MALWARE DETECTION & PREVENTION USING MACHINELEARNING ALGORITHMS

[1]**V. Padmaja,** Assistant Professor, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[2]**K. JyoshnaPriya,** Assistant Professor, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[3]**M. Bhavana,** UG Student, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[4]**V. Jyoshna Vaishnavi,** UG Student, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[5]**T. Lalitha Kumari,** UG Student, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India
[6]**Y. Krishna Mohan,** UG Student, Department of Computer Science and Engineering, DVR & Dr. HS MIC College of Technology, Kanchikacherla, Andhra Pradesh, India

Abstract:
Malware is the acronym of Malicious Software. It has become a big threat in today's computing world. The threat is increasing with a greater pace as the use of Internet in our day to day activities is growing extensively. The number of malware creators and websites distributing malware is increasing at an alarming rate which attracts researchers and developers to develop a better security solution for it. Polymorphic malware is a new type of malicious software that is more adaptable than previous generations of viruses. Polymorphic malware constantly modifies its signature traits to avoid being identified by traditional signature-based malware detection models. To identify malicious threats or malware, we used a number of machine learning techniques. Developing an efficient malware detection technique is still an ongoing research. Understanding malware, features of malware, analysis methods and detection techniques are the prerequisites of malware research. In this paper, we have studied a few past research works based on API calls, N-Grams, Opcodes features used in malware detection. A detailed fundamental concept of malware detection is also presented in this paper.

1. Introduction:

The definition of malware or malicious software is as follows. Malicious software is a program designed to intrude and damage a computer system & information without the owner's knowledge and permission, which is a serious threat to the security of computer systems from last few decades. With the fast advancement and development of the web, malware has turned out to be one of the major digital dangers in nowadays.

Antivirus tools are unable to provide the necessary security due to the growing diversity of malware in use today, which leads to the hacking of millions of hosts. However, because to the extensive availability of attacking tools on the Internet, the skills needed for malware production is also becoming less necessary.

In order to evaluate if a particular piece of software or network connection poses a security risk, malware detection modules must analyse data they have gathered and been educated with. Consider a machine learning system that can describe the underlying principles of the patterns it has discovered clearly. Using feedback on how well they performed on earlier tasks and using that information to make improvements, algorithms taught by machine learning systems can enhance their capacity to anticipate.

1.1 Types of Malwares:

Malware comes in many different forms, including viruses, worms, Trojan horses, botnets, rootkits, adware, scareware, spyware, ransomware, backdoors, key loggers, and erroneous security software as well as browser hijacker.

1.) **Virus**:

 DOI: 10.36893.JK.2023.V13I04N16.0077-0084

This is the software type that is easiest to use. It is essentially any piece of software that is loaded and launched without the client's consent, copies itself, or contaminates and modifies other software. This is frequently distributed via PCs exchanging files or programmes.

2.) **Worm**:
Standalone software that copies itself and deletes files and data from the computer. Worms, in contrast to viruses, don't require the user to take any more steps to duplicate and carry out.

3.) **Trojanhorse**:
It is frequently referred to as a "Trojan," which is harmful software that has been intentionally embedded in a system or application while seeming to be a helpful file or program.

4.) **Rootkit**:
The term "rootkit" refers to malicious software that is intended to access and operate a computer remotely while evading detection by security software or users.

5.) **Spyware**:
Spyware is a type of software that tracks user activities and collects sensitive data, including keystrokes, account information, financial information, credit card number, email address, and frequently visited web pages. When free and potentially harmful software is downloaded and installed without the user's awareness, it enters the system.

6.) **Backdoors**:
Backdoors are similar to trojans or worms, with the exception that they open a "backdoor" onto a computer system, creating a network link via which viruses or SPAM may be transferred or entered by intruders or other malware.

7.) **Browser Hijacker**:
This type of malware reroutes regular browser search operations and displays the results that the virus's creators want us to view. Making money off of the user's web browsing is the main goal of browser hijackers.

8.) **Keyloggers**:
Every keystroke you make on your computer is recorded, allowing the keylogging program's creator to collect your log-in details and other private data.

9.) **Scareware**:
Scareware is a malicious application that is disguising itself as free antivirus software, a trial version, or some other free online dangerous ploy. It enters the system when a user installs phoney security software, accesses a malicious website, or opens attachments.

1.2 Malware Propagation:

A computer or mobile device can be infected by malware in a variety of ways, including through infected email attachments, file sharing, instant messaging, the use of third-party applications while social networking, the use of pirated software, and the usage of USB and other portable media. Malware can harm the system's boot sector, installed software, data files, and even the system BIOS after entering the system, which causes the system to behave abnormally.

The primary goal of all malware developers is to install and spread their programmes on as many computers or mobile devices as they can. This may be accomplished either via the use of social engineering or by secretly infecting a system. These approaches often involve steps to get around antivirus software that has been installed on such devices and are frequently used in tandem.

1.3 Concealmentstrategies:

To prevent being found by anti-malware software, malware creators employ concealment techniques. Some malware is modified for both proliferation and transmission as a result of these camouflage techniques. It might be challenging to recover a virus's signature for malware detection since some malware encrypts both themselves and their malicious actions. The following list of concealing tactics includes a few.

1.) **Code Obfuscation:**

 DOI: 10.36893.JK.2023.V13I04N16.0077-0084

ISSN: 2278-4632
Vol-13, Issue-04, No.06, April : 2023

In order to prevent outsiders from learning about the code's basic logic, this strategy obscures the program's main logic. Obfuscated malware is incomprehensible until it is launched, and it has destructive capability.

Developers utilise this approach to avoid signature-based detection algorithms from identifying their malware by adding superfluous jumps, dead-code insertion, the usage of garbage instructions, register reassignment, instruction replacement, subroutine reordering, and code integration/transposition.

2.) **Code encryption:**

By using this technique, malware is encrypted and comprises of malicious programmes, keys, and encryption and decryption methods. Every time, the attacker creates a brand-new malware version using a fresh encryption technique and key. Since the decryption technique is constant, there is a larger chance of being discovered.

This approach aims to prevent static analysis and slowing down the inquiry process. In 1987, CASCADE was identified as the first malware that used encryption.

3.) **Polymorphic strategy:**

Polymorphic malware is designed to change its look each time it is executed while preserving all of its original code.

To prevent signature-based detection, this approach can create millions of decryptors simply modifying instructions in the next variant of the virus. In each execution, a new decryptor is constructed and joined with the encrypted malware body to form a new variant of the infection. Although a vast number of alternative decryptors can be built, signature-based techniques can still detect malware by identifying the original software via emulation technique.

4.) **Metamorphic strategy:**

Metamorphic malware alters itself in such a way that the new instance bears no resemblance to the original. Instead of generating a new decryptor, a new instance or body is generated with the same activities. The virus lacks a coding engine, and each transmission results in automated modifications to the malware source.

1.4 Malware Symptoms:

Malware infected computers may exhibit any of the following symptoms:

- Computer processing speed is slow.
- Slow web browser performance.
- Problems with network connectivity.
- High CPU consumption, as well as the appearance of unusual applications, files, or icons.
- Programs that are running, stopping, or rearranging themselves.
- System freezes or crashing.
- Files are automatically modified or deleted.
- Emails/messages are sent automatically without the user's consent.

1.5 Malware Analysis:

Malware analysis can be done in the following three ways:

1.) **Static Analysis**:

Static analysis refers to analysing harmful code without running it. Techniques in this category often evaluate the code-structure statically for infection attributes using a pre-defined set of known assail signatures without running the sample. The detection patterns used in static analysis include byte-sequence, n-grams, opcode (operational code) frequency distribution etc. Although static analysis approaches are capable of quickly detecting malware in a variety of applications and offer no danger of infection when studying malware, they require a large number of pre-defined signature datasets. Moreover, they have runtime overhead and cannot distinguish between variants of known or obscure malwares and zero-day intruders.

2.) **Dynamic Analysis:**

DOI: 10.36893.JK.2023.V13I04N16.0077-0084

Dynamic analysis refers to the process of analyzing the behaviour of malicious code as it is being executed. Dynamic analysis is carried out in a controlled environment by means of a virtual machine, emulator, simulator, sandbox, and so on. Before running the malware sample, the necessary monitoring tools are installed and enabled. Though dynamic analysis techniques are independent of malware source-code detect unknown and zero-day malware instances, they require more resources and high computational cost and false positive rate.

3.) **Hybrid Analysis:**

This approach is designed to overcome the constraints of static and dynamic analysis techniques. It initially analyzes the signature specification of any malware code before combining it with other behavioral data to improve overall malware analysis. Because of this method, hybrid analysis overcomes the constraints of both static and dynamic analysis.

2. Related Work:

2.1 Signature-Based Detection:

It is also known as misuse detection. It keeps the signature database up to date and compares patterns to the database to detect malware. The signatures are generated by analysing the disassembled code of malware binaries. The disassembled code is examined and characteristics are extracted. These characteristics are utilized to create the signature of a certain malware family.

The key advantages of this method are that it can reliably detect known cases of malware while using fewer resources. It is essential to identify malware and focuses on the signature of the assault. The main disadvantage is that it cannot identify fresh, unknown instances of malware because no signature for such malware is available.

2.2 Heuristic-Based Detection:

It is also called as anomaly-based detection. It normally happens in two stages: training and detection. During the training phase, the system's behaviour is studied in the absence of an attack, and a machine learning approach is employed to develop a profile of this normal behaviour. During the detection phase, this profile is compared to current behaviour, and any deviations are regarded as possible assaults.

The benefit of this method is that it can detect both known and undiscovered cases of malware. The downside of this strategy is that it requires data updates. It requires additional resources, such as CPU time, memory, and storage space.Additional disadvantages include a high incidence of false positives and difficulties picking characteristics to learn during the training phase.

3. Proposed System:

We use machine learning classifiers to detect malwares. Classification algorithms comes under the category of supervised machine learning technique, which is the process of categorizing the given set of input data into classes based on one or more variables. Generally these algorithms generate a possibility score to assign the data to a specific category like spam or not,yes or no etc..

In this proposed system we use three types of classifiers that provide more accurate and efficient results than the existing system. the algorithms are as follows:

1.) **MLP Classifier:**

MLPClassifier stands for Multi-layer Perceptron Classifier, which links to a Neural Network by definition. Unlike other classification methods such as Support Vectors and Naive Bayes Classifier, MLP classifiers do classification using underlying neural networks.

It is an artificial neural network feed forward model that translates input data sets to a collection of acceptable outputs. An MLP is made up of numerous layers, each of which is completely linked to the one before it. Except for the nodes of the input layer, the nodes of the layers are neurons with nonlinear activation functions. One or more nonlinear hidden layers may exist between the input and output layers.

2.) **REP Tree:**

 DOI: 10.36893.JK.2023.V13I04N16.0077-0084

ISSN: 2278-4632
Vol-13, Issue-04, No.06, April : 2023

REP Tree stands for "Reduced Error Pruning Tree".Rep Tree is a method for creating a decision tree from a dataset. It is considered an extension of C45 since it improves the pruning phase through the use of Reduced Error Pruning.

A distinct pruning dataset is used by the approach. It examines if a subtree can be replaced by a single node without reducing the classifier's performance on this pruning set for each subtree. As such, the pruning procedure is straightforward, but it is sometimes seen as overly aggressive, in that it may delete sub trees that are truly significant.

3.) AdaBoost:

The AdaBoost algorithm, short for Adaptive Boosting, is a Boosting approach used as an Ensemble Method in Machine Learning. It is named Adaptive Boosting because the weights are re-allocated to each instance, with larger weights applied to mistakenly identified instances. Boosting is used in supervised learning to minimize bias as well as variation. It is based on the notion of progressive growth of learners. Except for the first, each succeeding student is developed from previously developed learners. Simply said, poor learners are transformed into strong ones. With one exception, the AdaBoost algorithm operates on the same principles as boosting.

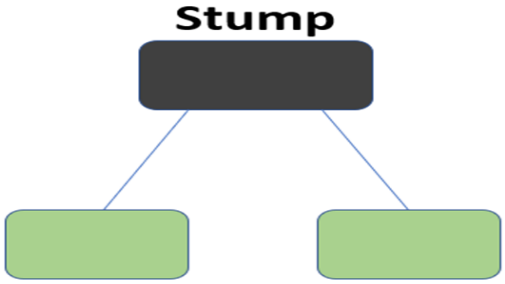

Fig: Decision Stump in Adaboost

Decision trees with one level, or decision trees with only one split, are the most commonly employed estimator with AdaBoost**.** These decision trees are also called as Decision Stumps.

3.1 Data Pre-processing:

The obtained data may contain missing values, resulting in inconsistencies. Data preprocessing converts data into a format that can be handled more readily and efficiently in data mining, machine learning, and other data science operations. To achieve reliable findings, the approaches are often utilized at the early phases of the machine learning and AI development pipeline. Outliers must be deleted, and variable conversion must be performed prior to processing.

3.2 Feature Extraction & Selection:

The databases usually contain tens of thousands of characteristics. As feature counts have increased in recent years, it has become evident that the resulting machine learning model is overfit. To overcome this issue, we created a smaller collection of features from a bigger set. This approach is widely used to retain the same level of accuracy while employing fewer characteristics. The purpose of this study was to improve the current dataset by preserving those that were most useful and removing those that were not useful for data analysis.

Feature extraction assists in reducing the amount of duplicated data in a data source. Finally, data reduction allows the model to be built with less machine effort and accelerates the pace of learning in the machine learning process.

Following the completion of feature extraction, which included the finding of additional features, feature selection was carried out. As it required selecting features, feature selection was a critical procedure for improving accuracy, simplifying the model, and decreasing over fitting from a pool of newly acknowledged attributes.

The performance of malware detection is determined on the feature representation and length used. The feature selection/dimensionality reduction process is carried out in order to get a collection of more discriminative characteristics for improved performance.

3.3 Model Workflow:

DOI: 10.36893.JK.2023.V13I04N16.0077-0084

ISSN: 2278-4632
Vol-13, Issue-04, No.06, April : 2023

1.) **Upload dataset:**
We use this module to upload malware datasets to the application.

2.) **Pre-processing:**
Converts data into a format that can be handled more readily and efficiently in machine learning. Outliers must be deleted, and variable conversion must be performed prior to processing.

3.) **Optimization:**
Feature selection/optimization gives better predictions. It improves accuracy, decrease overfitting problems. To do this we apply feature selection algorithms like PSO and select top most features to predict the model**.**

4.) **Run MLP classifier*:***
Using this module we split the data set in 70:30 ratio for training and testing and the build a MLP training model. This trained model takes test data as input and it predicts the accuracy and output**.**

5.) **Run REP tree:**
Using this module we split the data set in 70:30 ratio for training and testing and the build a REP training model. This trained model takes test data as input and it predicts the accuracy and output**.**

6.) **RunAdaboost *:***
Using this module we split the data set in 70:30 ratio for training and testing and the build a adaboost training model. This trained model takes test data as input and it predicts the accuracy and output**.**

7.) **Evaluationmetrics*:***
True Positives (TP) is the number of samples that were accurately predicted to be "positive."
False Positives (FP)is the number of samples that were incorrectly projected as "positive."
True Negatives (TN) is the number of samples that were accurately predicted to be "negative."
False Negatives (FN) is the number of samples that were incorrectly forecasted as "negative."

a.) **Precision:**
Precision is a measure of how close the computed outcomes are to each other. That means it measures the accuracy of positive results. It tells how good the model is at specific results.
Precision= (true +ve)/(true +ve)+(false +ve)

b.) **Recall:**
Recall refers to the percentage of total relevant results correctly classified by your algorithm. It measures the completeness of positive predictions. It tells the no. of results correctly classified by the algorithm.
Recall= (true +ve)/(true +ve)+(false -ve)

c.) **F1-score:**
F1-score is the harmonic mean of precision and recall. The problems where both the metrics are important can choose f1 score with maximum value.
F1-score = 2*((precision *recall)/(precision +recall))

8.) **Predict malware:**
Using this module the new file can be predicted as malware or benign files as output. if the file is a malware file it will send an alert message to the user.

4. Result &Discussion:

Table : Accuracy Measure

S/N	Algorithm	Accuracy				Precision			
		10 folds	70% split	10 folds	70% split	10 folds	70% split	10 folds	70% split

DOI: 10.36893.JK.2023.V13I04N16.0077-0084

ISSN: 2278-4632
Vol-13, Issue-04, No.06, April : 2023

1	MLP Classifier	97.3	97.1	97.3	97.1	97.3	97.1	97.3	97.1
2	REP Tree	96.9	96.6	97.0	96.7	96.9	96.6	97.0	96.6
3	AdaBoost	92.2	92.7	92.4	92.8	92.2	92.8	92.2	92.8

If display it as malware and regular files as safe files to the user. The project displays the alert message to the administrator[9] as a potentially hazardous file. The algorithms are highly accurate to predict the model.

Malware detection is an ongoing scientific project. Despite the fact that numerous approaches have been developed, we cannot entirely eradicate viruses from the internet-based digital world. Yet we can absolutely reduce its negative impact by constantly working on it to build new, robust, and effective detecting algorithms with a greater accuracy rate. In this context and machine learning approaches are critical.

Furthermore, before using the machine learning model, the best malware properties should be gathered from various forms of malware.

5. Conclusion :

In this paper, we have presented basic definition of malware, its types and its propagation strategy. Further we have presented advantages and disadvantages of different ways the malware can be analysed and detected. The major goal and purpose of this study is to grasp the fundamentals of malware detection, which will be highly useful in future research. We concentrated on realistic malware analysis methodologies and how static and dynamic analysis methods might be combined to construct an effective and resilient malware detection strategy. For varied-sized malware datasets, several machine learning classification techniques may be used. Our approach also includes a comparative analysis of the preceding phase.

Malware, especially those found in mobile and smart devices, has grown in sophistication and regularity in recent years. Despite the fact that various defensive technologies and techniques exist, mechanisms, malware detection, and analysis are difficult problems since malware authors are constantly concealing information in assaults or evolving cyber-attacks to bypass newer security approaches, and some past methods have limited applicability to unknown malwares and scalability concerns.

References:

Ekta Gandotra, Divya Bansal, Sanjeev Sofat, “Malware Analysis and Classification: A Survey”, Journal of Information Security, April 2014, pp: 56-64

Egele, M., Scholte, T., Kirda, E. and Kruegel, C. , “A Survey on Automated Dynamic Malware-Analysis Techniques and Tools”, Journal in ACM Computing Surveys, 44,2012, Article No. 6

Kirti Mathur, Saroj Hiranwal, “A Survey on Techniques in Detection and Analyzing Malware Executables”, International Journal of Advanced Research in Computer Science and Software Engineering, April 2013, Volume 3, Issue 4, ISSN: 2277 128X.

Robiah Y, Siti Rahayu S., Mohd Zaki M, Shahrin S., Faizal M. A., Marliza R., “A New Generic Taxonomy on Hybrid Malware Detection Technique, (IJCSIS)International Journal of Computer Science and Information Security”, Vol. 5, No. 1, 2009.

Matthew G. Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J. Stolfo, “Data Mining Methods for Detection of New Malicious Executables”, in Proceedings of the Symposium on Security and Privacy, 2001, pp. 38-49.

Raja Khurram Shahzad, NiklasLavesson, Henric Johnson, “Accurate Adware Detection using Opcode Sequence extraction”, in Proc. of the 6th International Conference on Availability, Reliability and Security (ARES11),Prague, Czech Republic. IEEE, 2011, pp. 189-195.

Pavithra, J.; Josephin, F.J.S. Analyzing various machine learning algorithms for the classification of malwares. IOP Conf. Ser. Mater. Sci. Eng. 2020, 993, 012099.

 DOI: 10.36893.JK.2023.V13I04N16.0077-0084

I. Santos, J. Devesa, F. Brezo, J. Nieves, P.G. Bringas, "OPEM: A static-dynamic approach for machine-learning-based malware detection", International Joint Conference CISIS'12-ICEUTE'12-SOCO'12, pp. 271- 280, 2012

G.Sai Chaitanya Kumar, Dr.Reddi Kiran Kumar, Dr.G.Apparao Naidu, "Noise Removal in Microarray Images using Variational Mode Decomposition Technique " Telecommunication computing Electronics and Control ISSN 1693-6930 Volume 15, Number 4 (2017), pp. 1750-1756

 DOI: 10.36893.JK.2023.V13I04N16.0077-0084