International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Optimizing Malware Detection Through Advanced Ensemble Methods

Bushiri Mwamba Jean Marie[1], Mrs. Shivangi Bansal[2]
Research Scholar[1], Noida International University
Assistant Professor[2], Noida International University
Email id: jeanbushiri64@gmail.com

Abstract: One of the biggest problems that institutions & organizations face is information security. The frequency and scope of cybercrime have increased recently, as new methods for stealing, altering, and destroying data or taking down computer networks are emerging daily. Malware is one kind of intrusion that can occur into information systems that handle sensitive data. When malware is introduced into a computer system, the attacker gains full or partial access to the system's vital data. An ensemble classification-based detection of malware algorithm is proposed in this paper. A layered ensemble of dense (fully connected) and CNN handles the initial step of classification, and a meta-learner handles the last stage of classification. We investigate and contrast 14 classifiers for a meta-learner. Eight ML techniques are KNN, SVM, Random Forest, AdaBoost, Decision Tree, Logistic Regression, Gradient Boosting, XGB, and the suggested approach are utilized as a baseline comparison. We report on the studies conducted using the CSV or XLS file datasets for the classification of malware. The results all demonstrate how simple it is to identify malware in software files using the suggested method.

Keywords: Cloud Computing, Malware detection, Machine Learning, Deep learning, Ensemble Learning.

I.INTRODUCTION

Cloud computing has become the preferred option for numerous corporate and public sector organizations because of its scalability & ease of use. Providing resources on-demand utilizing the pay-as-you-go paradigm, if and when required is one of the primary features of the cloud. The benefits of the cloud are generally hampered by the need for IT specialists at businesses to oversee and administer these resources. With the advent of cloud automation solutions, IT staff may now provide resources in the cloud autonomously. By using tools and configuration programs that can build, edit, and remove cloud resources, this kind of automation is made possible.

Although these orchestration tools greatly assist DevOps teams, they also increase the surface area for potential security breaches. Specifically, virtual machines are frequently created by automated configuration instruments, resulting in a huge number of identical or almost identically configured virtual machines. Because of these virtual machines' inherent redundancy, malware may spread quickly across them, particularly if these configuration programs have flaws. A single compromised VM has significantly less consequences than a collection of vulnerable VMs. Because cloud infrastructure is inherently complex and operates in a dynamic environment where threats are always changing and expanding, it requires significant security implementations. It is crucial to create rapid and accurate malware detection techniques for the same reason [1].

One of the biggest threats to cloud systems is malware. With benefits and drawbacks, several malware detection techniques have been put forth. An executable's signature is examined and contrasted with a database of known malware signatures in the widely used technique known as "static malware identification" [2]. Attackers have attempted to reduce the efficacy of static analysis through the use of packing and methods of obfuscation. Furthermore, static malware analysis can only identify executables of known malware; it cannot identify the constantly changing zero-day malware. Many studies on behavioral malware detection techniques have been conducted as a result of these two significant drawbacks. Two behaviorally based techniques for detecting malware are dynamic and online. In order to identify dynamic malware, the malicious executables are run in a protected environment, like a sandbox, and their activity is observed. By doing this, the detection system can study new zero-day malware because it is analyzing the executable's real activity rather than relying on previously known signatures. Nevertheless, bad actors have managed to introduce malware that recognizes when a sandbox is being used and stops acting nefariously to evade detection. Both dynamic and static approaches have the same drawback, which is that the detection system concentrates on finding malware in the provided executables prior to their execution on real systems. Nevertheless, malware frequently enters a system through security holes, eluding these antiquated methods of detection. Online malware detection [3–4] concentrates on a machine's behavior indicating it is attempting to defend against infection. Instead of examining applications or their actions, online techniques keep an eye on the virtual machine's overall performance and sound an alarm whenever any signs of malicious activity are discovered. Because they get around the drawbacks of static and dynamic malware detection techniques, online malware detection methods are therefore regarded as continuous real-time detection systems.

When there are enough dangerous programs or a wide enough variety of alternatives, ML is also frequently employed by security experts as a potent method to accurately identify harmful programs. The Windows Portable Executable 32-bit (PE32) file header analysis is one of the primary techniques [5]. Nisa et al. [6], for instance, converted malware code into pictures and used segmentation-based fractal texture analysis to extract features. For classification, two deep neural networks AlexNet & Inception v3 were employed. In the past, ML systems' abilities to identify malware in IoT environments or WSN were enhanced by the use of ensemble approaches like RF and extremely randomized trees.

Numerous research projects are being conducted to examine malware in an effort to stop the spread of malicious software. CNN, DBN [7], graph convolutional networks (GCN), LSTM and Gated Recurrent Unit (GRU) [8], VGG16, and generative adversarial networks (GAN) [9] are some of the DL based malware detection methods now in use. Nonetheless, the potential for generalization of algorithms based on ANN cannot be guaranteed.

Consequently, more generic & robust solutions are needed to overcome the aforementioned challenges. Many ensemble classifiers that are less vulnerable to malware feature gathering are being developed by researchers [10]. A class of methods known as ensemble methods [11] combines multiple learning approaches to improve the overall prediction accuracy. Several models for classification are integrated by these ensemble classifiers to reduce the likelihood of over fitting in the training results. This leads to a more efficient use of the training dataset and an improvement in generalization effectiveness. Even if a number of ensemble model classifications have already been created, researchers can still work to increase sample classification accuracy, which will help with recognizing malware. In order to overcome this, this research suggests an ensemble earning-based method for malware detection that uses convolution and completely linked neural systems as base learners. The following describes how the paper is organized: The Ensemble method is explained in Section II, the literature review is shown in Section III, the proposed study's findings, & discussions are explained in Sections IV and V, & the conclusion and future directions for the work suggested are provided in Section VI.

II. Ensemble Classification

Ensemble approaches work on the basic premise of rearranging training data sets in multiple manners (whether it's by resembling or reweighting) before constructing an ensemble of base classifiers by including a base classifier to each rearrange training set. Then, by integrating the forecasting impacts of all those base classifiers, a new ensemble classifier is created utilizing the stacked ensemble approach, where a novel model learns how to better integrate predictions from multiple base classifiers. The two-step stacking approach was employed. Initially, an inventory is used to train a number of models. Next, each model's output is processed to produce a fresh data set. Every instance in the present dataset is associated with the actual value that it is intended to represent. Second, the final output is obtained from the data set using the meta-learning process.

Base models, also known as level-0 models, are typically used in the construction of a stacking model (Figure 1), together with a meta-learner (or generalize) that combines base model projections, also known as a level-1 model. The base models are those that are compiled with forecasts and fit into the training data. A classification model trained to aggregate the base model's forecasts is called the meta-learner (level-1) model. Basic models provide the meta-learner with information about the decisions taken. The input & output value pairs from the training data are utilized to fit the meta-learner together with the predicted outputs provided by these forecasts. This process is repeated for learning the fundamental models using a fresh batch of formerly unused data.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

The ensemble learner approach contains of 3 phases:

1. Set up the ensemble:

a) Select N base learners;

b) Select a meta-learning algorithm.

2. Train the ensemble:

a) Train each of the N base learners on the training dataset (X_1, X_2,….X_M), where M is the set of instances;

b) Perform the k-fold cross-validation on each of the base learners and record the cross-validated predictions (Y_1, Y_2,….Y_M),

c) Combine cross-validated estimations from base learners to form a new feature matrix as obeys. Train the meta-learner on the new data (features x estimations from base- level classifiers) (X_1, X_2,….X_M , Y_1) (X_1, X_2,….X_M , Y_2),……..(X_1, X_2,….X_M , Y_n),, Integrate the meta-learner with base models of learning to produce forecasts on unknown data that are more accurate.

3. Conduct a test using fresh data by:

a) Documenting base learners' output decisions;

b) Forwarding base-level choices to the meta-learner for group decision-making.

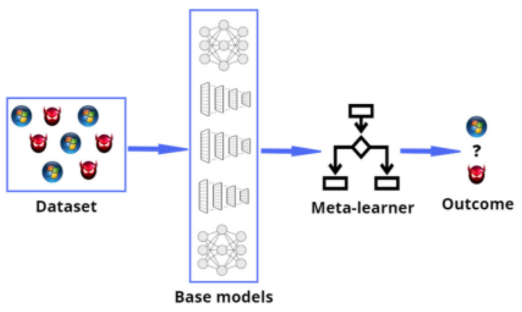

Figure 1. Schematics of ensemble classification approach[11]

III.LITERATURE SURVEY

Rajesh Kumar et al.,(2022) makes use of waterfall plots using Shapley value as a basis to identify trends in the characteristics responsible for incorrect classification. Inductive principles are derived from the trends in the five most important features for misclassification. In order to prevent misclassification and improve bagging method performance, inductive rules are utilized. Zero-day malware is a type of malware that is unknown in the future and can be detected using inductive principles, which can stop attacks on security systems. In the case of future undetected malware, the Extra tree bagging method's accuracy is 98.1%. Given that the inductive criteria also identify the incorrectly classified specimens, the accuracy is 100%[12].

Deepak Gupta et al.(2020) created two techniques to enhance malware detection on a wide scale, utilizing big data and ensemble learning. The first approach uses the ensemble learning weighted voting procedure, whereas the second selects the best possible set of base classifiers to be stacked. Utilizing Apache Spark, a well-liked large data processing framework, the suggested techniques are put into practice. Their effectiveness is assessed and tested on a dataset including 198,350 Windows files, of which 100,200 are malicious and 98,150 are benign samples. The proposed approach's efficacy is validated by the experimental results, as it enhances the generalization performance in identifying novel malware[13].

Apoorv Joshi et al.(2023) developed a ML malware detection model that uses stacking to identify malware on Android devices. The model building process uses four different machine learning models: Random Forest, Catboost, Histogram Gradient Boosting, & SVM. Using the two most current information sets, CIC-MalDroid 2020 and CIC-MalMem 2022, the efficacy of the suggested model is investigated. The model's accuracy is 99.99% and 98.0%, respectively. Furthermore, it was noted that the suggested model's performance surpassed that of certain cutting-edge models in terms of assessment metrics and classification accuracy[14].

Halit Bakır et al.,(2024) suggested using the random search optimization approach to determine the models' structure that will be utilized as ensemble classifier voter classifiers. CNN-ANN, pure CNN, and pure ANN are the three DL models that have been constructed using this optimization technique. The three models that were chosen have been trained and evaluated using the created picture dataset, with the optimal structure for each DL model having been chosen. Subsequently, we proposed combining the optimized three deep learning models into a hybrid model that combines two distinct working modes: MMR (Malware Minority Rule) and LMR (Label Majority Rule). This is the first instance of an ensemble classifier for malware detection that we are aware of that has been refined and hybridized in this manner. The findings

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

demonstrated the promise of the suggested models, with classification accuracy surpassing 97% across all tests[15].

IV.PROPOSED WORK

- **Methodology**

The proposed work methodology involves a comprehensive approach to building a robust machine learning model. The first step is to read the dataset from a CSV or XLS file, ensuring a solid foundation for subsequent analyses. Following this, data preprocessing techniques will be applied to handle missing values and eliminate NaN entries, ensuring the dataset's integrity. To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) will be employed to balance the dataset. Feature selection plays a crucial role in enhancing model efficiency and interpretability. This proposal suggests using Principal Component Analysis (PCA) or an Extra Tree-based classifier to identify and retain the most relevant features. Subsequently, the dataset will be divided into training and testing sets to assess model performance accurately. Three diverse machine learning algorithms, namely Random Forest (RF), Logistic Regression (LR), and XGBoost, will be trained individually. To harness the collective strength of these models, an ensemble learning approach using a Voting Classifier will be implemented, providing a more robust and accurate prediction. Finally, the proposed methodology emphasizes thorough performance evaluation. Metrics such as accuracy, precision, recall, and F1 score will be employed to assess the models' effectiveness. This holistic approach ensures a systematic and rigorous process to develop a reliable and high-performing machine learning model for the given dataset.

- **Objectives**

1.To implement effective data pre-processing technique for better accuracy
2.To implement ensemble learning with voting classifier to improve classification performance
3.To perform comparative analysis of proposed work.

Figure 2:Flowchart of proposed work

V.RESULTS

The Cloud Computing Environment requires the use of Data Mining Techniques. Cloud offers all of the software and hardware to its clients as online services. Thus, in order to find relevant patterns among the vast amount of data that is now available in the form of samples, this work suggested cloud mining techniques. As a result, the Cloud Server can receive the.apk files in order to use the ensemble classification model to identify malware. The server will next determine whether or not the.apk file includes malware. By doing this, the malware can be found utilizing the Android program on the.apk files. The accuracy score, True Positive Rate (TPR), and False Positive Rate (FPR) are the metrics use to report the outcomes. Accuracy is defined as the ratio of correct classifications to total classification tries. TPR is the proportion of malware samples correctly categorized as malware, and FPR is the fraction of non-malware samples wrongly classified as malware.

A tabular summary of a classification model's performance that shows the model's predictions on a dataset is called a confusion matrix. It is frequently used to clearly visualize true positive, true negative, false positive and false negative predictions in machine learning tasks, particularly in binary and multiclass classification issues.

The confusion matrix is a vital instrument for evaluating the effectiveness of a classification model. It is frequently used in conjunction with other assessment metrics to give a thorough picture of the model's ability to make accurate predictions.

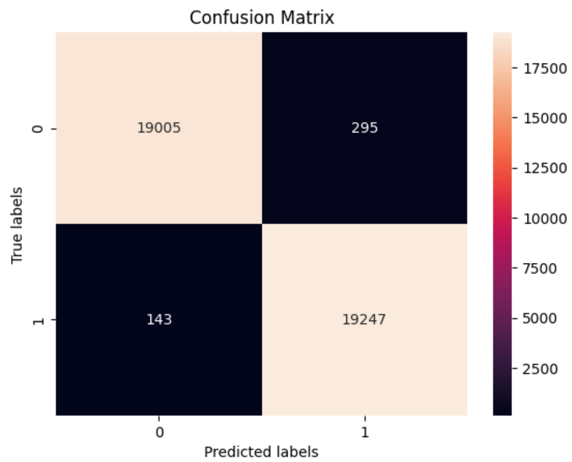

Figure 3:Confusion Matrix

Accuracy:

The fundamental requirement for the execution of arrangements is accuracy. The total number of correct instances is a measure of precision, whereas incorrectly arranged events do not form an efficient cluster when it comes to error rates.

$$\text{Accuracy} = \frac{\text{True Positive+True Negative}}{\text{True Positive+False Positive+True Negative+False Negative}}$$

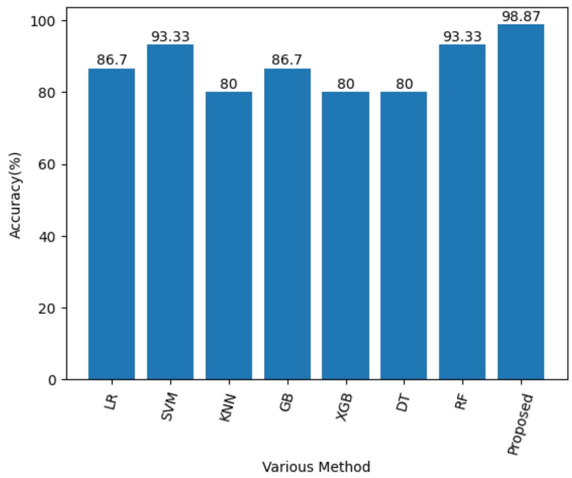

Figure 4: Accuracy

Figure 4 shows how different approaches to malware detection differ in terms of accuracy percentages; however, the recommended strategy is the most successful, as evidenced by its 98.87% detection rate.

Table 1: Accuracy for Various Approaches

Technique	Accuracy(%)
LR	86.7%
SVM	93.33%
KNN	80%
GB	86.7%
XGB	80%
DT	80%
RF	93.33%
Proposed	98.87%

True Positive Rate (TPR): Performance statistic TPR, sometimes known as sensitivity or recall, is employed in binary classification problems. The percentage of actual positive cases that a classification model correctly classifies is computed. Stated differently, it assesses the model's ability to predict positive events with accuracy out of all actual positive events.

$$\{\text{TPR}\} = \frac{\mathit{True\ Positive}}{\{\mathit{True\ Positives}\} + \{\mathit{False\ Negatives}\}\}]}$$

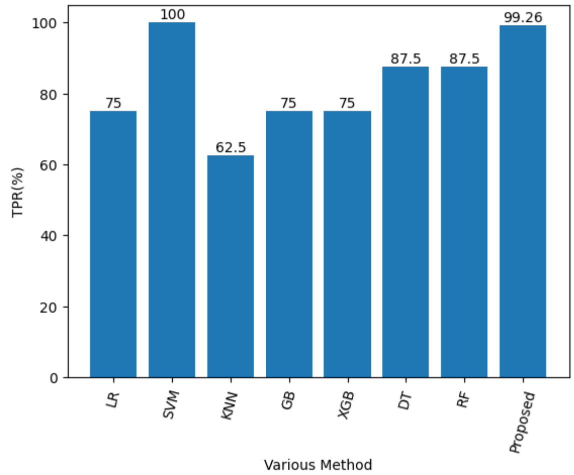

Figure 5:TPR

Figure 5 shows the TPR for the different approaches, showing that the recommended method correctly displays the instances with a 99.26% proportion.

False Positive Rate (FPR): In binary problem solving, the proportion of negative instances that a classification model incorrectly classifies as positive is known as the FPR, which is an indicator of performance. Put another way, it calculates the false alarm rate that is raised by the model when it misinterprets negative events as positive ones.

An excessive false positive rate (FPR) suggests that the model might be incorrectly labeling negative situations as positive, which could lead to incorrect data interpretations or unnecessary actions. Because of this, lowering the FPR is frequently necessary to raise the model's overall effectiveness.

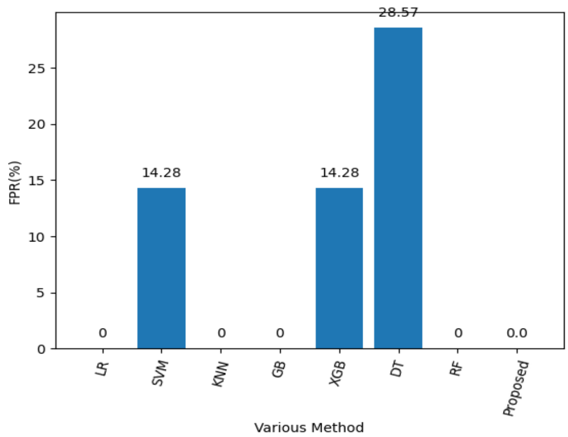

Figure 6:FPR

It is clear that Figure 6 shows zero results for inaccurate classifications for all techniques, in contrast to SVM, XGB, and DT. All of the outcomes show how easy it is to find malware in software files by applying the recommended technique.

VI.CONCLUSION

In this work, we examined and empirically verified the utilization of ensemble learning to merge the malware prediction outcomes provided by various DL and ML models. This procedure aims to increase the detection of Windows PE malware. It is not necessary to use a specific ML model when using ensemble approaches. Instead, a learning technique that yields the optimum malware detection performance is created by aggregating the prediction capabilities of each combination of ML models. In order to develop effective & efficient malware detection designs, we combined DL & ML approaches with lightweight fully integrated and CNN architectures to investigate our suggested ensemble classification approach. For a fair comparison, we carried out in-depth tests on a range of ML models and lightweight deep learning architectures within the ensemble learning structure under identical settings. The findings obtained demonstrate that ensemble stacking is more effective at detecting malware than other ML techniques, such as neural networks.

We demonstrated how the malware detection issue may be effectively addressed by the ensemble learning framework built on lightweight deep neural networks. The outcomes show that methods of ensemble learning can be applied and employed as clever malware identification strategies. The future study's texture evaluation and categorization techniques still have a lot of room for improvement. Therefore, by integrating a mixed model with texture, texture-based picture interpretation and classification from a huge dataset can be improved.

REFERENCES

[1] M. R. Watson, A. K. Marnerides et al., “Malware detection in cloud computing infrastructures,” IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 192–205, 2015.

[2] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, “Intelligent vision-based malware detection and classification using deep random forestparadigm,” IEEE Access, vol. 8, pp. 206303–206 324, 2020.

[3]M. Abdelsalam, R. Krishnan et al., “Malware detection in cloud infrastructures using convolutional neural networks,” in IEEE International Conference on Cloud Computing (CLOUD), 2018, pp. 162–169.

[4] A. McDole et al., “Analyzing CNN based behavioural malware detection techniques on cloud IaaS,” in International Conference on Cloud Computing (CLOUD). Springer, 2020, pp. 64–79

[5]Ye, Y.; Wang, D.; Li, T.; Ye, D.; Jiang, Q. An intelligent PE-malware detection system based on association mining. J. Comput. Virol. 2008, 4, 323–334.

[6]. Nisa, M.; Shah, J.H.; Kanwal, S.; Raza, M.; Khan, M.A.; Damaševičius, R.; Blažauskas, T. Hybrid Malware Classification Method Using Segmentation-Based Fractal Texture Analysis and Deep Convolution Neural Network Features. Appl. Sci. 2020, 10, 4966.

[7] Ren, Z.; Wu, H.; Ning, Q.; Hussain, I.; Chen, B. End-to-end malware detection for android IoT devices using deep learning. Ad Hoc Netw. 2020, 101, 102098.

[8] Čeponis, D.; Goranin, N. Investigation of Dual-Flow Deep Learning Models LSTM-FCN and GRU-FCN Efficiency against Sin- gle-Flow CNN Models for the Host-Based Intrusion and Malware Detection Task on Univariate Times Series Data. Appl. Sci. 2020, 10, 2373.

[9] Martins, N.; Cruz, J.M.; Cruz, T.; Abreu, P.H. Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review. IEEE Access 2020.

[10] Gupta, D.; Rani, R. Improving malware detection using big data and ensemble learning. Comput. Electr. Eng. 2020, 86, 106729.

[11] Sagi, O.; Rokach, L. Ensemble learning: A survey. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2018.

[12] Rajesh Kumar,Geetha Subbiah, "Explainable Machine Learning For Malware Detection Using Ensemble Bagging Algorithms", Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing, pages 453–460,2022.

[13] Deepak Gupta, Rinkle Rani , "Improving malware detection using big data and ensemble learning", Computers & Electrical Engineering,Volume 86, September 2020, 106729

[14] Apoorv Joshi & Sanjay Kumar, "Stacking-based ensemble model for malware detection in android devices", International Journal of Information Technology ,Volume 15, pages 2907–2915, 2023.

[15] Halit Bakır, "VoteDroid: a new ensemble voting classifier for malware detection based on fine-tuned deep learning models", Multimedia Tools and Applications,2024.