JOURNAL OF ADVANCED INDUSTRIAL TECHNOLOGY AND APPLICATION
e-ISSN: 2716-7097

JAITA

Vol. 6 No. 2 (2025) 37-47
https://publisher.uthm.edu.my/ojs/index.php/jaita

Performance Metrics of Different Machine Learning Models for Windows Malware Detection

Fadhil Mukhlif[1]*, Ibrahim Hashem[2], Norafida Ithnin[3]

[1] *Department of Cybersecurity Engineering Techniques, Technical Engineering College for Computer and AI, Northern Technical University, Kirkuk, IRAQ*
[2] *Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, UNITED ARAB EMIRATES*
[3] *Information Assurance and Security Research Group (IASRG), Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, MALAYSIA*

*Corresponding Author: fmukhlif@ntu.edu.iq
DOI: https://doi.org/10.30880/jaita.2025.06.02.004

Article Info

Received: 23 September 2025
Accepted: 3 November 2025
Available online: 31 December 2025

Keywords

Cybersecurity, malware detection, machine learning, AI models, performance metrics

Abstract

This study experimentally evaluates and analyzes the performance of various machine learning models for Windows malware detection. Their metrics are further analyzed to identify the most effective approach. For this purpose, the researchers employed a diverse dataset to train and assess the models. The used dataset contains known Windows malware samples and benign files. Besdies, the chosen machine learning algorithms, such as Logistic Regression (LR), AdaBoost, LightGBM (LGBM), XGBoost (XGB), Decision Trees (DT), Gradient Boosting, Bagging, Random Forest (RF), and Support Vector Machines (SVM), have various techniques. The study focuses on key performance metrics: Accuracy, Precision, Recall, F1 Score, Specificity, False Positive Rate (FPR), Negative Predictive Value (NPV), False Negative Rate (FNR), and Error Rate. They are used to thoroughly assess the models' effectiveness in distinguishing between malware and benign samples. Additionally, the exploration of the impact of feature selection and extraction methods on model performance is carried out to gain better insights. The study results demonstrate variations in the models' effectiveness. It is noted that certain algorithms like; random Forest, Bagging XGB, and LGBM are demonstrate superior performance in specific metrics. They also offer significant perspectives into the strengths and weaknesses of various machine learning models in the detection of Windows malware, contributing valuable knowledge to the development of more robust cybersecurity strategies. The study implications can hopefully be used to develop an effective and accurate malware detection model. It is expected the model may ultimately foster the security of Windows environments.

1. Introduction

With the evolving and shifting fields of cybersecurity, Windows malware detection and mitigation represent the cutting edge of protecting digital environments [1]. As malicious software becomes more sophisticated and diverse, there is a dire need for effective machine learning models to identify and neutralize these threats [2]. Fig. 1 illustrates a scenario where hackers perform various malware attacks on systems. The increase of machine learning algorithms has emerged recently in fostering the malware detection systems in terms of efficiency and

accuracy. Threats posed by Internet-based malware are discussed in [3], along with the shortcomings of human heuristic analysis in slowing down its spread. The author's proposed solution involves automated behavior-based malware detection utilizing machine learning. Malware behavior is scrutinized in a simulated environment, generating reports subsequently processed into sparse vector models for classification. Moreover, [4] build a malware detection system to be accessible online which on process-level performance metrics.

The study has tested the effectiveness of various techniques such as KNN, SVC, RFC, GNB and CNN. Utilizing a dataset comprising both malicious and benign samples, the research concludes that neural network models, specifically Convolutional Neural Networks, exhibit the highest accuracy in identifying the influence of malware on process-level features within virtual machines deployed in the cloud. The generation of the dataset was the result of running different malware families in the aforementioned cloud and collecting process-level features.

This research investigates the comprehensive evaluation of the performance metrics of various machine learning models employed for Windows malware detection. Moreover, the study addresses the adaptability of these models to evolving malware strains, considering the changeable nature of cyber threats. By scrutinizing the models' abilities to generalize and adapt to new and previously unseen malware patterns, the aim is to give insights that contribute to the ongoing development of resilient and responsive cybersecurity frameworks. The studies conducted on the Portable Executable malware dataset generated from Windows which show an ensemble of neural networks whose number is seven, with the Extra Trees classifier acting as a classifier in the final stage, resulted in the most promising results. This underscores the effectiveness of the proposed ensemble learning technique in malware detection, especially in addressing novel and unexplored threats.

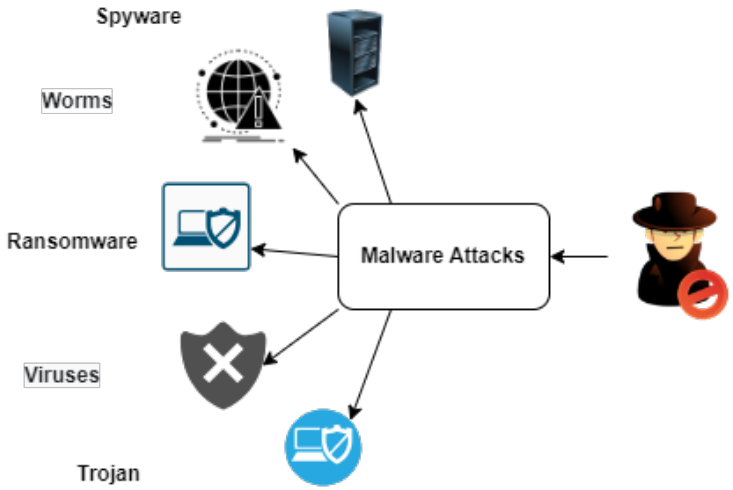

Fig. 1 *Types of malware attacks*

This study compares and evaluates the efficacy of different machine learning models in identifying Windows malware, taking into account variables like computational efficiency, false positive rates, and detection accuracy. The machine learning algorithms have been chosen to cover a variety of approaches, such as Logic Regression, XGB, LGBM, Decision Tree (DT), Random Forest (RF), Ada Boost, Gradient Boosting, Bagging and SVM. The study concentrates on important performance measures like recall, precision, accuracy, and F1 score in order to thoroughly evaluate the models' capacity to distinguish between malware and benign cases. The diversity of models under scrutiny encompasses traditional classifiers, ensemble methods, and deep learning architecture.

However, the contributions of this study are listed as follows:

- The primary emphasis of this study is on developing a methodology for detecting malware specifically tailored to IoT devices within the context of emerging technologies.
- We introduced a method to empirically evaluate and select the best classifiers for detecting malware.
- An inclusive assessment of the performance measures of the top nine machine learning classifiers was conducted to determine the best classifier's effectiveness in detecting Windows malware.

The study outlined the following: section 2 provides the problem formulation; section 3 presents research framework; section 4 provides the methodology of the study including datasets feature extraction, feature selection, model selection, performance evaluation, and comparative analysis; section 5 offers results and discussion, and section 6 concludes the paper.

2. Problem Formulation

In the context of cybersecurity, the problem at hand revolves around the detection of Windows malware, where the current methods and systems exhibit shortcomings in accurately identifying and mitigating malicious software threats [5]. The proliferation of sophisticated malware poses a significant risk to computer systems and user data, necessitating an improved and efficient detection mechanism [6]. The primary objective of this initiative is to enhance the accuracy and reliability of Windows malware detection systems within a specified timeframe, aligning with the broader goal of safeguarding user information and system integrity. However, the challenge results from resource limitations, including constraints on computational power, dataset size, and user privacy ethical issues. Key stakeholders in this scenario encompass both end-users, whose systems are vulnerable to malware attacks, and cybersecurity professionals whose aim is to develop effective defense mechanisms. It is expected to have more increased malware infiltration and compromised security should cybersecurity fail to do its tasks. To eliminate such threats, a successful solution will bolster the resilience of Windows systems against evolving cyber threats. This scope designed for Windows systems encompasses refining current detection algorithms and exploring innovative machine learning models, excluding considerations beyond the realm of software-based threats. To achieve this, we assume the reliability of existing malware datasets and the feasibility of implementing proposed solutions in line with given constraints. Metrics such as precision, recall, and overall accuracy, ensuring a robust and effective solution, function as yardsticks. A stacked ensemble of multilayered, dense neural networks is used for the first stage of classification, followed by a meta-learner for the final step.

The researchers put forward a proposal for a multi-phase ensemble classification approach for detecting malicious software and strengthening digital ecosystems against varied malware threats. Organizational commitments to cybersecurity and user protection are taken into consideration [7]. The current study endeavors to evaluate and compare the performance of 14 classifiers for the meta-learner. It also presents a baseline comparison with 13 machine learning methods. Experiments on the ClaMP dataset show that an ensemble of five convolutional and dense neural networks, along with the Extra Trees classifier as the meta-learner, performs the best. Present in [7] an ML-based malware detection system focusing on the analysis of Portable Executable file headers. Diverse machine learning models, such as Random Forest, Decision Tree, Support Vector Machine (SVM), AdaBoost, Gradient Boosting, and Gaussian Naive Bayes (GNB), are employed in the preparation of data. A comparative analysis is subsequently conducted to determine the most effective model. Based on the results, Random Forest has managed to score the highest accuracy at 99.44%. The results suggest the potential for developing a customizable desktop application to scan and detect malware on the Windows platform. [9] presents a two-stage method based on ensemble learning, that is, feature extraction and classification. A meta-learner constructed from 15 ML classifiers is used in the final classification stage, whilst a stacked ensemble of fully connected and one-dimensional CNNs is used in the first classification stage. [4] presents a virtual malware detection system depending on process-level performance metrics. Varied baseline machine learning models, including “Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), Random Forest Classifier (RFC), Gradient Boosted Classifier (GBC), Convolutional Neural Networks (CNN), and Gaussian Naive Bayes (GNB), are assessed for their effectiveness. According to the study, which used a dataset of 40,680 harmful and unharmful samples, neural network models—more especially, convolutional neural networks—show the best accuracy in identifying how malware affects the process-level characteristics of cloud virtual machines. The dataset was created by running various malware families in a real cloud environment and recording features at the process level. This study contributes to ongoing efforts in developing robust malware detection mechanisms for cloud services.

3. Research Framework

Regarding cybersecurity, the framework for assessing the performance criteria of several machine learning models for Windows malware detection is unavoidable to improve the efficacy and dependability of threat detection systems as depicted in Fig. 2. This initiative aims to systematically assess all aspects of diverse machine learning approaches in detecting Windows-based malware. It attempts to establish a thorough framework that makes it easier to compare the different models’ performance. The framework is designed to accommodate the unique challenges posed by the dynamic nature of malware threats, diverse dataset characteristics, and real-time detection efficacy. Furthermore, it seeks to apply metrics like accuracy, precision, recall, specificity, NPV, FPR, FNR, error, and F1 score. The study limitations are affected by resource scarcity and ethical considerations, which are taken into consideration during the framework development to ensure its practicality in day-to-day usage and scenarios. The impact of this proposed framework lies in its capacity to inform the selection and optimization of machine learning models. It may contribute to the development of robustness and adaptivity in Windows malware detection systems. The initiative extends to provide a thorough exploration and analysis of existing models, feature engineering techniques, and the incorporation of emerging technologies in order to deal with the swiftly evolving malware field. The success of the framework is measured by its ability to provide actionable insights for practitioners and researchers, fostering advancements in the ongoing battle against Windows-based cyber

threats. This initiative is motivated by the critical need to fortify cybersecurity measures, aligning with broader objectives of safeguarding digital ecosystems and user privacy.

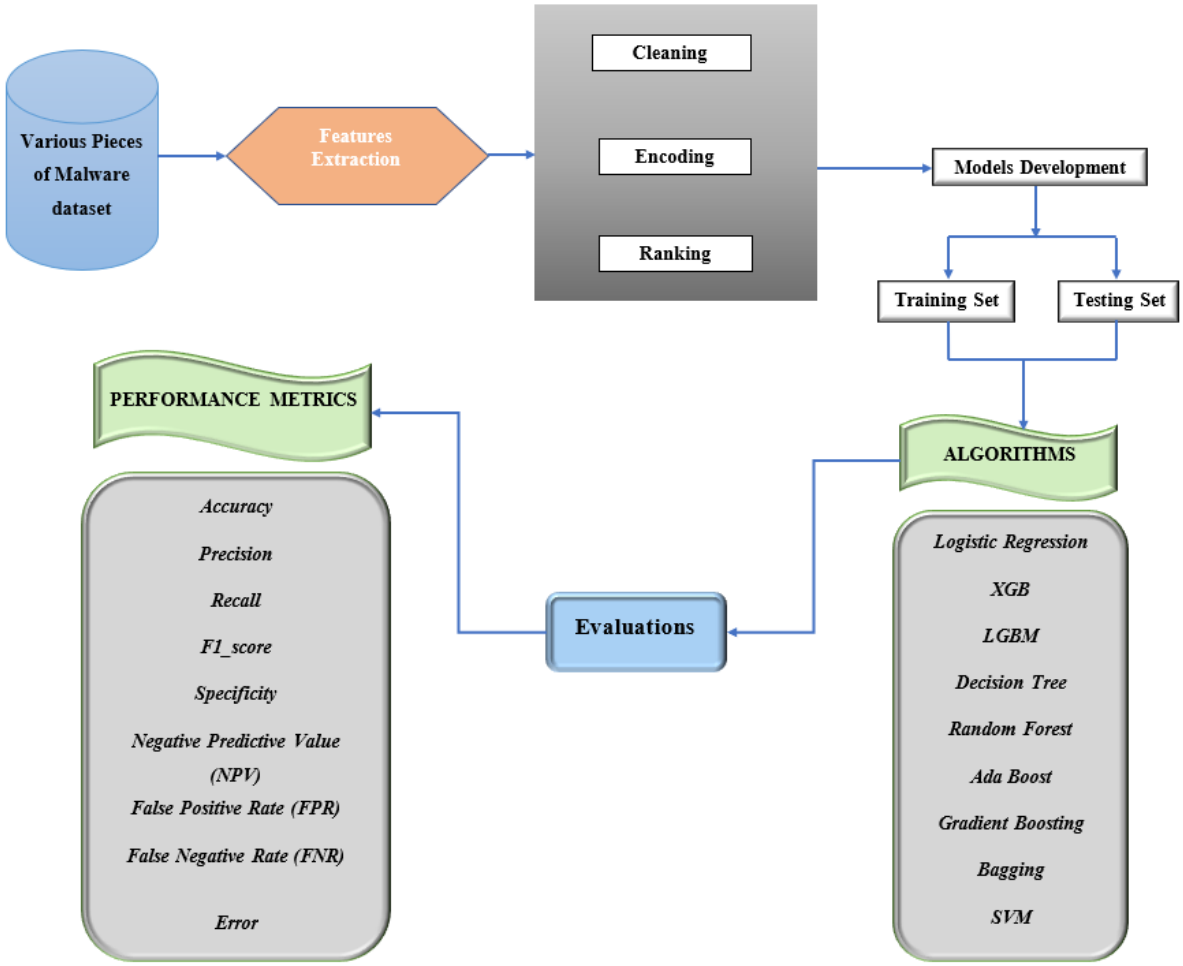

Fig. 2 *Proposed framework*

4. Methodology

A good methodology seeks to assess the performance metrics of different ML models in connection with the detection of Windows malware. The process shall be all-embracing and systematic. The collection of a diverse dataset, be malicious or benign, and the incorporation of real-world scenarios are vital steps to promote authenticity. A variety of ML models, including random forests, decision trees, support vector machines, and deep learning techniques, are selected after feature engineering and data preprocessing. For model training and evaluation, the dataset is divided into two sets: training and validation. Cross-validation methods are also used to guarantee reliability. The use of hyperparameter tuning is conducted to optimize the model performance. A wide range of performance parameters, including "accuracy, recall, F1 score, precision, specificity, negative predictive value, false positive rate, false negative rate, and error", are used to thoroughly assess the selected models. Confusion matrices offer comprehensive insights. Comparative analysis seeks to uncover patterns and assess the strengths and weaknesses of each model. Ethical considerations, including privacy and security implications, are carefully addressed throughout the process. Additionally, resource utilization is evaluated to gauge practical implementation in real-world scenarios with potential constraints. The entire methodology is thoroughly documented, providing a transparent account of data sources, preprocessing steps, model selection criteria, and evaluation metrics, ultimately contributing valuable insights for the selection and optimization of machine learning models for Windows malware detection.

4.1 Dataset

The study utilized data sourced from the Kaggle website. This employed dataset contains logs that have been compromised by different types of malwares. These logs encompass a wide range of file types, providing ample opportunities for training models with various log features. Analysis revealed the presence of five distinct malware families within the samples. The dataset comprises 29,499 columns and 534 rows, as illustrated in Fig. 3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	SHA256	Type	advapi32.(	kernel32.(	vspmsg.dl	ole32.dll	oleaut32.(	psapi.dll	setupapi.(	shlwapi.d	pdh.dll	xmllite.dl	msvcr110.	user32.dll	msvcrt.dll	she
2	002ce0d28	0	1	1	1	1	1	1	1	1	1	1	0	0	0	
3	2a053f32b	0	1	1	0	0	0	0	0	0	0	0	1	0	0	
4	2f031a175	0	1	1	0	1	0	0	0	0	0	0	0	1	1	
5	308e8bb2(	0	1	1	0	1	1	1	0	1	0	0	0	1	0	
6	31aaba443	0	1	1	0	0	0	0	0	0	0	0	1	0	0	
7	373d0778(	0	1	1	0	1	1	0	0	0	0	0	0	0	1	
8	39bbaf075	0	1	1	0	0	0	0	0	0	0	0	0	1	0	
9	3a001b8c4	0	1	1	0	1	0	0	0	1	0	0	0	1	1	
10	3a44fa455	0	1	1	0	0	0	0	0	0	0	0	0	0	0	
11	3ca96b622	0	1	1	0	1	1	0	0	0	0	0	0	1	1	
12	3e9173761	0	1	1	0	0	0	0	0	0	1	0	0	0	0	
13	3f3fe9eca	0	1	1	0	1	0	0	0	0	0	0	0	1	1	
14	42fd07c96	0	1	1	0	1	1	0	0	0	0	0	0	1	1	
15	456f313ed	0	1	1	0	1	0	0	0	0	0	0	0	1	1	
16	47b3c091f	0	1	1	0	0	0	0	0	0	0	0	0	1	0	
17	4859684e3	0	1	1	0	1	1	1	0	0	1	0	0	1	0	
18	48bb6fd27	0	1	1	0	0	0	0	0	0	0	0	0	1	0	
19	544a20393	0	1	1	0	1	1	1	0	0	0	0	0	1	0	
20	54e268557	0	1	1	0	0	0	0	0	1	0	0	0	1	1	
21	60b19233a	0	1	1	0	1	1	0	1	0	0	0	0	0	1	

Fig. 3 *Dataset preview*

4.2 Features Extraction

In the 21st century, datasets often comprise tens of thousands of features, which can lead to overfitting when used in machine learning models. To deal with this issue, we have adopted a strategy of feature selection, where we reduce the original large number of attributes to a more manageable set by selecting only the most relevant details.

4.3 Features Selection

After feature extraction comes feature selection, which involves identifying additional features. Feature selection is a process in which features are chosen from a pool of recently discovered properties. It is essential for improving accuracy, simplifying the model, and preventing overfitting. Historically, researchers have employed various feature categorization algorithms to detect malicious software. In this study, the feature ranking approach is heavily utilized due to its effectiveness in selecting relevant features for constructing malware detection models.

4.4 Model Selection

Several machine learning models are chosen for evaluation based on being suitable for binary classification tasks and having a track record in malware detection. The selected models include “DT, SVM, RF, Logic Regression (LR), XGB, LGBM, Ada Boost, Bagging, and Gradient Boosting”.

4.5 Performance Metrics

The effectiveness of the suggested approach and other modern techniques we looked at was evaluated using multiple indicators. These metrics are based on popular concepts used in binary classification issues, where outcomes are represented by True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN)”.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{4}$$

$$F1 - score = \frac{2 \times TP}{2 \times (TP + FP + FN)} \tag{5}$$

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{6}$$

$$Negative\ Predictive\ Value\ (NPV) = \frac{TN}{TN + FN} \tag{7}$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{TN + FP} \tag{8}$$

$$False\ Negative\ Rate\ (FNR) = \frac{FN}{TP + FN} \tag{9}$$

4.6 Comparative Analysis

Every machine learning model's performance is contrasted with baseline techniques. By highlighting each model's advantages and disadvantages, this comparative analysis helps choose the best strategy for Windows malware detection.

5. Results and Discussion

The research proposed model for comparing malware performance is implemented using Python 3.9 software to check its functionality and performance. Testing is carried out on a system with an "Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz (4CPUs), ~2.6GHz, 8.0 GB Memory (RAM), and a 64-bit operating system". Table 1 illustrates the comparison of performance metrics including Accuracy, Recall, Precision, F1_Score, Specificity, Negative Predictive Value (NPV), False Positive Rate (FPR), False Negative Rate (FNR), and Error across nine AI models: Logistic Regression, XGB, LightGBM (LGBM), DT, RF, AdaBoost, Gradient Boosting, Bagging, and Support Vector Machines (SVM). The testing aims to identify which of the tested models, or combination thereof, would be effective for real-world deployment in defending against malicious hackers attempting to steal data and threaten the security of the digital world. The cyber threats are one of the most trending hot topics to be researched and studied all over the world, especially after the explosion of using the smart devices by the individual users, companies, and governments all over the world. So, it was necessary to research in this emerging area to find suitable Artificial Intelligence models over testing many of them in this study to produce it to the real-world application to be efficient in terms of defending the smart world currently and in the future.

J. of Advanced Industrial Technology and Application Vol. 6 No. 2 (2025) p. 37-47

Table 1 *Performance metrics comparing the proposed and current methods*

Performance Metrics Vs Models	Logistic Regression	XGB	LGBM	DT	RF	Ada Boost	Gradient Boosting	Bagging	SVM
Accuracy	0.58	0.62	0.62	0.29	0.62	0.53	0.61	0.62	0.59
Precision	0.58	0.62	0.62	0.29	0.62	0.53	0.61	0.62	0.59
Recall	0.58	0.62	0.62	0.29	0.62	0.53	0.61	0.62	0.59
F1_score	0.58	0.62	0.62	0.29	0.62	0.53	0.61	0.62	0.59
Specificity	0.5	0.5	0.5	0.5	05	0.5	0.5	0.5	0.5
Negative Predictive Value (NPV)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
False Positive Rate (FPR)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
False Negative Rate (FNR)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Error	0.41	0.37	0.37	0.70	0.37	0.46	0.38	0.37	0.40

However, Fig. 4, Fig. 5, Fig. 6, and Fig. 7, illustrates the accuracy, Precision, Recall, F1_Score and Error performance for the tested models. Models of RF, Bagging, LGBM and XGB show the best performance compared to the other models tested in the proposed model. Furthermore, Fig. 8 shows the Error performance for the models. It appears clearly the Decision Tree (DT) model has the highest error among the others tested model. So, it's harmful to use DT model in Malware detection and cyber security in real world applications and systems. Further, this study shows "Negative Predictive Value (NPV), False Positive Rate (FPR), False Negative Rate (FNR)" has the fixed values which is 0.5 in all models testing.

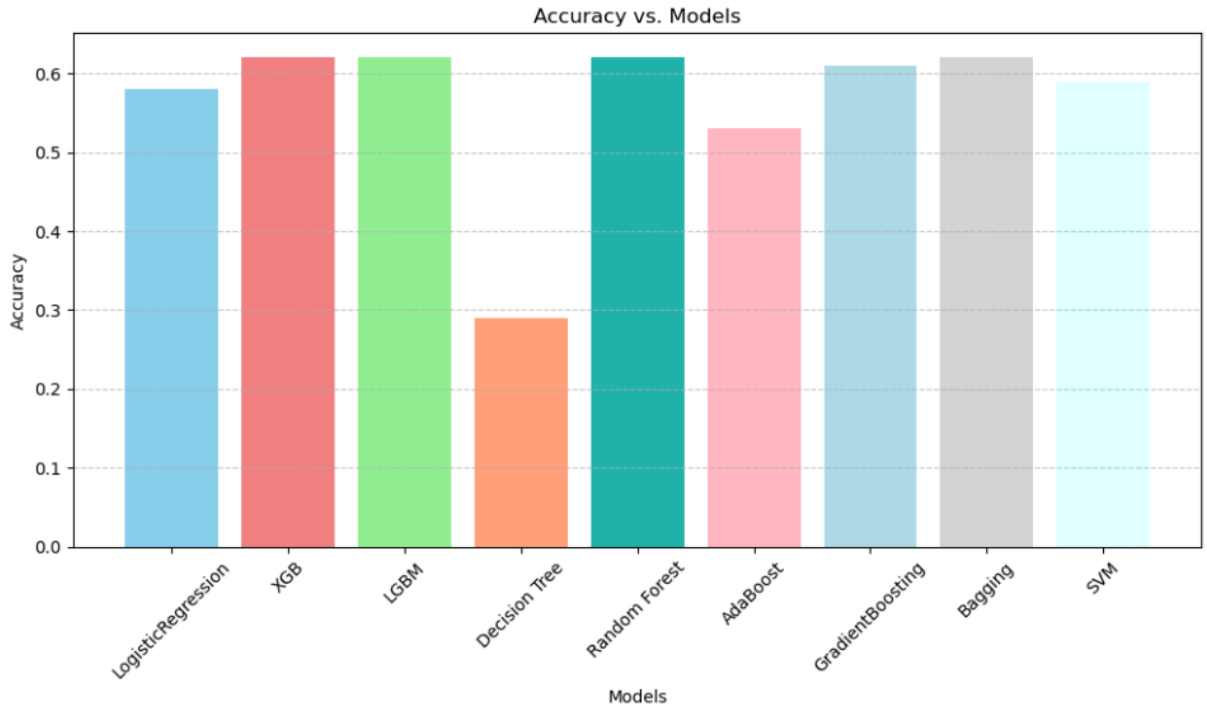

Fig. 4 *Accuracy vs. models*

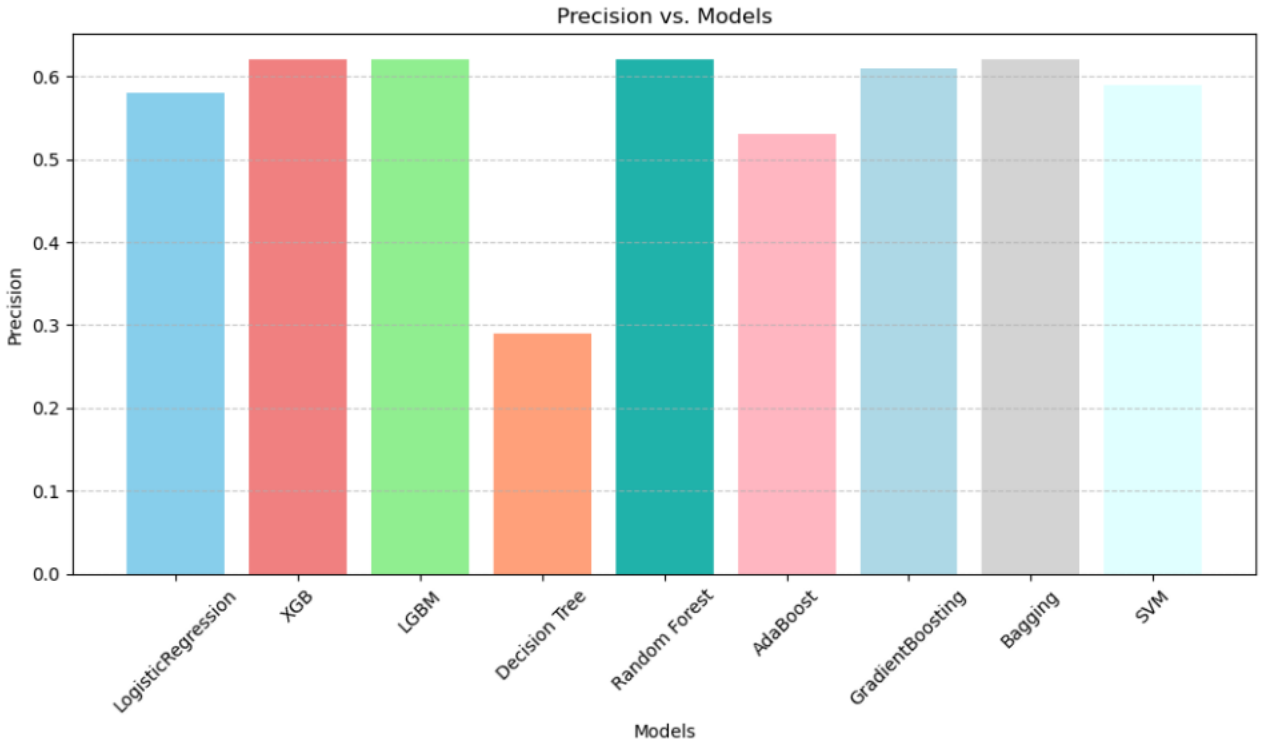

Fig. 5 *Precision vs. models*

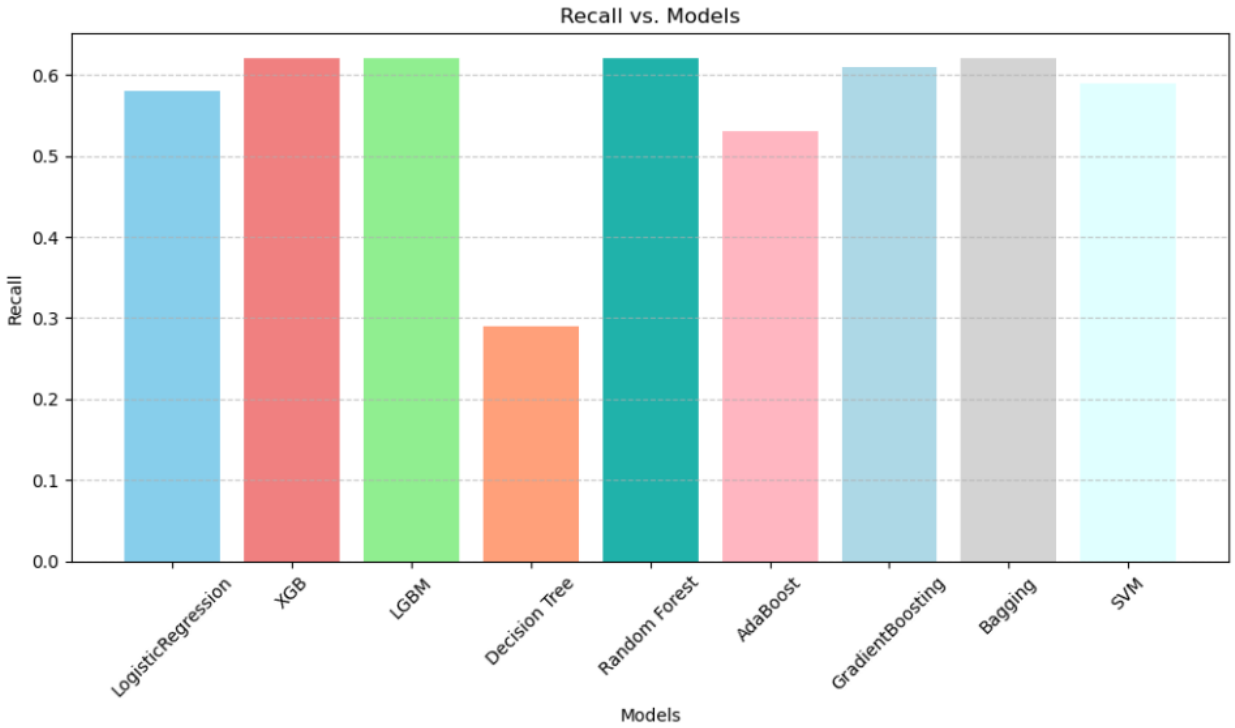

Fig. 6 *Recall vs. models*

J. of Advanced Industrial Technology and Application Vol. 6 No. 2 (2025) p. 37-47

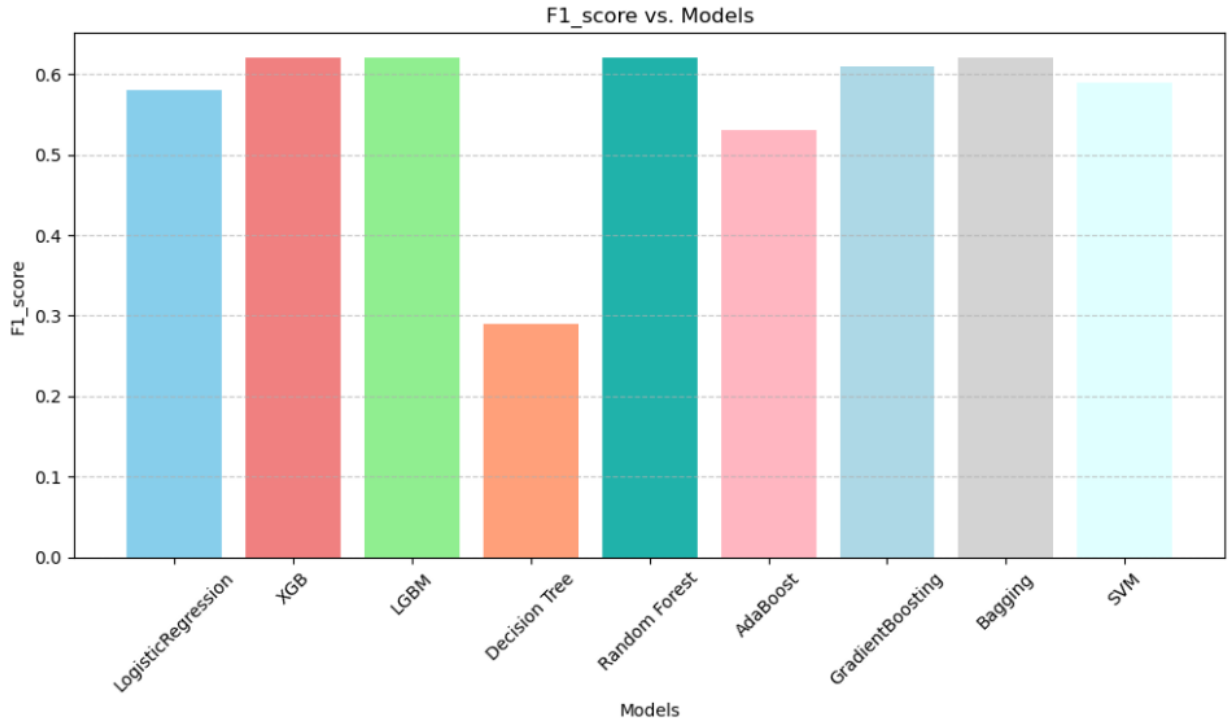

Fig. 7 *F1_score vs. models*

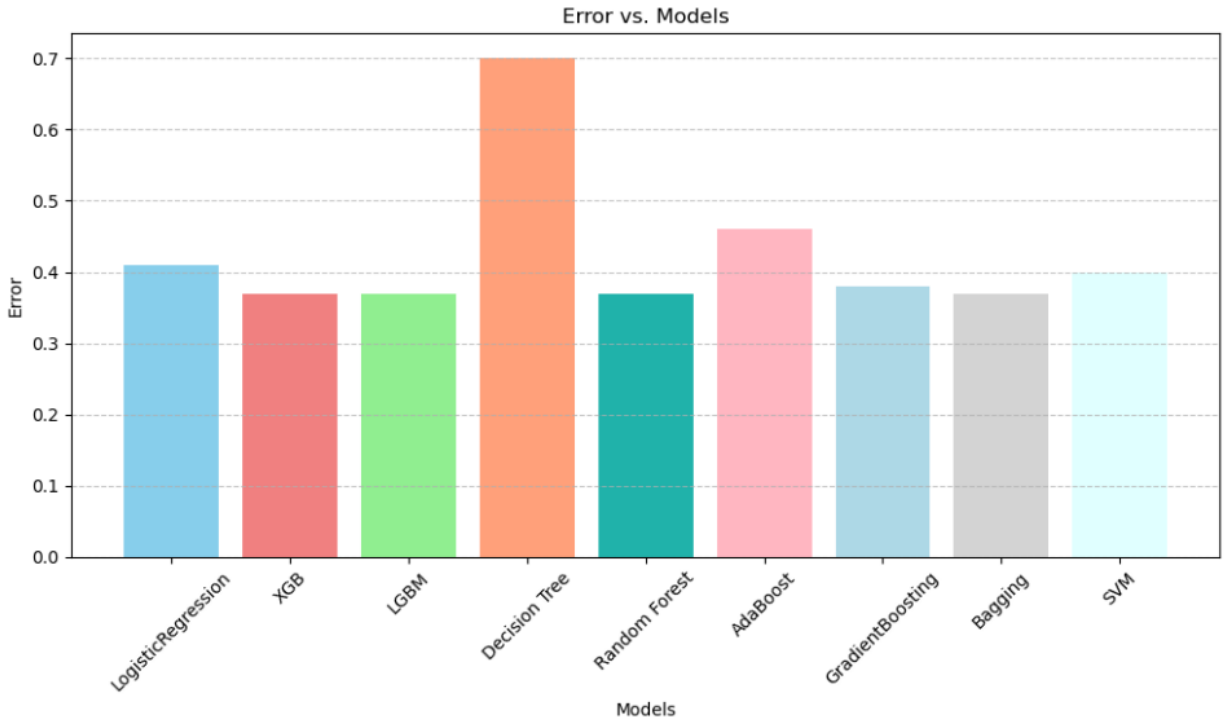

Fig. 8 *Error vs. models*

Evaluating the model performance is important to make informed decisions about the choice of algorithms. A number of metrics, including Accuracy, Precision, Recall, F1 Score, Specificity, NPV, FPR, FNR, and Error, provide a better level of understanding of a model's capabilities. The use of a variety of models such as Logistic Regression, XGBoost, LightGBM, Decision Tree, Random Forest, Gradient Boosting, and Bagging, offer valuable insights. Accuracy, the fundamental metric measuring overall correctness, is well-suited for balanced datasets, yet may falter in imbalanced situations. The F1 Score strikes a balance between Precision and Recall, making it suitable for

imbalanced datasets. This is due to the fact that Precision focuses on the accuracy of positive predictions, crucial when false positives carry substantial costs. On the other hand, Recall emphasizes capturing all positive instances. The capture here is significant when missing positives is costly, too. Specificity measures a model's ability to correctly identify negative instances, particularly vital in critical scenarios. By doing so, it is complementary to Recall. Negative Predictive Value (NPV) becomes imperative when avoiding false negatives holds paramount importance. Meanwhile, providing insights into specific error types, False Positive Rate (FPR) and False Negative Rate (FNR) offer a deeper understanding of the model behavior. When studying the models, Logistic Regression may not be able to handle complex relationships. On the other hand, other models such as XGB and LGBM which make use of ensemble methods and optimized frameworks have a better chance to exhibit robustness and efficiency. Interpretable though prone to overfitting, Decision Trees can aid in understanding feature importance. Random Forest can mitigate overfitting. Gradient Boosting excels in predictive accuracy, albeit with hyperparameter sensitivity. Bagging, utilizing bootstrap aggregating, reduces overfitting and variance. Fig. 9 shows the recap of all models compared to the metrics in with its value. So, it shows clearly the best and worst model based on its performance.

In summary, the choice of metrics and models must align with the intricacies of the dataset and the specific goals of the task at hand. By scrutinizing these metrics across different models, practitioners can discern trade-offs, strengths, and weaknesses, ultimately guiding the selection of models that best suit the complexities of the real-world problem being addressed. So, this study recommends the models showing the best performance, especially in terms of accuracy and error, to be used in the detecting process.

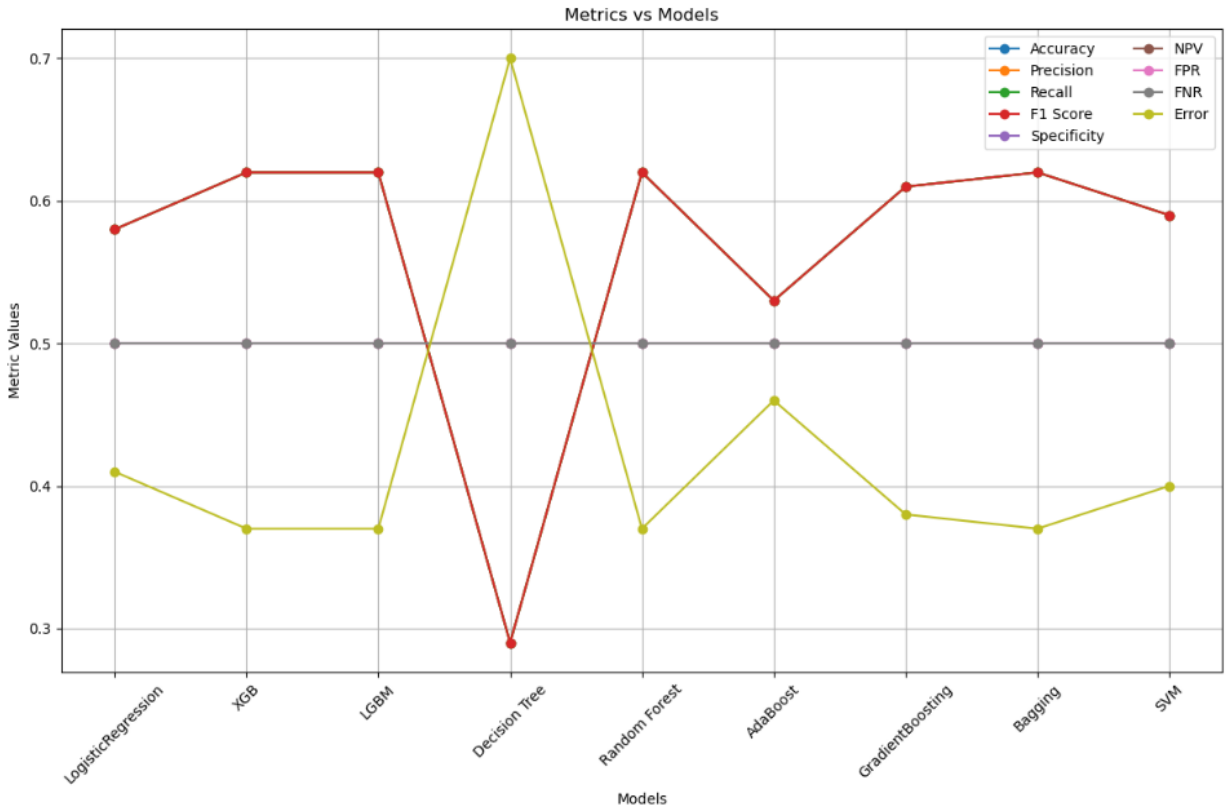

Fig. 9 *Metrics vs. models*

6. Conclusion

This study systematically evaluated the performance of various machine learning models for Windows malware detection, employing a diverse dataset and comprehensive set of performance metrics. The results indicate notable variations in the effectiveness of different algorithms, with each model exhibiting strengths and weaknesses in specific areas. The analysis of accuracy, precision, recall, F1 score, specificity, negative predictive value, false positive rate, false negative rate, and error revealed that; Logic Regression, XGB, LGBM, Decision Tree, Random Forest, Ada Boost, Gradient Boost, Bagging, and SVM, outperformed others in certain aspects of malware classification. This nuanced understanding of model performance is crucial for informed decision-making in deploying effective malware detection systems. Additionally, the study underscored the significance of feature selection and extraction techniques, stressing their influence on model efficacy. Future research could delve

deeper into optimizing these techniques to further enhance the robustness and efficiency of malware detection models. The insights gained from this research contribute to the ongoing efforts to bolster cybersecurity measures, particularly in the context of Windows environments. This research provides insights that enhance cybersecurity safeguards, especially within Windows environments. The findings presented here provide valuable guidance for practitioners and researchers working towards developing more resilient and effective solutions in the ever-changing landscape of cybersecurity. In conclusion, this study advances our understanding of the strengths and limitations of diverse machine learning models in Windows malware detection, paving the way for further improvements that address emerging challenges in the field.

Acknowledgement

The authors would like to thank all parties who contributed directly or indirectly to this research.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

All authors contributed equally to this work and manuscript preparation.

References

[1] A. S. George, A. H. George, and T. Baskar, "Digitally immune systems: building robust defences in the age of cyber threats," *Partners Universal International Innovation Journal,* vol. 1, no. 4, pp. 155-172, 2023.

[2] M. Imran, H. U. R. Siddiqui, A. Raza, M. A. Raza, F. Rustam, and I. Ashraf, "A performance overview of machine learning-based defense strategies for advanced persistent threats in industrial control systems," (in English), *Comput. Secur.,* Article vol. 134, p. 12, Nov 2023, Art no. 103445, doi: 10.1016/j.cose.2023.103445.

[3] I. Firdausi, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in *2010 second international conference on advances in computing, control, and telecommunication technologies*, 2010: IEEE, pp. 201-203.

[4] J. C. Kimmell, M. Abdelsalam, M. Gupta, and Ieee, "Analyzing Machine Learning Approaches for Online Malware Detection in Cloud," in *7th IEEE International Conference on Smart Computing (SMARTCOMP)*, Electr Network, Aug 23-27 2021, NEW YORK: Ieee, 2021, pp. 189-196, doi: 10.1109/smartcomp52413.2021.00046. [Online]. Available: <Go to ISI>://WOS:000853326000026

[5] S. S. Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A Survey on malware analysis and mitigation techniques," (in English), *Comput. Sci. Rev.,* Review vol. 32, pp. 1-23, May 2019, doi: 10.1016/j.cosrev.2019.01.002.

[6] C. Kolbitsch, P. M. Comparetti, C. Kruegel, E. Kirda, X.-y. Zhou, and X. Wang, "Effective and efficient malware detection at the end host," in *USENIX security symposium*, 2009, vol. 4, no. 1, pp. 351-366.

[7] R. Damaševičius, A. Venčkauskas, J. Toldinas, and Š. Grigaliūnas, "Ensemble-based classification using neural networks and machine learning models for windows pe malware detection," *Electronics,* vol. 10, no. 4, p. 485, 2021.

[8] A. Hussain, M. Asif, M. B. Ahmad, T. Mahmood, and M. A. Raza, "Malware detection using machine learning algorithms for windows platform," in *Proceedings of International Conference on Information Technology and Applications: ICITA 2021*, 2022: Springer, pp. 619-632.

[9] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," *Informatics*, vol. 8, no. 1, p. 10, 2021.