Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 08, August-2023

A Review on The Malware Detection Systems

L L N V S R K Sai Surya[1], Miriyala Sai Teja[2], A Basanth Reddy[3], Reddyvari Venkateswara Reddy[4],Prince Kumar[5]
[1, 2, 3] Student, Department of CSE (Cyber Security), CMRCET, Hyderabad India
[4] Associate Professor, Department of CSE (Cyber Security), CMRCET, Hyderabad India
[5] Assistant Professor, Department of CSE (Cyber Security), CMRCET, Hyderabad India

***Abstract*— Malware is a type of malware designed to compromise and exploit computer systems, posing a major threat to the digital ecosystem. As the volume and complexity of malware continue to increase, the need for effective malware detection becomes more important. This document provides an overview of malware detection, highlighting their key methods, techniques, and challenges. In this study, we explore the variety of malware detection methods, including signature-based detection based on predefined patterns and heuristic-based detection that identifies different behaviors. Additionally, we are exploring the use of advanced machine learning techniques such as deep learning and hybrid techniques to increase the accuracy of machine detection. The importance of behavior analysis and vulnerability detection in identifying previously unseen threats is also discussed. Although this system is very secure, problems persist in maintaining new signatures and dealing with zero-day issues. As the arms race between attackers and defenders continues, future research directions such as using artificial intelligence for adaptation and prevention became clear. Analyzing the strengths and limitations of existing systems, this article provides a deeper understanding of the changing field of malware detection, helping researchers and practitioners develop more cybersecurity strategies professionally. While these systems offer substantial protection, challenges persist in maintaining up-to-date signatures and addressing zero-day vulnerabilities. As the arms race between attackers and defenderscontinues, future research directions like leveraging artificial intelligence for adaptive and proactive defense mechanisms are highlighted.**

***Keywords*:**
malware, detection systems, cyber security, methods, techniques, threats, machine learning, behavior analysis, signature-based detection, anomaly detection, and heuristics.

I. INTRODUCTION

In today's connected digital environment, the growth of malware poses a threat to the security and integrity of computer systems. Malware; includes viruses, Trojans, and various malware such as bugs designed to exploit vulnerabilities for intrusion, data theft, or damage. Always signature-based detection methods attack to keep up with the rapid evolution and diversity of malware. Therefore, machine learning techniques have become important in building robust and adaptable malware detection systems. With its ability to learn patterns and behaviors from data, machine learning has shown great results in identifying what was known and unseen in the past. Among many machine learning algorithms, the random forest algorithm stands out withits effectiveness in classifying tasks, especially in malware detection. The Random Forest algorithm is a general learning method that combines the results of multiple decision trees to provide more accurate and reliable predictions. This makes it ideal for capturing correlations between features and malware events and allows the system to identify dynamic changes that could indicate malicious intent. This article examines the implementation of the Random Forest algorithm for malware detection, with a particular focus on its implementation programmed in Python. By leveraging Python's rich ecosystem of libraries and tools, researchers and practitioners can develop, train, and evaluate models in a random forest of malware files.

This approach allows security professionals to build robust systems that not only detect known malware but also detect emerging threats. In the remainder of this article,we'll cover in detail the methodology, dataset preprocessing, feature extraction, model training, and evaluation techniques for exploiting the potential of random forests for malware detection. Combining the results of machine learning with various random forest algorithms, this research aims to contribute to the advancement of antivirus and adapt malware to better protect our digital ecosystem.

II. LITERATURE REVIEW

The proliferation of computers, smartphones, and other Internet-enabled gadgets leaves the world vulnerable to cyber assaults. A plethora of malware detection methods have arisen in response to the explosion in malware activity. When trying to identify malicious code, researchers use a variety of big data tools and machine learning techniques. Traditional machine learning-based malware detection approaches have a considerable processing time, but may effectively identify newly emerging malware. Feature engineering may become obsolete due to the prevalence of modern machine learning algorithms, such as deep learning. In this study, we examined a variety of malware detection and classification techniques. Researchers have created ways to use machine learning and deep learning to check samples for malicious intent.

Armaan (2021) illustrated and tested the accuracy of various models. Without data, no application built for a digital platform can perform its function. There are several cyber risks, so precautions must be taken to safeguard data. Although feature selection is difficult when developing a model of any sort, machine learning is a cutting-edge approach that paves the way for precise prediction. The approach needs a workaround that is adaptable enough to handle non-standard data. To find patterns, IT security professionals may use malware analysis tools. The availability of technologies that analyze malware samples and determine their level of malignancy significantly benefits the cybersecurity sector. These tools help monitor security alerts and prevent malware attacks. If malware is dangerous, we must eliminate it before it transmits its infection any further. Malware analysis is becoming increasingly popular as it helps businesses lessen the effects of the growing number of malware threats and the increasing complexity of the ways malware can be used to attack.

Chowdhury (2018) proposed a viable malware detection approach that uses a machine-learning classification technique. We explored whether or not adjusting a few parameters might increase the accuracy with which malware is classified. N-gram and API call capabilities were incorporated into our approach. Experimental evaluation confirmed the efficacy and dependability of our proposed technique. Future work will focus on merging a large number of features to increase detection precision while decreasing false positives. Our Chowdhury approach was superior. At this time, the proliferation of malicious software poses a significant threat to global stability. In the 1990s, as the number of interconnected computers exploded, so did the prevalence of malicious software, which eventually led to the widespread distribution of malware. Multiple protective measures have been created in response to this phenomenon. Unfortunately, current safeguards cannot keep up with modern threats that malware authors have created to thwart security programs. In recent years, researchers' focus on malware detection research has shifted toward ML algorithm strategies. In this research paper, we present a protective mechanism that evaluates three ML algorithm approaches to malware detection and chooses the most appropriate one. According to statistics, the decision tree approach has the maximum detection accuracy (99.01%) and the lowest false positive rate (FPR; 0.021%) on a small dataset.

Malware continues to develop and propagate at an alarming rate. Nur (2019) compared three ML classifiers to analyze and quantify the detection accuracy of the ML classifier that used static analysis to extract features based on PE information. As a group, we trained machine learning algorithms to recognize dangerous versus benign information [24]. The DT machine learning method attained 99% accuracy, as illustrated in Table 2 making it the most successful classifier we examined. This experiment demonstrated the potential of static analysis based on PE information and chosen key data features to achieve the highest detection accuracy and the most accurate depiction of malware.

Malicious programs and their threats, or “malware,” became increasingly common and sophisticated as the Internet developed. Their rapid dispersion over the Internet has provided malware authors with access to a wide varietyof malware generation tools. Every day, the reach and sophistication of malware grows. This study focused on analyzing and measuring classifier performance to better understand how machine learning works. Latent analysis extracted features from the recovered PE file and library information; six classifiers based on ML techniques were evaluated. It was recommended that ML systems be trained and tested to determine whether or not a file is harmful. Experimental outcomes verified that the random forest method is preferable for data categorization, with 99.4 percent accuracy. These results showed that the PE library was compatible with static analysis and that focusing on only a few properties could improve malware detection and characterization. The main benefit is that it is less likely that malicious software will be installed by accident, as users can check a file's validity before opening it.

III. OBJECTIVE

The main purpose of this research is to develop and evaluate a malware detection system using the Random Forest machine learning algorithm in the Python programming environment. This research aims to leverage the power of machine learning to improve the accuracy andefficiency of malware detection using random forests.

Through a random forest operation and multiple learning algorithms, this research aims to achieve the following goals: improvement of statistical accuracy, critical analysis,good action, and expansion for new models. By achieving these goals, this research contributes to the development of malware detection techniques using the capabilities of random forests in the Python programming environment. The findings of this research are expected to provide insights into the development of robust and adaptive malware detection systems that can help protect digital ecosystems from change, andcyber threats.

IV. SYSTEM REQUIREMENTS

Hardware Requirements:

1. Minimum 4GB RAM
2. Hard Disk 500GB
3. Network connected with good bandwidth.
4. Processor: Intel Core i5

Software Requirements:

1. Operating system: Windows 10.
2. Coding Language: Python3
3. Database: CSV File.
4. VS Code

Libraries:

1. Matplotlib
2. NumPy
3. Pandas
4. Seaborn

V. PROBLEM DEFINITION

The problem addressed in this study is to find malware in the digital environment using random forest machine learning algorithms. Due to its ever-evolving nature and necessity, malware poses a threat to computer systems, networks, and data. Signature-based attack techniques are needed to keep up with the rapid evolution of new malware, thus more sophisticated detection techniques are needed.

VI. EXISTING SYSTEM

Many malware solutions do not rely onmachine learning. Here are some key features and their estimated accuracy:

1. Signature-based check:

The signature check involves creating a database of known malware names and comparing them to a database that matches letters or numbers. It is very true for known malware, but quite true for new and unknown malware.

2. Heuristic-based search:

A heuristic to identify malicious behavior by examining the code for specific patterns or actions in malware. This system adapts more easily to new threats, but it can also create security vulnerabilities.

3. Behavior Analysis:

Behavior Analysis examines software or systems for unusual behavior that may indicate the presence of malware. This approach is useful for investigating zero-day attacks and persistent threats.

4. SANDBOXING:

Sandboxing deals with monitoring the behavior of crimesthat have the potential to damage the surrounding area.
It is useful for detecting new and unknown malware butcan be potentially useful.

VII. LIMITATIONS OF EXISTING SYSTEM

Some of these limitations are:

1. Feature Engineering Complexity: Many machinelearning-based malware attempt to extract relevant features from malware samples. Building an effective system requires domain expertise and cannot capture all thecomplex features of advanced malware, resulting in lower accuracy.

2. Bad Data: In real-world data, bad examples often outnumber good ones. Due to the control of most classes (benign software), inconsistent classes can damage the training model and cause poor performance of restricted classes (malware).

3. Generalizing Zero-Day Threats: Once machine learning models can be trained on historical data, they will have a hard time detecting any new malware (zero-day threats) that exhibits behavior not found in their teaching materials.
The lack of prior knowledge about these threats limits the ability of models to accurately identify them.

4. Attacks: Malware authors can deliberately modify their code or behavior to avoid detection by machine learning models. Malicious attacks can lead to a cat-and-mouse situation, with attackers constantly tweaking their malwareto evade detection systems.

5. High False Positive Rate: The sensitivity model can result in a large number of false positives where benign software is misclassified as malware. This can result in user frustration and reduced performance.

6. Computational requirements: Some machine learning models, including combinatorial algorithms such as random forests, may include requirements during training and inference.

7. Unlimited interpretation: While random forests can yield important points, it can be difficult to understand the real logic behind decision-making patterns. Interpreting mixed patterns can compromise clarity and confidence in findings.

8. Lack of stability in malware updates: As malware evolves rapidly, existing detection systems will quickly become outdated.

VIII. ARCHITECTURE

Using machine learning algorithms such as Random Forest, the architecture of a malware detection system includes multiple components working together to identify and classify malware sources. Here is an overview of the architecture:

1. Data Collection and Processing:

Collects good and bad models from a variety of sources, including data storage, network connections, and consumer products.

Pre-process data and use data attributes, API calls, network behavior, transaction state, etc. Remove related features such as Performing data normalization, transformation, and cleaning to ensure training and test data is consistent and valid.

2. Feature Extraction:

Extract key features from previous data.

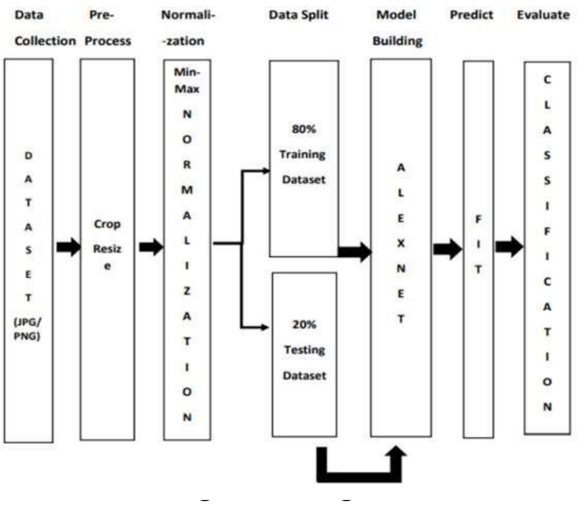
These features should reflect the static and dynamic properties of the structure.

Static signatures can include data size, entropy, and foreign libraries; dynamic signatures can include API calls and system call lines.

3. Feature Selection and Size Reduction: Performs custom selection to select the most importantfeatures that aid in malware detection. These steps helpimprove the performance of the model and reduce thecomplexity of the computer using techniques such asclustering, correlation analysis, or special selectiontechniques.

4. Dataset Splitting:

Divides the dataset into training, validation, and test sets. This separation allows the model to be trained on one subset, validated in another subset for hyperparameter



5. Model Training:

Training random forest learning models using training data and feature selection.

Use cross-validation in the application set to set parameters such as trees, depth of trees, and minimum number of samples per leaf. The model learns to distinguish between good and bad examples based ongiven characteristics.

6. Model Evaluation:

Evaluate the performance of the training model on test data using metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve (AUC-ROC). assesses the model's ability to expand into new,unprecedented models and detect known and previouslyunseen diseases.

7. Live Detection:

uses a real-time detection system that processes incoming samples and uses training samples to classify them as benign or malicious.

IX. CONCLUSION

The use of machine learning in malware detection is a promising approach to fend off the constantly changing threat landscape. ML algorithms can be used to more accurately identify and categorize both known and unknown malware strains. This project's methodology includes important processes like data collecting, preprocessing, feature extraction, model training, and real-time deployment. The created malware detection system can improve overall security measures through meticulous analysis and performance improvement. However, it's critical to address the drawbacks and difficulties of ML-based methods, such as adversarial attacks and the requirement for ongoing updates. To keep ahead of new malware threats and maintain effective protection for computer systems and networks, ongoing research, and development are crucial.

X. RESULTS

Fig-1 shows the training data set given tothe algorithm and when the testing data is given the results are shown as follows:

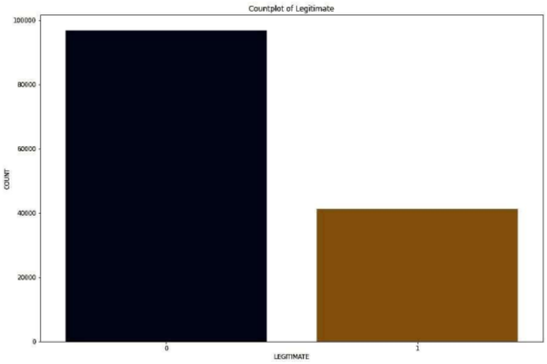


Fig – 1 Results

Fig- 2 shows the confusion matrix when a random forest module is used for finding the accuracy ofthe project.

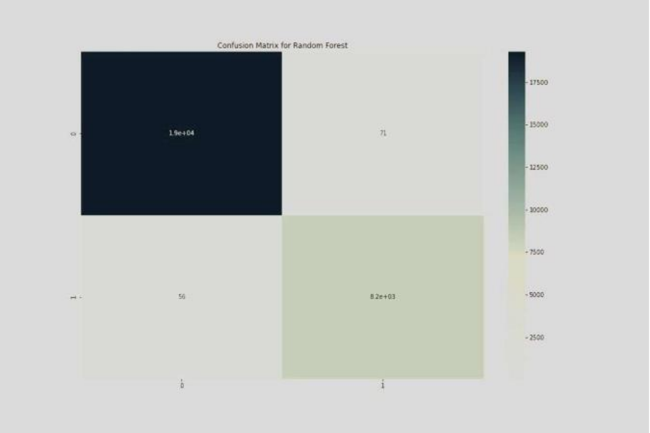


Fig – 2 Confusion Matrix

XI. REFERENCES

[1] Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. (2008). Learning and classification of malware behavior. Journal of Computer Security, 19(4), 587-610.

[2] Kolter, J. Z., & Maloof, M. A. (2006). Learning to detect and classify malicious executables in the wild. Journal of Machine Learning Research, 7(Sep), 2721-2744.

[3] Nataraj, L., Karthikeyan, S., & Jacob, G. (2011). Malware images: visualization and automatic classification. In Proceedings of the 8th International Conference on Information Technology: New Generations (ITNG) (pp. 852-857).

[4] Abdulbasit, A.; Darem, F.A.G.; Al-Hashmi, A.A.; Abawajy, J.H.; Alanazi, S.M.; Al-Rezami, A.Y. An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning. *IEEE Access* **2021**, *9*, 97180–97196.

[5] Feng, T.; Akhtar, M.S.; Zhang, J. The future of artificial intelligence in cybersecurity: A comprehensive survey. *EAI Endorsed Trans. Create. Tech.* **2021**, *8*, 170285.

[6] Sharma, S.; Krishna, C.R.; Sahay, S.K. Detection of advanced malware by machine learning techniques. In Proceedings of the SoCTA 2017, Jhansi, India, 22–24 December 2017.

[7] Chandrakala, D.; Sait, A.; Kiruthika, J.; Nivetha, R. Detection and classification of malware. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–3.

[8] Zhao, K.; Zhang, D.; Su, X.; Li, W. Fest: A feature extraction and selection tool for Android malware detection. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 714–720.

[9] Akhtar, M.S.; Feng, T. Detection of sleep paralysis by using IoT-based device and its relationship between sleep paralysis and sleep quality. *EAI Endorsed Trans. Internet Things* **2022**, *8*, e4.

[10] Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Using convolutional neural networks for classification of malware represented as images. *J. Comput. Virol. Hacking Tech.* **2019**, *15*, 15–28.

[11] Firdaus, A.; Anuar, N.B.; Karim, A.; Faizal, M.; Razak, A. Discovering optimal features using static analysis and a genetic search based method for Android malware detection. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 712–736.

[12] Dahl, G.E.; Stokes, J.W.; Deng, L.; Yu, D.; Research, M. Large-scale Malware Classification Using Random Projections And Neural Networks. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing-1988, Vancouver, BC, Canada, 26–31 May 2013; pp. 3422– 3426.