Journal of Frontiers in Multidisciplinary Research

A Comparative Analysis of Machine Learning Techniques for Malware Detection in Downloaded Files

Ghaith Mousa Hamzah Amlak
General Directorate of Education Babylon, Ministry of Education, Terbia Street, Hila, Iraq

* Corresponding Author: **Ghaith Mousa Hamzah Amlak**

Article Info

E-ISSN: 3050-9726
P-ISSN: 3050-9718
Impact Factor (RSIF) = 7.34
Volume: 06
Issue: 02
July – December 2025
Received: 05-09-2025
Accepted: 18-10-2025
Published: 05-11-2025
Page No: 457-463

Abstract
The widespread availability of malware presents a significant and growing threat to contemporary societies. The internet hosts numerous types of viruses and other dangerous programs. Research has indicated a substantial increase in the prevalence of malware over the past decade, resulting in substantial financial losses for numerous organizations. Malware is a type of software that can cause significant harm to a user's computer system. There are various ways in which a user system can be affected. To address this issue, a range of machine learning methodologies are employed in the suggested solution to determine the presence or absence of malware within a downloaded file obtained from the internet. The goal is to effectively distinguish between dangerous and harmless files using multiple machine-learning methods. The primary objective is to analyze various attributes of the downloaded file, such as the MD5 hash, size of the Optional Header, and the Load Configuration Size, to determine if the file is malicious or non-malicious. The classification of files was based on the results of this analysis. The models underwent training using various features, which enabled them to acquire the knowledge necessary for file classification. After appropriate training, the models were evaluated and compared using multiple criteria on the validation and testing datasets. This comparison was used to determine the model with the highest level of accuracy. This approach enables the identification of various forms of malware that can cause significant harm to the user's computer system. The methodology used in this study allows the identification of several types of malwares, including Adware, Backdoors, Trojan, Unknown, Rbot, Multidrop, Spam, and Ransomware. Machine learning techniques were used to detect malicious programs. The dataset contained 100000 samples and 35 characteristics from PE headers and sections. Correlation analysis was used to select important features, reducing them to only the necessary ones. We compared and optimized XGBoost and LightGBM using an 80/20 split for training and testing. Data visualization tools were used to analyze feature linkages and distributions. Benchmarking showed XGBoost and LightGBM was the most accurate (99%) for malware detection. This data-science-based procedure enhances malware detection. It shows how machine learning can be used to detect malicious code in executables.

Keywords: Malware Detection, Executable Files, Machine Learning, Xgboost, Lightgbm, Static Analysis, Malware Classification, Model Evaluation

1. Introduction

Over the course of the last ten years, there has been a substantial 87% upsurge in deleterious malware and potentially undesirable programs [1], contributions to which have largely stemmed from downloaded files from the vast expanse of the Internet [2]. A multiplicity of websites specializes in propagating malicious computational loads by expressly offering them for acquisition or secretly bundling desired installations with clandestine new programs.

Malware is computational software that harbors malicious intent and is expressly designed to wreak havoc on a victim's computer [1]. These nefarious scripts can simultaneously manipulate sensitive data, encrypt critical files, covertly monitor activity without consent, and inflict manifold forms of harm to a system. The hazards of compromise have proliferated exponentially as personal computational devices amass copious amounts of privileged information, with contaminations transpiring within seconds of visiting a corrupted web page [3, 4]. Individuals and organizations alike are susceptible to these risks.

Hence, malware detection prior to execution has become an indispensable necessity in current technologically pervasive environments.

Illicit financial fraud carried out using malware has similarly escalated, compelling certain organizations to resort to automated fraud analytics [5, 6]. However, as they emerge, defensive systems frequently lag novel threats. Victims fervently scour the web for software and manipulative hackers adeptly leverage search engine optimization strategies to elevate malware-containing sites to the highest echelons. In addition, they invest heavily in advertisements to dupe unwitting victims. By maintaining a harmless façade, they successfully convince users to download infected files [7]. Ransomware, which encrypts sensitive data until ransom funds are furnished, epitomizes one prevalent attack methodology [8].

Signature-based malware detection techniques are becoming increasingly inadequate as state-of-the-art malware strains mutate rapidly to circumvent antivirus scanners [1]. Machine learning has thus gained significant traction, with experts convincing that ML-driven anti-malware software can identify fledgling threats overlooked by conventional methods. Previous research has validated the immense promise of ML for malware detection by extracting informative features from files [9, 10].

This study realizes multiple ML models to categorize downloaded Portable Executable (PE) files as malware or legitimate programs. The substantive dataset encompasses over 100000 Windows executables with labeled statuses and 35 metadata attributes, including checksums, imports, and section particulars. After preprocessing, 35 features subsist across 100,000 samples partitioned by approximately 50% and 50% between malware and legitimate software, respectively.

These insights pave the way for developing robust antivirus solutions capable of preempting malware infection using analogous feature engineering concepts. As downloads from potentially untrustworthy sources grow unabated, such automated systems will inevitably increase their indispensable utility.

2. Related Work

Detecting malware is critical for safeguarding computer systems and protecting them against cyber threats. As malware becomes more complex and diverse, machine learning techniques have become invaluable tools for identifying and categorizing malicious software in downloaded files [11]. Machine learning models offer advantages such as analyzing large amounts of data, detecting patterns, and adapting to emerging malware. Such algorithms can differentiate malicious anomalies from benign behaviors, thereby improving the accuracy of intrusion detection systems [12]. In addition, machine learning has proven effective in catching previously unseen malware and is applicable across domains, including IoT devices [11]. Contemporary research highlights the increasing use of deep learning for malware detection, particularly for identifying sophisticated fileless threats. Machine learning is also being combined with other techniques to leverage its adaptability, scalability, and big data processing capabilities. The literature also emphasizes feature extraction techniques, such as static, dynamic, and hybrid analysis, to comprehend malware behavior and characteristics, enabling robust detection [13]. However, overcoming the evolving challenges and enhancing malware detection systems require ongoing research [14]. Robustness and resilience have become critical in the face of increasing threats. This literature review presents a comprehensive analysis of machine and deep-learning techniques. The primary objective of Acharya *et al.* [15] is to identify and respond to malicious URLs and programs. They utilized various signature matching and machine learning algorithms to recognize threats. A Random Forest model was developed using features from executable file headers to categorize files as malicious or benign. For the URLs, a Logistic Regression model was trained on suitable datasets using a tokenizer for feature extraction. The trained model predicts whether the URL is safe to visit. These modules are integrated into an application to protect systems from malware. Similarly, Udayakumar *et al.* [16] sought to understand several types of malwares and how machine learning can detect and classify malware. To capture malware data, several settings, such as the debug size and export size, were used. Three methods were used for classification: decision trees, multilayer perceptron, and Multi SVM. Multiple binary SVMs allow accurate multiclass categorization. The neural network model performed the best, with 98% and 99% accuracy on the training and test data, respectively. The most important feature discovered was the debug size. Adding to that, Khalid *et al.* [17] conducted a study on fileless malware detection using machine learning. They proposed a classification method for fileless malware based on the attack techniques and characteristics. This study demonstrated the effectiveness of machine learning techniques, such as random forest, in detecting fileless malware. Chowdhury *et al.* [18] reviewed the current state of Android malware detection using machine learning. They discussed various machine-learning approaches for Android malware detection and compared their performance. Maleki *et al.* [19] focused on packed-malware detection using a PE header and section-table information. They highlighted the limitations of traditional techniques and the application of machine-learning algorithms for identifying malware behavior. Zheng and Omote *et al.* [20] addressed the robustness of malware detection models and proposed a robust malware detection model using dimensionality reduction and machine learning techniques. They also introduced a black-box adversarial attack method called IMAGE RESOURCE attack. This study emphasized the need to consider both attackers' and defenders' perspectives in the field of malware detection.For instance, MalNet is a novel malware detection method proposed by Yan *et al.* [21]. It uses CNN and LSTM to automatically learn raw data features. Opcode sequence extraction was made possible via decompilation, and LSTM picked up the harmful code patterns. The 40,000+ sample collection contained 20,650 benign and 21,736 malicious files. The two-stage model had an accuracy rate of 99.88%, a true positive rate of 99.14%, and a false positive rate of 0.1%. Moreover, Qing Wu *et al.* [22] concentrated on detecting Android malware. Android security concerns have increased owing to the rapid expansion of users and applications [23]. Machine-learning algorithms based on static app properties have been presented for identification; however, despite their high accuracy, they are inefficient.

3. Methodology

This methodology uses machine learning techniques to detect malware based on the characteristics of PE executable files (PPEs). The dataset contains over 100,000 samples with 35

features derived from PE headers and sections. After data cleaning, features were selected using correlation analysis to identify the most relevant attributes. Models such as XGBoost and LightGBM were evaluated and optimized using an 80:20 training-testing split. Data visualization techniques were used to explore feature distributions and relationships. A comprehensive evaluation identified XGBoost and LightGBM as the top performers, achieving over 99% accuracy in distinguishing between malware and clean files. This workflow reflects a holistic data science approach to improving malware detection. It provides a model for applying machine learning to the security challenges involved in detecting malware in binary files.

3.1. Dataset Collection

To accomplish this task, a malware dataset, a diverse collection of data available online at https://www.kaggle.com/datasets/blackarcher/malware-dataset, was used. This dataset contains both malicious and benign data. These datasets were combined to create a comprehensive final dataset, encompassing only the most important characteristics of a file's nature. This dataset, comprising 100,000 datasets, provides a rich set of attributes that help distinguish between malicious and benign programs.

Each sample contained values for 35 features related to PE file properties, including headers, sections, and imports. This rich characterization of PE files can help to differentiate malware from clean programs. However, this dataset presents some challenges that need to be addressed. The target variable 'legitimate' indicates malware as 0 and benign software as 1 but has a class imbalance with only 45% of malware examples. This imbalance must be handled carefully during modeling to prevent prediction bias. Additionally, certain features like 'Machine' contain invalid entries not mapping to any real machine code. Such anomalies in the data require pre-processing to avoid technical errors when training the models. Nevertheless, the overall large sample size and meaningful PE file attributes make this a viable real-world dataset for developing machine-learning classifiers to detect malware. Although the data holds potential, data cleaning and preprocessing steps are required before further usage to build robust and unbiased malware detection models. Figure 1 highlights the bar plots for class imbalance.

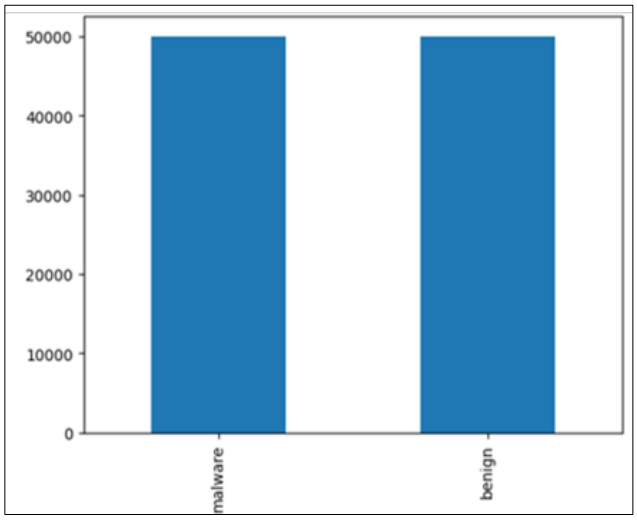

Fig 1: Bar Plot Showing Class Imbalance

3.2. Pre-processing Data:

Several data preprocessing steps were performed to prepare the dataset for effective modeling. The 'ID, ' 'md5,' and unnamed columns were dropped as they did not provide any useful predictive information. Missing values were removed from the samples after thorough checking to avoid technical errors during the model building. The 'Machine' feature was converted to an integer data type since it represented numerical machine codes. This process is achieved via one hot encoding technique. One Hot Encoding is an encoding technique that is used to convert categorical parameters into numeric values (often 0s and 1s). This enables categorical information about the dataset to be presented in a way that is compatible with the various Machine Learning Algorithms. Moreover, the original string version caused mismatch issues and required fixing. Duplicate samples were discarded to prevent bias in the models, as duplicates can skew the results. Finally, feature normalization was performed using the Standard Scalar method to rescale all variables to a common scale. This step assists certain models to converge better during training. In short, critical data preprocessing, such as dropping non-predictive columns, handling missing data, converting data types, removing duplicates, and feature scaling, readied the dataset. For clean modeling and prevented multiple issues that could negatively impact model performance.

These steps are explained as follows:

3.2.1. Exploratory data analysis and Data Visualization (EDA): To develop an effective malware detection model, comprehensive exploratory data analysis and visualization were performed, including computing summary statistics to reveal overall data characteristics and outliers, evaluating feature uniqueness to identify non-predictive attributes, generating histograms to visually profile feature distributions, highlighting aspects such as skew and multimodality, performing correlation analysis and heat maps to identify redundant attributes with high collinearity for removal, and using pair plots to visualize feature interactions and clustering correlated variables. These analyses have helped in informed data pre-processing and feature engineering by addressing issues such as class imbalance, non-normal distributions, low-signal and redundant features, and retaining promising predictive attributes. The new dataset provides an interesting mix of common software files and Virusshare malware samples. The file names were extracted into a categorical feature denoting the program type (. exe,.dll, unknown) can provide predictive signals. Summary statistics reveal the feature distribution properties and flagging pre-processing requirements. Visualizations can surface file type clusters, whereas correlation analysis identifies attributes requiring regularization. Studying distributions can highlight non-normal variables that require transformation. The resulting insights guided the optimized model building by informing nuanced data preparation decisions. Following the modeling, additional visualizations allowed intuitive performance comparisons. Model accuracy and error measures were compared using bar graphs. In general, systematic data visualization aids quantitative analytics. The iterative analysis, visualization, and model refinement loop improved malware detection. The foundation for the robust models was laid through visually guided data cleaning and feature selection. The model evaluation was aided by post-modelling

pictures. This improved the malware classification techniques and outcomes. Figure 2 shows the correlation between predictor attributes whereas figure 3 shows pair plots [24].

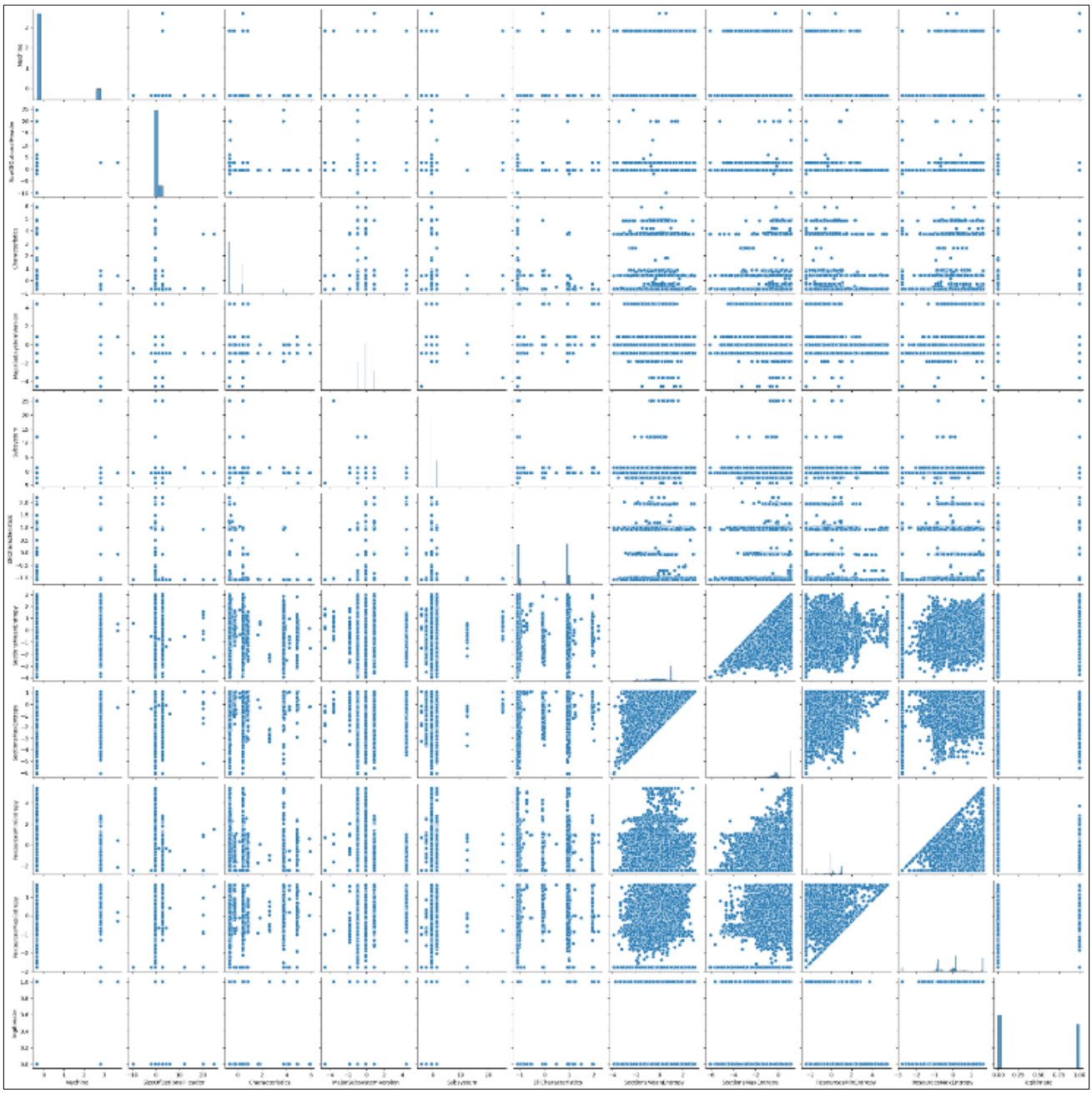

Fig 2: Pait Plot for the Predictor Variables

3.2.2. Performance Evaluation Metrics: The Accuracy measures the overall prediction correctness but can be insufficient with class imbalance. F1-score supplements accuracy by balancing precision and recall. This accounts for the false positives and negatives. Together, accuracy and F1-score enable comprehensive model evaluation, particularly for uneven class distribution where false positives and negatives require equal weighting. Accuracy has been given in eq (1).

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (1)$$

Accuracy computes the ratio of correct predictions to total predictions, signifying the fidelity of the model. The numerator sums true positives and true negatives, that is, correct positive and negative labels. The denominator totals all the classified data points. This ratio reveals the proportion of precise classifications out of the total predictions. F1 score and precision have been produced in eq (2) and eq(3) respectively.

$$F1\text{-}score=2*(precision*recall)/(precision+recall) \quad (2)$$

With:

$$Precision = TP / (TP + FP) \quad (3)$$

Precision measures the model's ability to accurately identify true positives out of all positive predictions. Higher precision means fewer false positives.

$$recall = TP / (TP + FN) \quad (4)$$

Recall measures a model's ability to correctly detect true positives. It is the ratio of true positives to the sum of true positives and false negatives. Higher recall means more true positives are correctly identified, with fewer detrimental false negatives. Maximizing recall is critical in sensitive domains like malware detection where overlooking true positives has severe repercussions.

3.2.3. Testing and Training subsets: In line with established research methodology, a partitioning approach is employed to allocate 80% of the dataset as the training set, while reserving the remaining 20% as the test set. This systematic division allows for rigorous evaluation and assessment of the model's accuracy and performance.

3.3. Implementation

3.3.1. Experimental Setup

Our malware detection system is built on a strong architecture and an optimized technology stack. It requires 8-16 GB of RAM, an Intel Core i5 CPU, and 128 GB of storage. It runs on Linux, Windows, or MacOS and has high-speed Internet connectivity. Our tech stack (Python, Scikit-Learn, Pandas, and Matplotlib) enables coding, visualization, and analysis. Jupyter Notebook/Google Colab provides an interactive dev environment. Our setup is adept at malware detection with a firm understanding of ML and data science.

3.3.2. Classification Algorithm

Classification can be done with several machine learning methods. The models can be categorized into two main types: supervised and unsupervised. Classification can be accomplished with a variety of machine learning methods. Supervised and unsupervised models are the two main categories here. XGBoost AND LightGBM are examples of supervised machine learning models.

XGBOOST Algorithm

The xgboost model achieved an exceptional accuracy of 99% for the malware classification task. The low RMSE of 0.1 further demonstrated the strong predictive performance of the model. Examining the confusion matrix and classification report, the results were balanced across both benign and malware classes. For the benign class, the precision was 96%, meaning that positive predictions were almost always correct, whereas the recall was 96%, indicating that the model correctly identified nearly all benign instances. The F1 score is excellent (0.95). Similarly, for the malware class, precision was 96% and recall was 94%, with an F1 score of 0.95. The model exhibited high accuracy in correctly predicting both positive and negative cases. With both classes having high precision and recall, the macro-average F1 was 0.98, indicating no bias. The weighted F1 was equally high at 0.98, underscoring the effectiveness of benign and malware examples. Figure shows accuracy, RMSE and confusion matrix for Random forests algorithm.

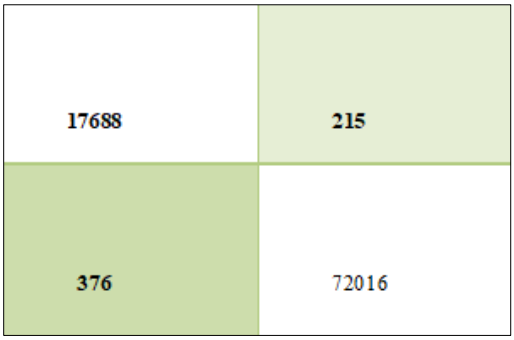

Fig 3A: Confusion Matrix for XGBOOST

Fig 3B: Confusion Matrixe for XGBOOST

LightGBM Algorithm

The lightGBM model demonstrated a notable level of accuracy, reaching 99% when used for the classification of malware test data. Additionally, the model exhibited a low root mean square error (RMSE) value of 0.1, which suggests the presence of a strong and reliable predictive modelling approach. Upon analysis of the classification metrics, it was determined that the performance exhibited a state of equilibrium for both the benign and malware categories. In the benign class, the precision was seen to be 96%, indicating a high level of accuracy in positive predictions. However, the recall rate was somewhat lower at 94%, suggesting that the model failed to identify some occurrences of benign cases. The F1 score, which is a metric that combines precision and recall, achieved a value of 0.93. In the context of malware examples, the precision metric yielded a value of 96%, indicating the proportion of correctly identified malware instances out of all instances classified as malware. Similarly, the recall metric achieved a value of 93%, representing the proportion of correctly identified malware instances out of all actual malware instances. These metrics collectively resulted in an F1 score of 0.94, which is a measure of the model's overall accuracy in identifying malware instances.

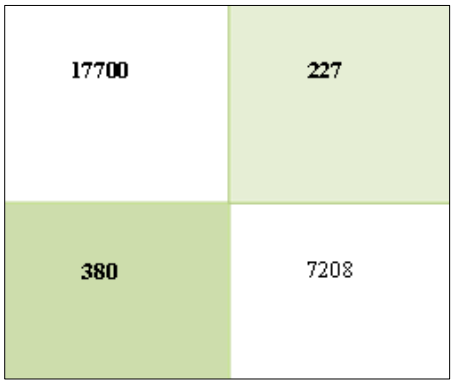

Fig 4A: Confusion Matrix for Lightgbm

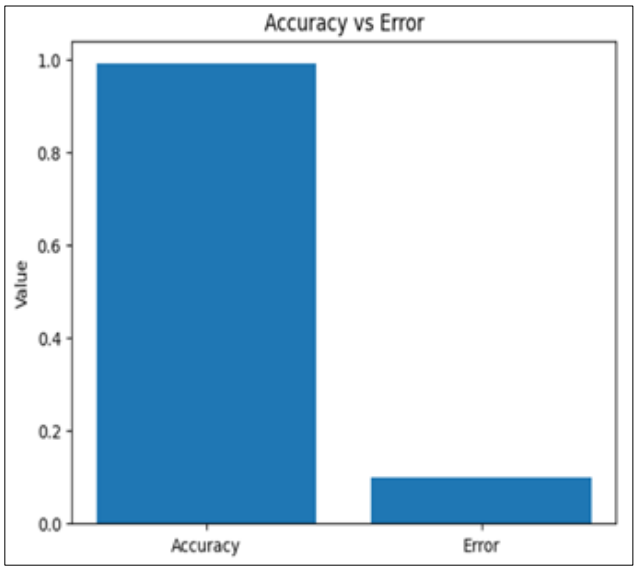

Fig 4B: Accuracy and RMSE for LIGHTGBM

Conclusion

In summary, our study employed machine learning methodologies to enhance the efficiency of malware identification by leveraging the characteristics of Portable Executable (PE) files. The dataset utilized in this study consists of a substantial number of samples, exceeding 100,000, and has 35 attributes that pertain to PE headers, sections, imports, and other relevant factors. Following the data cleaning process, the performance of various models, including XGBOOST and LIGHTGBM algorithms, was assessed using an 80/20 train-test split. These algorithms demonstrated superior performance, with a remarkable accuracy rate of 99% in effectively discerning between malicious software and harmless data. The use of one-hot encoding facilitated the transformation of the data into inputs that are compatible with the model. This exemplified the significant improvements that may be achieved through feature engineering. The use of data visualization and statistical analysis facilitated significant data changes. Through a process of rigorous benchmarking and optimization, the most successful strategies were determined. Both shallow and deep learning algorithms have demonstrated proficiency in detecting harmful patterns. The study presents a framework for utilizing machine learning in security applications, demonstrating the effectiveness of informed feature engineering and modelling in enhancing predictive accuracy.

References

1. Kamboj A, Kumar P, Bairwa AK, Joshi S. Detection of malware in downloaded files using various machine learning models. Egyptian Inform J. 2023;24(1):81-94. doi:10.1016/j.eij.2022.12.002
2. Han W, Xue J, Wang Y, Huang L, Kong Z, Mao L. MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics. Comput Secur. 2019;83:208-33. doi:10.1016/j.cose.2019.02.007
3. Burnap P, French R, Turner F, Jones K. Malware classification using self organising feature maps and machine activity data. Comput Secur. 2018;73:399-410. doi:10.1016/j.cose.2017.11.016
4. Yousif H, Al-saedi KH, Al-Hassani MD. Mobile phishing websites detection and prevention using data mining techniques. iJIM. 2019;13(10):205-17. doi:10.3991/ijim.v13i10.10567
5. Sharma A, Sahay SK. Evolution and detection of polymorphic and metamorphic malwares: a survey. arXiv. 2014:arXiv:1406.7061. Available from: https://arxiv.org/abs/1406.7061
6. Al-Saedi KH, Saeed Abdulrazzaq N, Selman DI. Feature extraction mining method for intrusion detection systems based on DARPA 1999 dataset. (No journal/publication details provided – please complete citation if published)
7. Singh J, Singh J. A survey on machine learning-based malware detection in executable files. J Syst Archit. 2021;112:101861. doi:10.1016/j.sysarc.2020.101861
8. Akhtar Z. Malware detection and analysis: challenges and research opportunities. arXiv. 2021:arXiv:2101.08429. Available from: https://arxiv.org/abs/2101.08429
9. Sharma A, Sahay SK. An effective approach for classification of advanced malware with high accuracy. arXiv. 2016:arXiv:1606.06897. Available from: https://arxiv.org/abs/1606.06897
10. Tchakounté F, Hayata F. Supervised learning based detection of malware on Android. In: Mobile Security and Privacy. Elsevier; 2017. p. 101-54.

11. Ali MM, Maqsood F, Hou W, Wang Z, Hameed K, Zia Q. Machine learning-based malware detection for IoT devices: understanding the evolving threat landscape and strategies for protection. Research Square. 2023. doi:10.21203/rs.3.rs-2754989/v1
12. Carpenter M, Chunbo L. Behavioural reports of multi-stage malware. arXiv. 2023:arXiv:2301.12800. Available from: https://arxiv.org/abs/2301.12800
13. Xuan CD, Thi Nga NT, Dinh NT, *et al*. Building a malware detection system based on a machine learning method. Int J Innov Technol Explor Eng. 2020;9(5):123-9. doi:10.35940/ijitee.E2945.039520
14. Souri A, Hosseini R. A state-of-the-art survey of malware detection approaches using data mining techniques. Hum Cent Comput Inf Sci. 2018;8:3. doi:10.1186/s13673-018-0125-x
15. Acharya J, Chaudhary A, Chhabria A, Jangale S. Detecting malware, malicious URLs and virus using machine learning and signature matching. In: 2021 2nd International Conference for Emerging Technology (INCET); 2021 May 21-23; Belgaum, India. IEEE; 2021. p. 1-5. doi:10.1109/INCET51464.2021.9456226
16. Udayakumar N, Saglani VJ, Cupta AV, Subbulakshmi T. Malware classification using machine learning algorithms. In: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI); 2018 May 11-12; Tirunelveli, India. IEEE; 2018. p. 1-9.
17. Khalid O, *et al*. An insight into the machine-learning-based fileless malware detection. Sensors (Basel). 2023;23(2):612. doi:10.3390/s23020612
18. Chowdhury MNUR, Haque A, Soliman H, Hossen MS, Ahmed I, Fatima T. Android malware detection using machine learning: a review. TechRxiv. 2023. doi:10.36227/techrxiv.22580881
19. Maleki N, Bateni M, Rastegari H. An improved method for packed malware detection using PE header and section table information. Int J Comput Netw Inf Secur. 2019;11(9):19-28. doi:10.5815/ijcnis.2019.09.02
20. Zheng W, Omote K. A study on robustness of malware detection model. Ann Télécommun. 2021;76:789-99. doi:10.1007/s12243-021-00899-z
21. Yan J, Qi Y, Rao Q. Detecting malware with an ensemble method based on deep neural network. Secur Commun Netw. 2018;2018:7247095. doi:10.1155/2018/7247095
22. Wu Q, Zhu X, Liu B. A survey of Android malware static detection technology based on machine learning. Mobile Inf Syst. 2021;2021:8864665. doi:10.1155/2021/8864665
23. Joshi R, Bairwa AK, Soni V, Joshi S. Data security using multiple image steganography and hybrid data encryption techniques. In: 2022 International Conference for Advancement in Technology (ICONAT); 2022 Jan 21-22; Goa, India. IEEE; 2022. p. 1-7. doi:10.1109/ICONAT53423.2022.9725892
24. Komorowski M, Marshall DC, Salciccioli JD, Crutain Y. Exploratory data analysis. In: Secondary Analysis of Electronic Health Records. Cham: Springer; 2016. p. 185-203. doi:10.1007/978-3-319-43742-2_15