Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

AI-Powered Malware Detection: Leveraging Machine Learning for Enhanced Cybersecurity

Sagar Rane[1], Sita Yadav[2], Yogita Hambir[2], Anshu Gupta[3], Eshaan Kapoor[4]

[1]Associate Professor, Department of Computer Engineering Army Institute of Technology, Pune, MH, India
[2]Assistant Professor, Department of Computer Engineering Army Institute of Technology, Pune, MH, India
[3]Senior Analyst Department of Banking Deutsche Bank, Pune, MH, India
[4]Research Student, Department of Computer Engineering Army Institute of Technology, Pune, MH, India
Email: sagarrane@aitpune.edu.in

The threat landscape in cyberspace has changed substantially due to the widespread use of digital technologies, creating difficulties for computer systems and sensitive data security. Malware, malicious software intended to interfere with, harm, or allow unauthorized access to computer systems, is still a major problem in the digital sphere. Given how quickly malware is developing, conventional signature-based detection techniques are no longer effective. This research study explores cutting-edge methods to improve cybersecurity by utilizing ma- chine learning algorithms for malware detection. This paper thoroughly examines the field of malware detection utilizing numerous machine-learning techniques. The study digs into the inner makeup of malware, examining its various manifestations, traits, and methods of dissemination. It explores the difficulties experienced by conventional malware detection techniques and clarifies how machine learning models handle these difficulties by providing creative solutions. The paper also emphasizes current developments in the field, such as the use of anomaly detection, ensemble learning, adversarial, and machine learning to strengthen malware detection systems. It objectively assesses the benefits and drawbacks of current machine learning-based malware detection techniques, highlighting new developments and unresolved problems.

Keywords: Malware Detection, Machine Learning, Cyber- security, Deep Learning, Adversarial Machine Learning.

1. Introduction

Detecting malware within portable executable (PE) files presents a critical challenge in contemporary cybersecurity. PE files, commonly utilized in Windows operating systems, often

conceal malicious code through sophisticated obfuscation techniques, necessitating innovative detection methods. In this introduction, we delineate the imperative of employing machine learning (ML) models, including Decision Trees, Random Forest, Linear Regression, Logistic Regression, and K-means Clustering, for enhancing malware detection within PE files.

Malicious software, or malware, encompasses a spectrum of threats ranging from viruses to ransomware, each posing significant risks to digital systems. Traditional signature-based detection techniques have become inadequate in the face of polymorphic and metamorphic malware variants, which can easily evade fixed signatures. Consequently, there is a pressing need for adaptive and robust detection mechanisms capable of discerning subtle patterns indicative of malware within PE files.

Machine learning offers a promising solution to this chal- lenge. By leveraging ML algorithms, such as Decision Trees, Random Forest, Linear Regression, Logistic Regression, and K-means Clustering, systems can analyze features extracted from PE files and make informed predictions regarding their malicious intent. Decision Trees provide a straightforward yet powerful method for classification, while Random Forest harnesses the collective wisdom of multiple decision trees to improve accuracy and robustness. Linear Regression and Logistic Regression models offer additional insights into the relationships between variables and the likelihood of malware presence. K-means Clustering, on the other hand, facilitates grouping of PE files into clusters based on similarities, aiding in the identification of anomalous or potentially malicious files.

Moreover, the dynamic nature of malware necessitates a proactive approach to detection. Traditional signature-based methods struggle to keep pace with the rapid evolution of malware variants, making them ineffective against emerging threats. Machine learning models, on the other hand, offer the capability to adapt and learn from new data, enabling them to recognize evolving patterns of malicious behavior within PE files. By continuously updating and refining their detection algorithms based on real-time data, ML-based systems can provide a more agile and robust defense against malware infiltration.

Moreover, the emergence of adversarial malware under- scores the importance of resilient detection systems. Adversar- ial training, which involves exposing ML models to adversarial examples during the training process, can enhance the model's ability to detect sophisticated threats within PE files.

Additionally, the interconnected nature of modern digital environments, exacerbated by the proliferation of the Internet of Things (IoT), amplifies the need for sophisticated malware detection mechanisms. As everyday devices become increasingly interconnected, they provide new vectors for malware infiltration, necessitating intelligent and adaptable detection systems capable of securing diverse digital ecosystems. Machine learning models offer the flexibility to scale and adapt to the evolving threat landscape, making them well-suited for safeguarding interconnected networks against malware threats lurking within PE files and other digital artifacts. Through a comprehensive exploration of ML-based approaches for malware detection in PE files, this paper aims to contribute to the ongoing efforts to fortify cybersecurity defenses in the face of evolving threats.

In this paper, we delve into the application of Decision Trees, Random Forest, Linear

Regression, Logistic Regression, K-Means Clustering models for malware detection within PE files. Through a comprehensive review and analysis, we aim to elucidate the efficacy of these ML techniques in fortifying cybersecurity defenses against evolving malware threats in the digital landscape.

A. Problem Statement

"Malware Detection in PE files using several different Machine Learning algorithms. Comparing these models based on parameters such as accuracy, efficiency, and F-1 Score. Finally, after proper comparison, determining the best model."

2. LITERATURE SURVEY

In the constantly changing field of cybersecurity, malware presents a serious threat that necessitates creative methods of detection and prevention. Numerous scholarly articles have made contributions in this field, examining various approaches and strategies. A notable area of concentration is the convergence of malware detection with machine learning, utilising advances in artificial intelligence to improve security protocols.

Due to the widespread use of computers, cellphones, and other Internet-enabled devices, cyberattacks are becoming more frequent. The surge in malware activity has led to the emergence of numerous malware detection techniques. Researchers employ a range of big data technologies and machine learning approaches to try and discover dangerous code. Although they take a long time to process, traditional machine learning-based malware detection techniques can be useful in identifying freshly discovered malware. Because deep learning and other contemporary machine learning methods are so common, feature engineering might eventually become outdated. We looked at a range of malware detection and classification methods in this study. Researchers have developed methods to examine samples for malevolent intent using deep learning and machine learning [1].

The correctness of several models was evaluated and illustrated by Armaan (2021). No application created for a digital platform can function without data [2]. Precautions must be taken to protect data because there are numerous cyber dangers. While creating any kind of model, selecting features might be challenging, but machine learning is a cutting-edge method that opens the door to accurate prediction. A flexible solution that can accept non-standard data is required for this method. We must analyse malware and develop new guidelines and patterns in the form of new malware types in order to control and stop attacks in the future [3]. IT security experts may employ malware analysis tools to look for patterns. The cybersecurity industry benefits greatly from the advent of technology that examine malware samples and assess their level of malignancy. These resources support malware attack prevention and security alert monitoring. If malware poses a threat, we have to get rid of it before it spreads its infestation. Because it helps organisations mitigate the effects of the growing quantity of malware threats and the increasing complexity of the methods in which malware can be utilised to attack, malware analysis is growing in popularity [4].

Malware is still evolving and spreading at a startling rate. In order to assess and measure the detection accuracy of the ML classifier that extracted features based on PE information using static analysis, Nur (2019) compared three ML classifiers. We collectively trained machine

learning algorithms to distinguish between benign and harmful material [5]. T he DT machine learning approach was the most successful classifier we looked at, reaching 99% accuracy. This experiment showed how to get the best detection accuracy and most accurate representation of malware using static analysis based on PE information and selected critical data elements.

The susceptibility of adversarial attacks—which alter input data in order to avoid detection—to Android malware detection technologies. The study points out flaws in the detection systems currently in use and suggests mitigating techniques like adversarial robustness and group learning approaches [6]. Improving the dependability and security of Android malware detection solutions requires an understanding of these issues.

Malware threats that target Internet of Things (IoT) devices are a rising source of concern. In order to safeguard user privacy and data security, the research presents lightweight convolutional neural networks (CNNs) designed for IoT malware detection in conjunction with federated learning. In resource-constrained situations, this technique offers a workable way to detect and mitigate malware risks by utilising the collective intelligence of IoT devices [7]. Adversarial query attacks can compromise malware detection systems that rely on machine learning. In order to adaptively detect and stop adversarial attacks, the research presents MalProtect, a stateful defence mechanism [8] that combines historical profiling with real-time query analysis. By using this method, detection systems become more resilient to changing malware threats.

The challenge of zero-day malware, which presents a significant threat due to its novelty and evasion strategies. The study introduces PlausMal-GAN, a framework based on Generative Adversarial Networks (GANs) [9], to simulate zero-day threats and enhance malware detection systems' adaptability. By generating synthetic malware instances, the framework prepares detection systems for previously unknown threats, thereby strengthening cybersecurity defenses.

Cyber Code Intelligence (CCI) is a technique for Android malware detection that combines artificial intelligence and deep code analysis. Through the use of deep learning methods like Long Short-Term Memory (LSTM) networks, CCI offers an all-encompassing framework for identifying instances of malware, even in the face of constantly changing attack strategies. The study highlights how crucial cutting-edge machine learning techniques are to thwarting threats from sophisticated malware [10].

The creation of malware variations that are evasive and the part perturbations play in avoiding detection. Attackers can produce malware variations [11] that evade standard protection measures by carefully modifying the architecture of the infection. The study looks into how disturbances affect malware detection and highlights the necessity for defence techniques that are more flexible in order to tackle ever-changing threats.

A deep learning-based categorization system for reliable malware identification. Even in the face of intricate obfuscation techniques, the framework's deep learning algorithms enable it to classify malware samples with accuracy. This work advances the field of malware detection by offering a flexible and all-encompassing classification scheme [12] .

FAMD is a framework that uses multifeature analysis to swiftly identify Android malware. FAMD boosts detection efficiency and accuracy by combining a variety of information, such as dynamic behaviours and static qualities [13]. The study confirms the efficacy of the suggested methodology and emphasises the significance of quick malware detection in mobile cybersecurity.

A thorough architecture for defending deep neural networks (DNNs) from malicious malware assaults. The approach improves DNNs' resistance to malicious manipulation by fusing ensemble learning, robust optimisation, and adversarial training [14]. The creation of stronger and more flexible defence systems against changing cyberthreats is aided by this research.

A process that makes use of visualisation tools and deep learning algorithms to identify virus variations. Detailed analysis of malware behaviour and enhanced detection accuracy are made possible by the technique, which makes use of convolutional neural networks (CNNs) and opcode- level characteristics. This study uses cutting-edge visualisation techniques to advance malware analysis approaches [15].

A cutting-edge method for malware identification and classification based on network flow data that makes use of graph neural networks (GNNs). The method improves malware detection accuracy by capturing intricate patterns in network interactions and using graph neural networks (GNNs) [16] to represent network flow data. This research embraces the complexity of network-based malware threats, which helps to improve cybersecurity defences.

3. RESEARCH PROBLEMS

Computers and the interconnected technology they entail are ubiquitous, yet inherently perilous. This ubiquity renders them vulnerable to cybercriminal exploitation, enabling the development and deployment of malicious software aimed at commandeering systems and pilfering data [17]. Despite the widespread usage of computers, consistent and impenetrable protection remains elusive for security experts due to several challenges. Cybercriminals adeptly craft and disseminate harmful code across numerous systems, either to gain unauthorized access or to cause harm. Organizations employ various security measures such as antivirus software, log file analysis, and interaction monitoring to detect behavioral patterns indicative of established risks or attack vectors [18].

The identification of malicious elements within PE files necessitates a multifaceted approach, encompassing both static and dynamic analysis techniques. Static analysis involves decompiling viruses and parsing malware files to uncover concealed malicious strings. Conversely, dynamic analysis entails monitoring the execution of malicious code in a secure environment, such as a virtual machine. While each approach possesses distinct advantages and drawbacks, a comprehensive analysis often entails employing both static and dynamic techniques [19]. Furthermore, optimizing virus detection involves identifying and leveraging less malicious characteristics during their creation, thereby providing analysts with additional time for thorough examination. However, striking a balance between the quantity of characteristics employed and the efficacy of malware detection remains a critical challenge [20].

Addressing the research problem requires identifying potential strategies or algorithms for

effectively detecting malware within PE files. Particularly, mitigating the sharp decline in the number of attributes required for identifying previously unseen malware variants is paramount [21, 22].

By scrutinizing these challenges and exploring innovative detection methodologies, this research endeavors to enhance cybersecurity measures against evolving threats posed by malicious software concealed within PE files.

4. MACHINE LEARNING MODELS

For classification, a variety of machine learning models are available. Supervised and Unsupervised are the two main categories into which these models can be separated. Support Vector Machine (SVM), Random Forest, Decision Trees, and Logistic Regression are a few supervised machine-learning methods. However, several unsupervised machine-learning models—like Principal Component Analysis and K-Means Clustering also exist.

However, from supervised machine learning models, we have chosen Linear & Logistic Regression, Decision Trees and Ran- dom Forest and from unsupervised machine learning models we are using K-Means Clustering.

A. Decision Tree

A decision tree is a popular supervised learning algorithm used for classification and regression tasks. It partitions the input space into regions based on feature values, forming a tree structure. Each internal node represents a decision based on a feature, and each leaf node corresponds to an output label. The algorithm determines optimal splits using criteria such as entropy or Gini impurity to maximize information gain. This process creates an interpretable and visualizable model. The decision-making process at each node is guided by criteria such as entropy or Gini impurity, which quantify the homogeneity of the data within each partition by using the below equation.

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

where

- H(X) represents the entropy of the dataset
- n is the number of classes
- pi is the probability of class i occurring.

B. Random Forest

Random Forest, a popular ensemble learning technique in machine learning, builds numerous decision trees during training. Each tree is constructed using random subsets of the training data and features, reducing the risk of overfitting and enhancing prediction accuracy. Predictions are made by aggregating the individual predictions of each decision tree, commonly through a majority vote for classification tasks and averaging for regression tasks.

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x)$$

where:

- yˆ is the predicted output
- N is the number of trees in the forest,
- Ti(x) is the prediction of the i-th tree for the input x.

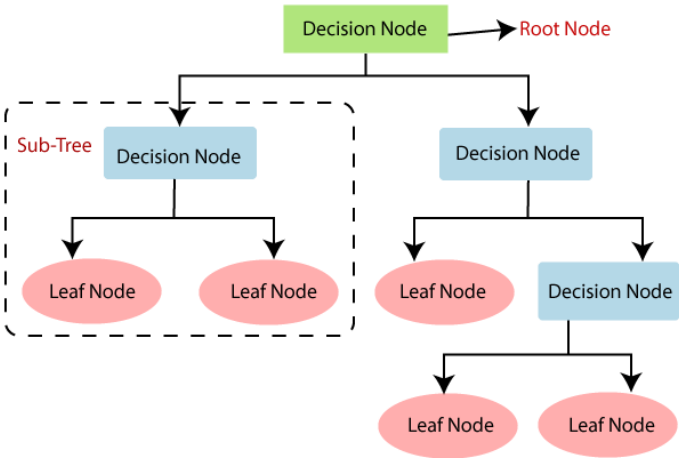

Fig. 1. Decision Tree.

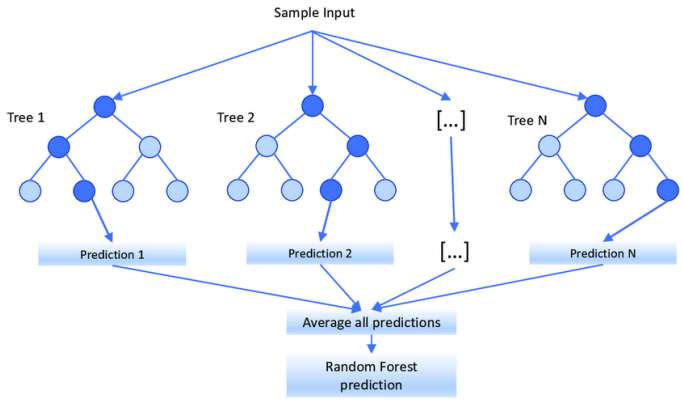

Fig. 2. Random Forest.

C. Logistic Regression

When there are only two possible outcomes for the de- pendent variable in a binary classification task, the statisti- cal technique known as logistic regression is employed. In actuality, logistic regression is a linear model that predicts the probability of a binary event

occurring based on one or more predictor variables, despite its name. Although it is used for classification tasks, the reason it is called ”regression” is because it models the relationship between the predictor variables and the binary outcome. The logistic regression model can be represented by the sigmoid function:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$$

where:

- P (Y = 1 X) is the probability of the dependent variable Y being 1 given the predictors X,
- β0, β1, β2, . . . , βn are the coefficients of the model,
- X1, X2, . . . , Xn are the predictor variables,
- e is the base of the natural logarithm.

D. Linear Regression

A statistical technique called linear regression is used to model the connection between one or more independent variables and a dependent variable. It is assumed that a linear equation, represented by a straight line on a graph, can approximate this relationship. The aim of linear regression is to find the best-fitting line that minimizes the gap between the dependent variable’s actual values and the values predicted by the model.

The equation for simple linear regression is given by:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

where:

- Y is the dependent variable,
- X is the independent variable,
- β0 is the intercept (y-intercept),
- β1 is the slope (coefficient),
- ϵ is the error term (residual).

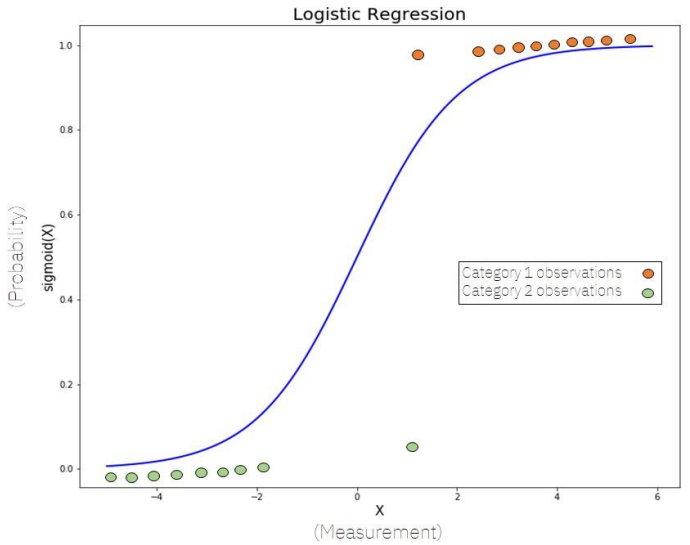

Fig. 3. Logistic Regression.

E. K-means

K-means is a popular unsupervised machine learning tech- nique used for clustering related data points into K distinct clusters without overlap. Initially, K cluster centroids repre- senting cluster centers are randomly initialized. The algorithm then iteratively assigns each data point to the closest centroid using a distance metric, commonly the Euclidean distance. After all data points have been assigned, the centroids are recalculated as the mean of all data points within each cluster. This process continues until convergence, either when a prede- termined number of iterations is reached or when the centroids no longer exhibit significant movement.

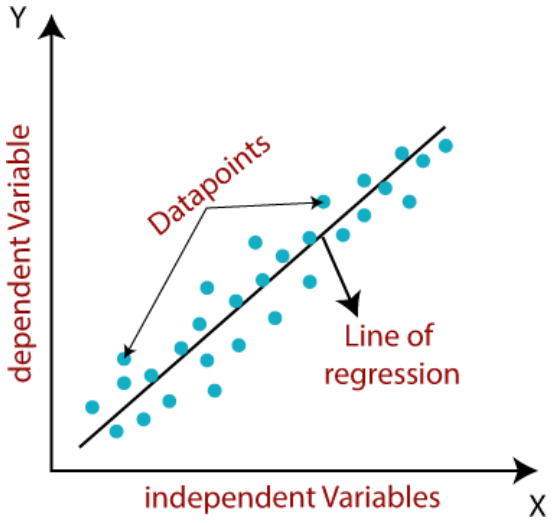

Fig. 4. Linear Regression.

The k-means clustering algorithm involves two main steps:

$$\text{Assignment Step:} \quad c_i^* = \operatorname{argmin}_j \|xi - \mu j\|^2$$

$$\text{Update Step:} \quad \mu_j = \frac{\sum_{i=1}^{n} 1\{ci = j\} \cdot xi}{\sum_{i=1}^{n} 1\{ci = j\}}$$

where:

- $ci*(x)$ represents the index of the closest centroid to data point x.
- μj represents the centroid of cluster j.
- $\| \cdot \|^2$ represents the Euclidean distance between two points.
- $1\{ci = j\}$ is an indicator function that equals 1 if ci = j (i.e., data point xi belongs to cluster j) and 0 otherwise.

5. METHODOLOGY

An overview of our machine learning-based malware detection process is shown in this figure. This approach involves a number of processes, such as locating engaging datasets for the purpose of training a classifier, identifying complex malware, and choosing features for the model. A more detailed explanation of the methodology used in this study may be found below. The figure below illustrates the suggested approach.

A. Dataset

The dataset that was chosen was obtained from the Kaggle repository. We constructed this training set by combining native and non-native features that were taken from Windows applications. The file contained 1,38,047 samples in total, of which 41,323 were safe and 96,724 malicious. There were 57 features mentioned, one of which was a label column that said whether or not the file is dangerous. The study only made use of the Kaggle data. This bundle contained a large number of files containing log data that was compromised by several types of malware. The recovered log data can be used to train a wide variety of models. It was discovered that five distinct families of malware had infected the samples. There were 1,38,047 rows in the data and 57 columns.

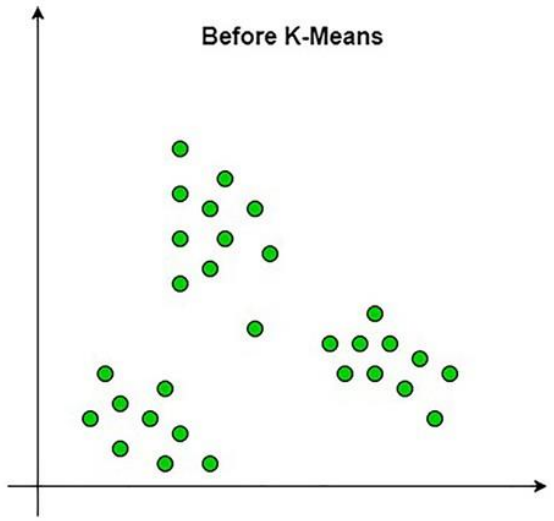

Fig. 5. K Mean

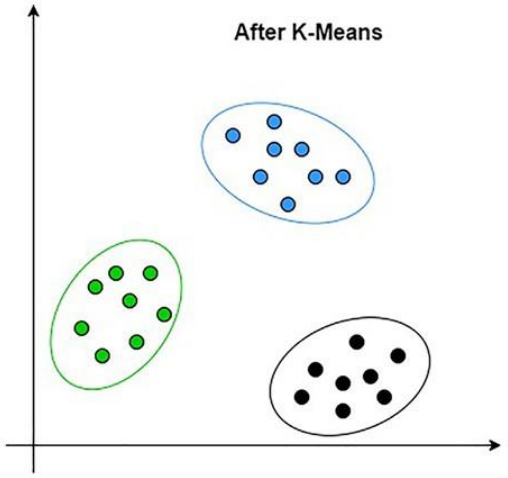

Fig. 6. K Mean

B. Exploratory Data Analysis (EDA) and Analysis

In contemporary datasets, features in the tens of thousands are typical. As a machine learning model's characteristic count rises, the problem becomes more apparent.

C. Pre-Processing

The files themselves were unprocessed executables, and the data were saved in the file system as binary code. They were ready when we started our investigation. A virtual machine (VM) or protected environment was necessary to unpack the executables.

D. Features Extraction

Static and dynamic analysis can be employed separately or in combination to extract features from malware binaries. Static analysis examines malware files without running them; features are primarily derived from PE headers or by break- ing down executable files and examining assembly language. Dynamic analysis involves running an executable file in a control environment and monitoring its behaviour, such as identifying system calls that are made dynamically and do not involve code, attempting to connect to external networks, and attempting to modify registry entries. Malware analysis makes use of all these trails and actions. [23]

E. Features Selection

Selecting which features to use comes next once new features are found through the feature extraction process. Feature selection is the process of choosing characteristics from a group of recently identified qualities. It is crucial for increasing model accuracy, simplifying the model, and lowering overfitting [24]. From the 57 features present, final 14 features were selected for determining whether the file present is Malware or not. Researchers have employed a range of feature classification techniques in an effort to detect malicious software. This work makes extensive use of the feature rank technique because its main goal is to create models for virus detection [25].

Following the feature selection process, Figure 8 makes it evident that 70.1% of the data points denoted by value 0 are in our dataset were classified as malicious, while the remaining 29.9% denoted by 1 were classified as non-malicious.

6. RESULTS AND DISCUSSION

The results that have been observed in the case of supervised machine learning models are much better as compared to those obtained in the case of unsupervised machine learning models. Among supervised machine learning models, the Random Forest Model is the most accurate. The malware detection result and PE features used have been shown in Figure 7. The Bar Graph for comparison of accuracy of different models are shown in Figure 8. Confusion Matrix of all the models have been shown in Figure 10, 11 and 12.

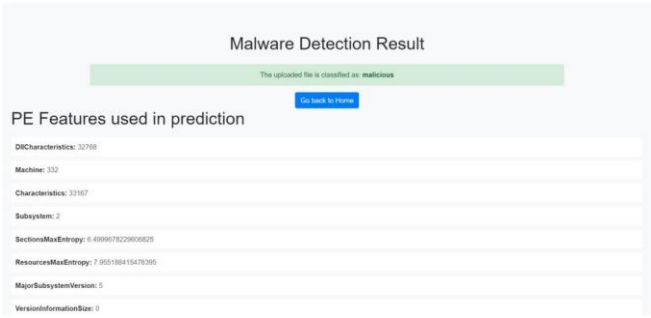

Fig. 7. Malware Detection

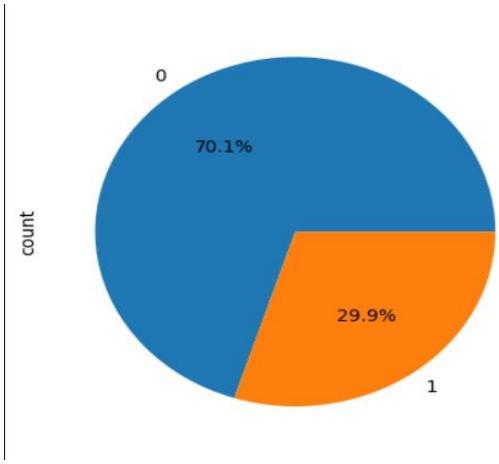

Fig. 8. Pie Chart

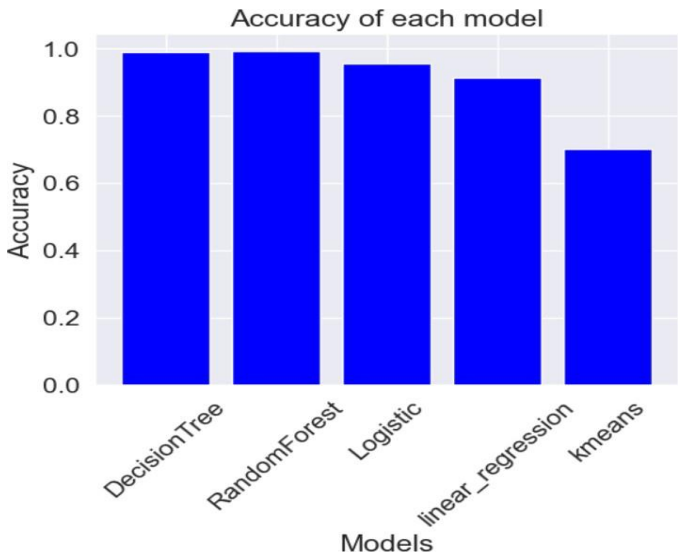

Fig. 9. Bar Graph

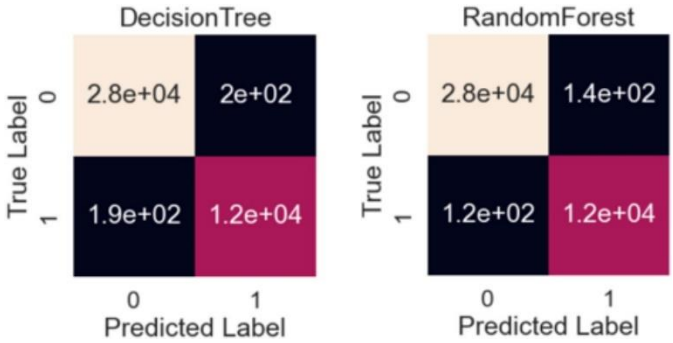

Fig. 10. Confusion Matrix 1 and 2 Model

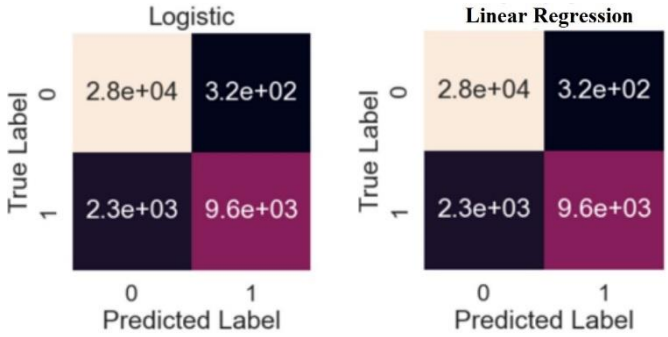

Fig. 11. Confusion Matrix 3 and 4 Model

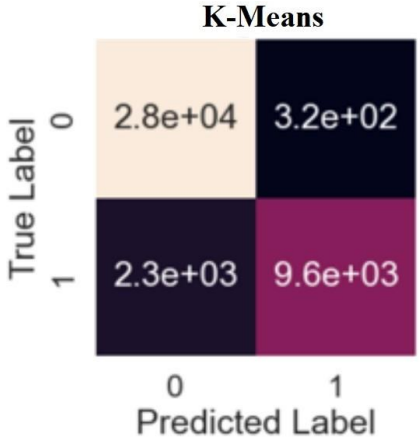

Fig. 12. Confusion Matrix 5 Model

Accuracy Values :

- Decision Tree : 98.97%
- Random Forest : 99.36%
- Logistic Regression : 91.26%
- Linear Regression : 95.56%
- K-Means : 70.07%

With an accuracy of 99.36%, Randam Forest beat the other models, proving its resilience in managing intricate data and attaining great precision. With a respectable accuracy of 91.26%, logistic regression demonstrated its promise for malware instance classification. With an accuracy of 95.56%, Linear Regression proved to be a useful tool for predicting malware behaviour. Nonetheless, K-Means demonstrated a significantly reduced accuracy of 70.07%, indicating its limits in precisely classifying malware cases within this particular environment. These results emphasise how important it is to use Random Forest for efficient malware identification.

7. CONCLUSION & FUTURE SCOPE

In this paper, we used machine learning algorithms to tackle the urgent problem of virus detection within PE files. Computers are everywhere, which makes them susceptible to cyberattacks and calls for strong security protocols. Our research concentrated on using supervised and unsupervised machine learning approaches to detect malware effectively. We assessed five distinct machine learning models—Decision Tree, Random Forest, Logistic Regression, Linear Regression, and K-Means—through extensive experimentation. Our findings demonstrated Random Forest's supremacy in precisely detecting malware instances, with an astounding accuracy of 99.36%. Promising findings were also shown using logistic regression and linear regression, with accuracies of 91.26% and 95.56%, respectively. Nevertheless, K-Means only produced an accuracy of 70.07% when it came to correctly categorising malware cases in our dataset.

FUTURE SCOPE :

- Feature Engineering: Improving the methods of feature extraction and selection over time can help machine learning models operate more effectively. Investigating cutting-edge characteristics obtained from both static and dynamic analytic methods may raise malware detection systems' overall accuracy.

- Ensemble Techniques: By combining the advantages of several algorithms, boosting and bagging are two ensemble techniques that go beyond Random Forest and may improve classification accuracy.

- Deep Learning: Investigating deep learning designs, such recurrent neural networks (RNNs) and convolutional neural networks (CNNs), may reveal more complex pat- terns in malware data, improving detection capacities.

- Real-Time Detection: To keep ahead of emerging cy- berthreats, it will be essential to develop real-time detection systems that can recognise and mitigate malware threats as they arise.

- Adversarial Robustness: In order to guarantee the ef- ficacy of detection systems in practical situations, it is crucial to look into methods for improving the robustness of machine learning models against adversarial attacks, in which adversaries purposefully alter malware samples to avoid detection.

By following these research directions, we may improve cybersecurity defences and shield systems and data from the constantly changing risks posed by malicious software.

References

1. Tahtaci, B.; Canbay, B. Android Malware Detection Using Machine Learning. In Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 15–17 October 2020; pp. 1–6.
2. Baset, M. Machine Learning for Malware Detection. Master's Disserta- tion, Heriot Watt University, Edinburg, Scotland, December 2016.
3. Akhtar, M.S.; Feng, T. Deep learning-based framework for the detection of cyberattack using

feature engineering. Secur. Commun. Netw. 2021, 2021, 6129210.
4. Altaher, A. Classification of android malware applications using feature selection and classification algorithms. VAWKUM Trans. Comput. Sci. 2016, 10, 1.
5. Patil, R.; Deng, W. Malware Analysis using Machine Learning and Deep Learning techniques. In Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA, 28–29 March 2020; pp. 1–7
6. H. Li, S. Zhou, W. Yuan, J. Li and H. Leung, ”Adversarial- Example Attacks Toward Android Malware Detection System,” in IEEE Systems Journal, vol. 14, no. 1, pp. 653-656, March 2020, doi: 10.1109/JSYST.2019.2906120.
7. M. Abdel-Basset, H. Hawash, K. M. Sallam, I. Elgendi, K. Munasinghe and A. Jamalipour, ”Efficient and Lightweight Convolutional Networks for IoT Malware Detection: A Federated Learning Approach,” in IEEE Internet of Things Journal, vol. 10, no. 8, pp. 7164-7173, 15 April15, 2023, doi: 10.1109/JIOT.2022.3229005.
8. Aqib Rashid ”MalProtect: Stateful Defense Against Adversarial Query Attacks in ML-Based Malware Detection” arXiv:2302.10739v3
9. D. -O. Won, Y. -N. Jang and S. -W. Lee, ”PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Anal- ogous Zero-Day Malware Detection,” in IEEE Transactions on Emerging Topics in Computing, vol. 11, no. 1, pp. 82-94, 1 Jan.-March 2023, doi: 10.1109/TETC.2022.3170544.
10. J. Qiu et al., ”Cyber Code Intelligence for Android Malware Detection,” in IEEE Transactions on Cybernetics, vol. 53, no. 1, pp. 617-627, Jan. 2023, doi: 10.1109/TCYB.2022.3164625.
11. B. Jin, J. Choi, J. B. Hong and H. Kim, ”On the Effectiveness of Perturbations in Generating Evasive Malware Variants,” in IEEE Access, vol. 11, pp. 31062-31074, 2023, doi: 10.1109/ACCESS.2023.3262265.
12. O¨ . Aslan and A. A. Yilmaz, ”A New Malware Classification Framework Based on Deep Learning Algorithms,” in IEEE Access, vol. 9, pp. 87936-87951, 2021, doi: 10.1109/ACCESS.2021.3089586.
13. H. Bai, N. Xie, X. Di and Q. Ye, ”FAMD: A Fast Multifeature Android Malware Detection Framework, Design, and Implementation,” in IEEE Access, vol. 8, pp. 194729-194740, 2020, doi: 10.1109/AC- CESS.2020.3033026.
14. D. Li, Q. Li, Y. Ye and S. Xu, ”A Framework for Enhancing Deep Neural Networks Against Adversarial Malware,” in IEEE Transactions on Network Science and Engineering, vol. 8, no. 1, pp. 736-750, 1 Jan.- March 2021, doi: 10.1109/TNSE.2021.3051354.
15. Abdulbasit Darem, Jemal Abawajy, Aaisha Makkar, Asma Alhashmi, Sultan Alanazi, Visualization and deep-learning-based malware variant detection using OpCode-level features,Future Generation Computer Sys- tems,Volume 125,2021,Pages 314-323,ISSN 0167-739X
16. Julian Busch, Anton Kocheturov, Volker Tresp, Thomas Seidl ”Network Flow Graph Neural Networks for Malware Detection and Classification”
17. Akhtar M.S., Feng T. Deep Learning-Based Framework for the Detection of Cyberattack Using Feature Engineering. Secur. Commun. Netw. 2021;2021:6129210. doi: 10.1155/2021/6129210.
18. Baghirov E. Techniques of Malware Detection: Research Review; Pro- ceedings of the 2021 IEEE 15th International Conference on Application of Information and Communication Technologies (AICT); Baku, Azer- baijan. 13–15 October 2021; pp. 1–6
19. Akhtar M., Feng T. Comparison of Classification Model for the De- tection of Cyber-attack using Ensemble Learning Models. [(accessed on 15 October 2022)];EAI Endorsed. Scal. Inf. Syst. 2022 9:e6. doi: 10.4108/eai.1-2-2022.173293.
20. Saad S., Briguglio W., Elmiligi H. The Curious Case of Machine Learning in Malware Detection. arXiv. 20191905.07573
21. Muppalaneni N., Patgiri R. Malware Detection Using Machine Learning Approach;

Proceedings of the International Conference on Big Data, Machine Learning and Applications; Vancouver, BC, Canada. 29–30 May 2021; Singapore: Springer; 2021
22. Singhal P., Raul N. Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks. Int. J. Netw. Secur. Its Appl. 2012;4:61–67. doi: 10.5121/ijnsa.2012.4106.
23. Michael Sikorski and Andrew Honig. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. No Starch Press,USA, 1st edition, 2012.
24. Dada E.G., Bassi J.S., Hurcha Y.J. Performance Evaluation of Machine Learning Algorithms for Detection and Prevention of Malware Attacks. IOSR J. Comput. Eng. 2019;21:18–27.
25. Huang T., Zhao R., Bi L., Zhang D., Lu C. Neural Em- bedding Singular Value Decomposition for Collaborative Filtering. IEEE Trans. Neural Netw. Learn. Syst. 2022;33:6021–6029. doi: 10.1109/TNNLS.2021.3070853.
26. Y. Zhang, Z. Liu and Y. Jiang, ”The Classification and Detection of Malware Using Soft Relevance Evaluation,” in IEEE Transac- tions on Reliability, vol. 71, no. 1, pp. 309-320, March 2022, doi: 10.1109/TR.2020.3020954.
27. Q. Han, V. S. Subrahmanian and Y. Xiong, ”Android Malware Detection via (Somewhat) Robust Irreversible Feature Transformations,” in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3511- 3525, 2020, doi: 10.1109/TIFS.2020.2975932.
28. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., et al. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 2011;12:2825–2830.
29. Introduction to Simple Imputer Class. [(accessed on 15 October 2022)].