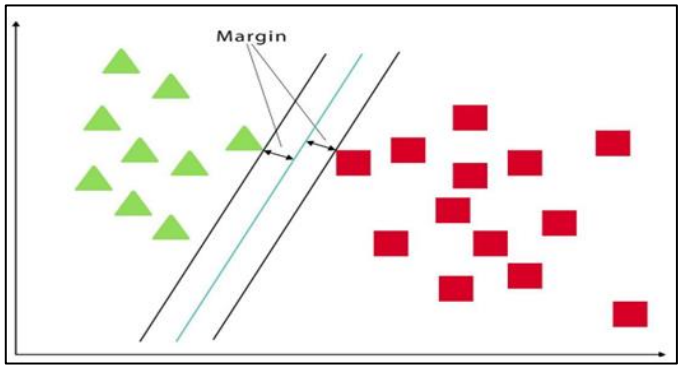

Fig -3.4: SVM Scheme

4. System Architecture

Machine Learning calculations are a kind of calculations that are a part of man-made brainpower and that makes the framework or the product application to be eager enough to get the option to progressively precise without being expressly customized and can anticipate results. The principle thought behind these kinds of calculations is that it gets input information as content or pictures and the framework or the model is prepared with the factual contributions to distinguish or foresee the yield and even refreshed the yields as new information gets accessible. It requires the calculation to look through the informational collection and search for examples or likenesses and controlling or changing the framework as needs be.

From a machine learning perspective, malware detection can be seen as a problem of classification, or unknown malware types should be classified based on certain properties identified by the algorithm. However, after training a model on a large dataset of dangerous and benign files, we can simplify this problem to classification. This challenge may be cut down to classification just for known malware families; with a small number of classes, to one of which the malware sample undoubtedly belongs, it is easier to identify the right class, and the result is more accurate than using algorithms. This section provides theoretical background on all of the methodologies employed in this research with algorithms. In this section, the theoretical background is given for all the methods used in this project.

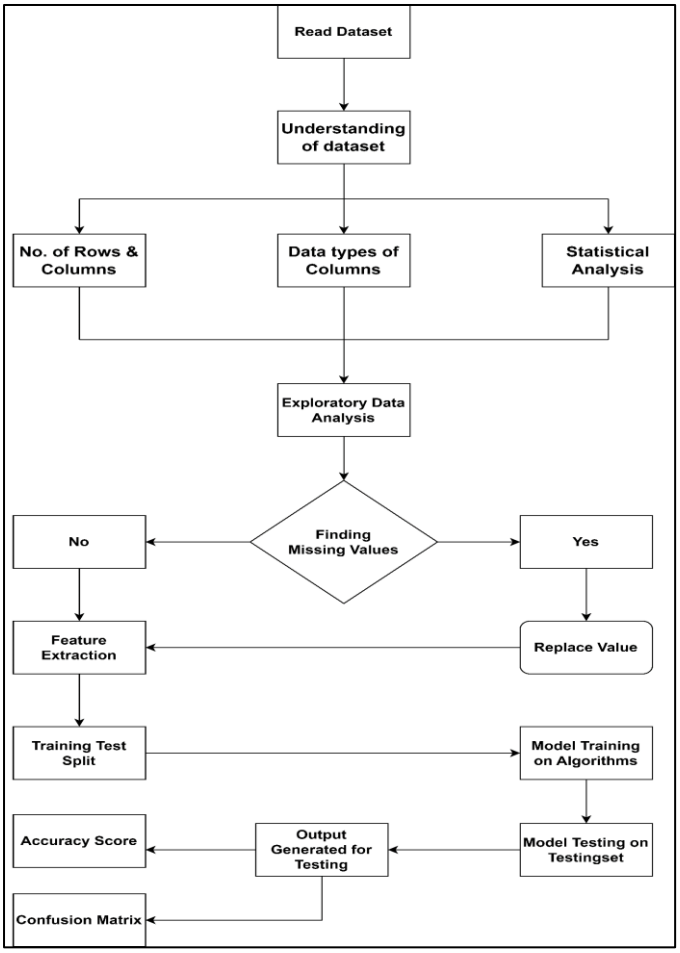

Fig -4: Dataflow Diagram

- **Reading the Dataset:**

The malware dataset was loaded into the program using read_csv () function of Pandas library which returns the object of Data Frame class.

- **Initial understanding of the dataset:**

The Dataset was understood superficially using various functions and properties of the Data Frame like shape which show no of rows and no of columns of the dataset, Info property show the detailed information of the dataset column, types is used to observe the data types of the columns. The sample of records of the dataset can be viewed by the function head() and tail() of the Pandas library that show the top and bottom n number of records.

- **Finding missing values in the dataset:**

The missing values are those column data which are blank or without any value. This may create wrong generation of output when ML algorithms are applied to the dataset. SO, finding the missing values and replacing them (if missing values are found) with proper data is an important activity in a data science project. Fortunately, our dataset does not have any missing values so this step of program implementation is eliminated.

- **Statistical analysis on dataset:**

The Statistical analysis on the dataset shows us more insights of the data for the numeric columns. It also helps us to understand the range of numeric observation spread across. Pandas library provides a function called describe() to perform the stat. analysis on the dataset. This function provides us the statistical output on the dataset with respect to following parameters.

- **Exploratory Data Analysis**

One of the most critical tasks in a data science or machine learning project is EDA. This analysis gives a detailed and deeper understanding of the dataset. The exploratory data analysis can be performed by various means like Distribution Analysis, Frequency Analysis or Categorical Frequency Analysis. In this step we generally visualize the data from the dataset to generate the graphs or chart mostly the bar graphs for displaying distribution analysis, line charts for frequency analysis and box plots for showing the quartiles and also the outliers of the dataset. Outliers are those observations in the dataset which far away from the normal range of the data. We generated bar charts and lines charts to showcase our data in the form of visualized output.

- **Feature Selection**

For performing the Machine Learning model training all features of the dataset are not necessary. We need to select only those features from our dataset necessary for developing the detection and prediction models. The feature selection is a process of eliminating the features not required for model training. We have selected only those features which are appropriate for training the model and for detecting the Benign or Malware attack.

- **Splitting data set into training and testing Set**

This step of machine learning splits the dataset into two parts as training set and a testing. The training set is always larger than the testing set. The ratio of training set and the testing set was 80:20 which means 80% of the dataset was used to train the model and 20% of the dataset was used to test and evaluate the trained model.

- **Model Training:**

We used 4 ML algorithms in our project as Naïve Bayes, Decision Tree, K Nearest Neighbor and Support Vector Machine. The functions required to use these algorithms come from sklearn library. The training set was supplied to each of the algorithms to train the model. Models using different algorithms give different accuracy and errors in detecting and predicting the output label.

- **Model Evaluation:**

The sklearn metrics package provides us various types of output metrics to evaluate our ML trained model. There are various types of metrics generated by the trained model on the testing set like accuracy percentage in prediction, confusion matrix, precision, recall, F1 Score and support.

❖ **Comparative analysis of trained models:**

We used testing set which is the 20% parts of our dataset to test each model developed from all 4 algorithms. The accuracy percentage generating by each model on the testing set was captured and presented in a tabular form and also visualized using bar charts.

CONCLUSION

Security teams may benefit from machine learning in a number of ways, including the ability to detect and predict malware. By analyzing enormous amounts of data, machine learning algorithms may discover patterns and anomalies that are hard to identify using traditional methods. This is especially useful for locating fresh malware that is resistant to detection by more traditional methods. Malware detection and prediction have been fundamentally changed by machine learning, which now offers very precise and sophisticated methods that may significantly improve the efficiency and efficacy of security teams in detecting and thwarting malware assaults.

We provided a defense mechanism that assessed four ML algorithm methods for malware detection and selected the best one. The findings indicate that, as compared to alternative classifiers, there is a growing interest in ML algorithm solutions for malware detection. We provided a defense mechanism that assessed four ML algorithm methods for malware detection and selected the best one. In comparison to other classifiers, the findings suggest that NB (62.47%), DT (97%), KNN (93.71%), and SVM (72.21%) fared well in terms of detection accuracy. The performance of the NB, DT, KNN, and SVM algorithms in detecting malware in a specific dataset was compared. In this experiment, we compared the detection accuracy of a machine learning (ML) classifier that employed static analysis to extract features from PE data to that of two other ML classifiers. Machine learning algorithms can now distinguish between harmful and benign data as a consequence of our work.

The accuracy of the DT machine learning approach was the greatest (97%) of any classifier we tested. Static analysis based on PE information and properly selected data showed promise in experimental findings, in addition to possibly giving the best detection accuracy and accurately characterizing malware. The fact that we don't have to do anything to determine whether data is malicious is a major benefit. Using the Kaggle dataset, the four ML models (NB, DT, KNN, and SVM) were trained, tested, and their efficiency evaluated.

FUTURE SCOPE

The future scope of malware detection and prediction using machine learning is vast, with numerous opportunities to enhance the effectiveness and efficiency of security teams. As the sophistication of malware threats continues to increase, there is a pressing need for advanced detection and prediction systems that can keep up with the rapidly evolving threat landscape. One area of future research is the development of machine learning algorithms. These algorithms, such as Decision tree, KNN, SVM and Naïve Bayes, can analyze complex features of malware and improve their performance over time. This approach has the potential to significantly increase the accuracy of malware detection and prediction.

The integration of machine learning with other cyber security measures, such as intrusion detection systems and firewalls, is an area of growing interest. By combining these technologies, security teams can create a multi-layered defense against malware threats, providing comprehensive protection against a wide range of attacks. This approach can also help to reduce the number of false positives and improve the efficiency of security teams, enabling them to focus on the most critical threats. Model of machine learning in malware attribution to identify the origin and source of malware attacks. Application of natural language processing (NLP) techniques in malware detection and prediction. Use of machine learning in threat intelligence: The application of machine learning in threat intelligence can help to identify emerging threats and predict the likelihood of future attacks.

Integration of human expertise with machine learning: Combining human expertise with machine learning algorithms can help to improve the accuracy of malware detection and prediction. Development of hybrid models: The development of hybrid models, combining both supervised and unsupervised learning, can improve the performance of malware detection and prediction systems. Real-time malware detection: The development of real-time malware detection systems using machine learning can help to quickly identify and respond to malware threats, reducing the risk of damage to systems and networks. Advancements in data privacy and security to ensure the protection of sensitive data used in machine learning models for malware detection and prediction.

ACKNOWLEDGEMENT

With great pleasure we hereby acknowledge the help given to us by various individuals throughout the project. This Project itself is an acknowledgement to the inspiration, drive and technical assistance contributed by many individuals. This project would have never seen the light of this day without the help and guidance we have received. We would like to express our profound thanks to Prof. Ms. R. S. Badre for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We would also thank the faculties of the Department of Computer Science & Engineering, for their kind co-operation and encouragement which help us in completion of this project. We owe an incalculable debt to all staffs of the Department of Computer Science & Engineering for their direct and indirect help.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities. We extend our heartfelt thanks to our parents, friends and well-wishers for their support and timely help. Last but not the least; we thank the God Almighty for guiding us in every step of the way.

REFERENCES

[1] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian event classification for intrusion detection," in 2003 IEEE Symposium on Security and Privacy (S&P), pp. 148-160, 2003.

[2] M. Christodorescu, S. Jha, S. Seshia, D. Song, R. Bryant, and S. Idika, "Semantics-aware malware detection," in 2005 IEEE Symposium on Security and Privacy (S&P), pp. 32-46, 2005.

[3] A. Kolter and M. Maloof, "Learning to detect and classify malicious executables in the wild," Journal of Machine Learning Research, vol. 7, pp. 2721- 2744, 2006.

[4] Y. Tian, J. Yang, and Y. Han, "A new method for malware detection using SVM," in 2007 IEEE International Conference on Communications (ICC), pp. 4208-4212, 2007.

[5] D. Dagon, G. Gu, C. Lee, and W. Lee, "A taxonomy of botnet structures," in 2007 ACM SIGCOMM Workshop on Large-scale Attack Defense (LSAD), pp. 33-42, 2007.

[6] X. Hu, Z. Wu, X. Fu, and Y. Zhang, "Fast malware classification by automated behavioral graph matching," in 2010 IEEE International Conference on Communications (ICC), pp. 1-6, 2010.

[7] J. Wang, Y. Xu, J. Li, and Y. Chen, "Malware detection using dynamic analysis on cloud," in 2013 9th International Conference on Computational Intelligence and Security, pp. 344-348, 2013.

[8] X. Huang, Z. Xu, and X. Zhu, "A feature selection approach based on fisher score and mutual information for malware detection," in 2013 IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, pp. 557-562, 2013.

[9] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Computers & Security, vol. 45, pp. 100- 123, 2014.

[10] F. Ahmad, M. A. Imran, and M. A. Zia, "Machine learning techniques for malware detection: A comparative study," in 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), pp. 611-616, 2015.

[11] S. S. Patel and S. H. Patel, "Malware detection using machine learning algorithms," in 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 1535-1539, 2016.

[12] Y. Wang, C. Zhou, and K. Ren, "Malware detection using ensemble learning with deep belief networks," IEEE Transactions on Information Forensics and Security, vol. 12, no. 9, pp. 2207-2216, 2017.

[13] X. He, Y. Zhang, L. Nie, and X. Liu, "Practical malware detection with feature selection by feature importance," IEEE Transactions on Information Forensics and Security, vol. 12, no. 9, pp. 2145-2156, 2017.

[14] M. F. Hasan, S. S. Hasan, S. M. Hasan, and M. A. Hoque, "Malware detection using machine learning algorithms: A review," in 2018 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 524-529, 2018.

[15] Y. Kim, J. Kim, and J. Han, "Malware detection using convolutional neural networks and ensemble methods," IEEE Access, vol. 6, pp. 24063-24073, 2018.

[16] C. Zhang, H. Cao, J. Wu, Z. Zhang, and C. C. Tan, "Malware detection with attention-based convolutional neural networks," IEEE Access, vol. 7, pp. 66145-66154, 2019.

[17] S. Kumar and R. Yadav, "Malware detection using machine learning techniques: A systematic review," in 2019 IEEE 4th International Conference on Computing, Communication and Security (ICCCS), pp. 291-296, 2019.

[18] J. A. L. Pérez, P. F. Sanz, and E. Caballero, "Deep learning for malware detection using convolutional neural networks," in 2020 IEEE Conference on Games (CoG), pp. 1-8, 2020.

[19] M. Khalid, M. A. Imran, and M. A. Zia, "Malware detection using machine learning and deep learning techniques: A survey," Journal of Network and Computer Applications, vol. 168, pp. 102760, 2021.

[20] H. Jin, Y. Zhang, X. Chen, J. Li, and X. Liu, "Malware detection with machine learning and feature selection," IEEE Transactions on Cybernetics, vol. 51, no. 2, pp. 609-619, 2021.

[21] Z. Li, M. Li, J. Wu, C. Zhu, and Z. Zhang, "Malware detection using GNN with edge-centric and node-centric feature learning," IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 2185-2198, 2021.

[22] H. Zeng, F. Xie, Y. Wang, X. Zeng, and Y. Zhang, "Malware detection using a hybrid approach of deep learning and graph embedding," IEEE Access, vol. 9, pp. 35096-35105, 2021.

[23] S. S. M. Al-Shaibani, A. Z. Kouzani, and D. Nasrulloh, "Malware detection using deep learningbased approach: A review," in 2021 International Conference on Data Science, E-learning and Information Systems (Data'21), pp. 1-6, 2021.

[24] S. Saha, S. Das, and S. Dasgupta, "A deep learning approach for malware detection using hybrid convolutional neural network," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1272-1286, 2021.