International Journal of Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2023.9.7.3

Volume 9, Issue 7
July - 2023

Malware Mitigation Through Detection Using Support Vector Machine and Random Forest Algorithm

Agu, Edward .O.[1], Ebelogu, Christopher Ubaka[2]

[1]Research Scholar, Computer Science Department

Federal University Wukari, Taraba, Nigeria

[2] Computer Science Department

University of Abuja, FCT Abuja, Nigeria

ABSTRACT

Due to the ever-growing threat of malware application, diverse malware detection mechanism has been developed by researchers. Malware detection relates to the procedure of finding malware on a host device or determining whether a particular program is malicious or benign. An instance of a malware detection mechanism is an anti-malware program designed to automatically identify malicious programs from the benign program to prevent damage to the host system. The methodology used incorporated cutting-edge detection techniques to provide an effective solution to the problem of malicious programs. This study applied a support vector machine and random forest algorithm on malware detection using a dataset obtained from the Kaggle machine learning repository webpage. In an approach to provide a feasible solution, this study structured three methodical approaches that encompass data filtering techniques referred to as preprocessing and the utilization of the correlation metric to select the most relevant features in the first phase. The second approach involves the application of the filtered and selected dataset attributes and tuples to the adapted machine learning models in particular the random forest algorithm and the support vector machine. The final phase as an approach covers the evaluation of the derived model performance using metrics such as precision, accuracy score, and, f1_score. From the statistical result from the two models concerning the evaluation metrics also, it can be deduced that the random forest classifier performs more effectively in the detection of malicious malware from the dataset sourced from the Kaggle machine learning repository.

Keywords: Detection, Mitigation, Malware, Random Forest, Support Vector.

1.0 INTRODUCTION

Over the past decades, malicious application is increasingly becoming a significant threat to cybersecurity due to the increased use of Internet technology. Malware is a program software with malicious behavior designed explicitly to penetrate or damage a computer device without the owner's permission. Hence, malware when it is successfully present in its host causes great damage with a potential risk of destroying the files system and the host itself. This implies that there is a need for adequate security of data transmitted over the network by mitigating the tools and methods [1, 2].

Due to the ever-growing threat of malware application, diverse malware detection mechanism has been developed by researchers. Malware detection relates to the procedure of finding malware on a host device or determining whether a particular program is malicious or benign. An instance of a malware detection mechanism is an anti-malware program designed to automatically identify malicious programs from the benign program to prevent damage to the host system [3]. Malware researcher that conducts malware analysis in detecting malware range from the field of academic to industrial practice. Taking into cognizance academic practice, [4] has investigated the use of evolutionary computing techniques to automatically develop new variants of mobile malware that can successfully evade static-based anti-malware systems to develop better security solutions against them. Furthermore, [5, 6] implemented a deep learning model that can automatically detect whether or not an application is infected with malware. Recently machine learning techniques have been widely applied for classification problems, more generic for malware detection as they help to eliminate the difficulty in the manual way of crafting and updating malware detection patterns. [7] applied the ensemble decision tree forest to malware detection and were able to achieve a good result. Furthermore, other researchers [8] applied shared nearest neighbor (SNN) to discover new malware families. All this signifies the variabilities of the machine learning approaches to solving malware detection problems. Hence, considering the increasing attempt to alleviate malware by several researchers, incorporating cutting-edge detection techniques is essential to providing an effective solution to the problem of malicious programs. Hence, this study proposes the application of the support vector machine and random forest algorithm on malware detection using a dataset obtained from the Kaggle machine learning repository webpage.

2.0 LITERATURE REVIEW

Miramirkhani, N et al., proposed an evading malware analysis system using wear and tear artifacts for spotless sandboxes. They realized the challenges of environment-aware malware that alter the behavior of the system and approaches taken by most researchers in an attempt to deal with malicious wares, which was mostly by ensuring that well-known properties of analysis environments are replaced with realistic values and that any instrumentation artifacts remain hidden. The authors further identified that for sandboxes implemented using virtual machines, malware detection can be achieved by scrubbing vendor-specific drivers, processes, BIOS versions, and other VM-revealing indicators, while more sophisticated sandboxes move away from emulation-based and virtualization-based systems towards bare-metal hosts. They also observed that as the fidelity and transparency of dynamic malware analysis systems improve, malware developers resort to other system characteristics that are indicative of artificial environments. Taking into cognizance all their observations they proposed a novel class of sandbox evasion techniques that exploit the "wear and tear" which inevitably occurs on real systems as a result of normal use. By moving beyond how realistic a system looks, to how realistic its past use looks, malware can effectively evade even on sandboxes that do not expose any instrumentation indicators, including bare-metal systems. They investigated the feasibility of their evasion strategy by conducting a large-scale study of wear-and-tear artifacts collected from real user devices and publicly available malware analysis services. The results of their experimental evaluation using simple decision trees which were applied to the analyzed data revealed an accuracy of 92.86% [9].

Bat-Erdene, M et al., presented a strategy for characterizing the packing algorithms of some unknown packed malware executable dataset. The authors measured the entropy estimations of the executables and change over the entropy estimations of a specific area of memory into typical representations. The authors utilized symbolic aggregate approximation (SAX), which is known to be viable for huge information changes. They further ordered the conveyance of images utilizing managed learning order strategies which were the credulous Bayes and bolster vector machines for recognizing pressing calculations. The result of the author's analysis revealed that their strategy can distinguish pressing calculations of a given executable with a high precision of 95.35%, a review of 95.83%, and an accuracy of 94.13 [10] in conclusion deduced that pressing calculations can be recognized through an entropy examination given a measure of the instability of the running procedures and without earlier information on the executable.

Yeo, M et al., proposed flow-based malware detection using a convolutional neural network and some machine learning models. The author utilized 35 different features extracted from packet flow, instead of the port numbers and protocols of selected packages. They utilized the stratosphere IPS project data to evaluate their model performance, in which nine different public malware packets and normal state packets in an uninfected environment were converted to flow data with Netmate, and the 35-features were extracted from the flow data. CNN, multi-layer perceptron (MLP), support vector machine (SVM), and random forest (RF) were applied for classification. Their experimental result revealed a greater than 85% accuracy, precision, and recall for all classes using CNN and RF.

2.1 Summary of Literature Review

Authors	**Algorithms**	**Best Model**	**Accuracy**
Miramirkhani (*et al.*, 2017).	Decision tree.	Decision tree.	92%%
Bat-Erdene *et al.*, (2017)	symbolic aggregate approximation	symbolic aggregate approximation	94.13%.
Yeo *et al.*, (2018)	CNN, multi-layer perceptron (MLP), support vector machine (SVM), and random forest (RF)	CNN and RF	85%

3.0 RESEARCH METHODOLOGY

To capitalize on the sheer size of modern datasets, optimizing the scalability and effectiveness of machine and deep learning algorithms in an attempt to provide a feasible solution to the problem of the increasing growth of malicious is of great importance to this study. Hence, in an approach to provide a feasible solution, this study structured three methodical approaches that encompass data filtering techniques most often referred to as preprocessing, and the utilization of the correlation metric to select the most relevant features in the first phase. The second approach involves the application of the filtered and selected dataset attributes and tuples to the adapted machine learning models in particular the random forest algorithm and the support vector machine. The final phase as an approach covers the evaluation of the derived model performance using metrics such as precision, accuracy score, and, f1_score. The methodological approach is shown in Figure 3.1.

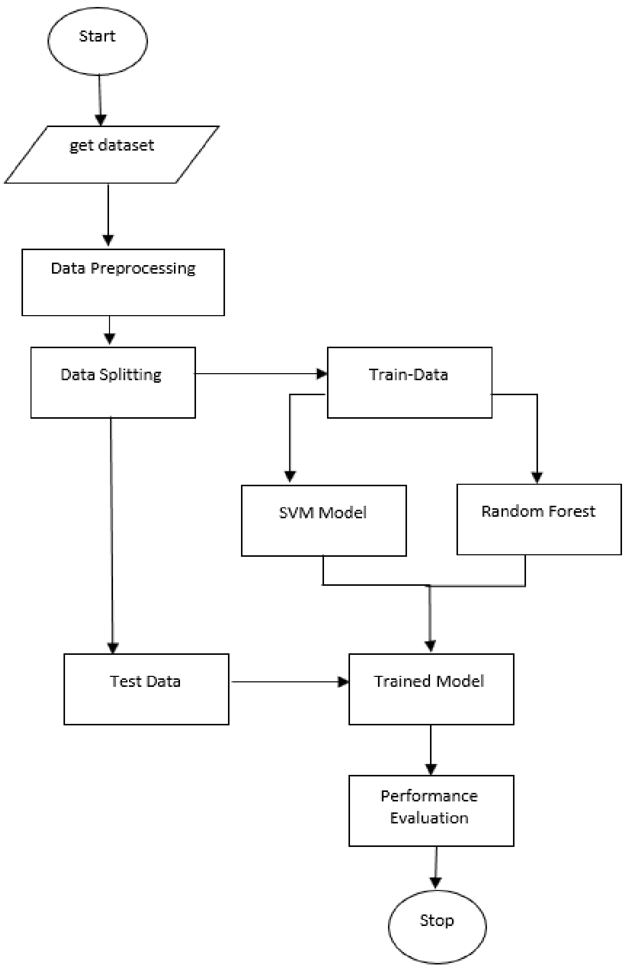

Figure 3.1: Research Flowchart

3.1 Dataset Description and Data Preprocessing

The dataset utilized for malware classification is the ClaMP (Classification of Malware with PE headers) integrated dataset sourced from the Mendeley data [12] from the Kaggle machine learning repository website. The dataset featured a total sample of 5000$^+$ malware datasets of the class label Malware or Benign with a record of 70 attributes out of which 54 were raw features while 16 were derived features.

The preprocessing step incorporates the identification and removal of null values before encoding the categorical data. Furthermore, the correlation matrix was applied to evaluate the dataset feature's relevance and select the influential attribute before the dataset is scaled and passed to the adapted machine learning classifier.

3.2 Classification Approach

Having identified the current problem of malware detection as a classification problem from the adopted dataset and considering that the approach proposed by this study is the machine learning techniques via the supervised machine learning techniques. Thus, this study proposed the utilization of the random forest classifier and the support vector machine as a supervised learning approach to the malware classification problem. The proposed training and testing proportion on the dataset before feeding both models is 70% for the training dataset and 30% for the test dataset. The splitting of the dataset also involves taking into account the relative populations of each malware family to ensure that each family is well represented on the split dataset.

3.2.1 Random Forest

Random forest is a supervised machine learning algorithm that was constructed from the decision tree algorithms. It can be used to solve regression and classification problems. The Random forest algorithm utilizes the concept of ensemble learning techniques which combines many classifiers to provide solutions to complex problems as in the case of the malware classification problem adopted by this study. The random forest algorithm produces a result based on the predictions of the decision trees by taking the average mean of the output of the various tree.

```
Algorithm 1: Random Forest (X, t, v)
Inputs: X – input data t – number of trees v – subsampling size
Output: a set of t, $itree$
1: Initialize $Forest$
2: set the height limit l = ceiling ($log^2 v$)
3: for i = 1 to t do
4:     X' ← sample (X, V)
5:     $Forest$ ← $Forest \cup itree(X, o, 1)$
6: end for
7: return $Forest$
```

3.2.2 Support Vector Machine

A Support Vector Machine just like the random forest algorithm is a supervised machine learning model that uses classification algorithms for two-group classification problems. The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N — the number of features) that distinctly classifies the data points. For instance, assuming the two data points from the diagram Figure 3.2 represents the malignant and benign class labels. To separate the two classes of data points, many possible hyperplanes could be chosen. Hyperplanes are decision boundaries that help classify the data points and it largely depends upon the number of features. Hence, to find a plane that has the maximum margin, the maximum distance between data points of both classes is considered.

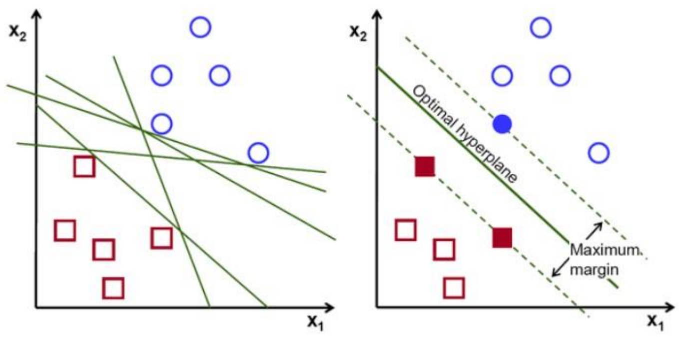

Figure 3.2: Support Vector Machine Hyperplane.

Algorithm 2: *Support Vector Machine*

```
SVM $cSv = \{closest\ pair\ from\ opposite\ plane\}$
initialize $i, v = violator, @ = violator\ checker$
while there are violating points do
    Find $v$        $cSv = cSv * v$
    if any $@ < 0$ then
        $cSv = cSv/i$
        repeat till all such points are pruned
    end if
    increment $i$
end while
```

3.3 Performance Evaluation Metrics

To evaluate the performance of the Random Forest and Support Vector Machine algorithm on the malware datasets obtained from Kaggle incorporation. This research utilized the following evaluation metric:

Precision: measures the classifier's accuracy as the percentage of the number of correctly predicted positive instances divided by the total number of predicted positive instances:

$$precision = \frac{TP}{TP + FP} \dots \dots \dots 3.1$$

Recall: is identified as the percentage of correctly predicted positive instances to the actual number of positive instances from the dataset.

$$Recall = \frac{TP}{TP + FN} \dots \dots \dots 3.2$$

F-measure (or F-score): defines the harmonic mean of precision and recall. It combines recall and precision metrics to obtain a score.

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots \dots \dots 3.3$$

Accuracy is measured as the percentage of the number of correctly predicted instances to the total number of instances present in the dataset. Thus, the accuracy calculates the ratio of inputs in the test set correctly labeled by the classifier:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots \dots 3.4$$

Where;

TP for True Positives defines the case where the actual class from a data record is true and thus predicted true by the model, TN for True Negative simulates a scenario where a data record was false and hence predicted false by the model, FP for False Positives defines an instance where the actual data record class is false but the model predicted true and finally, the FP for False Negatives which defines an instance where an actual data record point is true but the model predicted false.

4.0 RESULT AND DISCUSSION

To provide a feasible solution to the problem of malware detection, this study after a rigorous survey of kinds of literature on malicious wares, ventured into the classification approach on malware detection and thus applied two machine learning models namely the random forest classifier and the support vector machine on a malware classification dataset from Kaggle machine learning repository that contains 5210 malware records with 70 attributes. Of the 5210 malware records, 2722 matches the class label of malignant malware, and 2488 record corresponds to the class label of benign. Herein, to evaluate the performance of the model after applying the correlation metric as feature selection techniques (that resulted in the use of 61 attributes) on the filtered dataset from the preprocessing phase, this study utilized the viability of the precision, recall, f1_score, and accuracy score as the performance evaluation metric. The analytical deduction from the experiment conducted revealed the accuracy score of the random forest classifier as 98% with a training score of 99%, and the precision, recall, and fl_score, for the benign label were 0.98, 0.99, and 0.98 respectively, whereas for the class label malignant, the precision, recall, fl_score was 0.99, 0.98. 0.99 respectively. The classification report of the support vector machine is shown in Table 4.1. Considering the support vector machine algorithm, the analytical result of the statistical experiment revealed an accuracy score of approximately 76% with an equivalent training score of 76%. The precision, recall, fl_score, for the benign class label was 0.87, 0.58, and 0.70 respectively, whereas for the class label malignant, the precision, recall, fl_score was 0.70, 0.92. 0.80 respectively. The classification report of the support vector machine is shown in Table 4.2. Hence, from the analytical finding and deductions, the accuracy score of the random forest with an astonishing score of 98% outperforms the support vector machine that emerges with an accuracy score of 76%. So, therefore, it can be concluded that the random forest surpasses the support vector machine on the featured and scaled malware classification dataset sourced from the Kaggle machine learning repository.

Table 4.1: Random Forest classification report

		Random Forest		
	Precision	**Recall**	**F1_Score**	**Support**
0	0.98	0.99	0.98	630
1	0.99	0.98	0.99	673
accuracy			0.98	1303
Macro avg	0.98	0.98	0.98	1303
Weighted avg	0.98	0.98	0.98	1303

Table 4.2: Support Vector classification report

		Support Vector Machine		
	Precision	**Recall**	**F1_Score**	**Support**
0	0.87	0.58	0.70	630
1	0.70	0.92	0.80	673
accuracy			0.76	1303
Macro avg	0.79	0.75	0.75	1303
Weighted avg	0.78	0.76	0.75	1303

5.0 CONCLUSION

The survey conducted from several kinds of literature on the effect of malicious wares has prompted this study to investigate and apply the viability of the machine learning techniques that are currently the evolving discipline of study as they emerged to provide a feasible solution to real-world problems. In providing solutions to the problem of malware classification and providing insight into the best feasible techniques to tackle malicious wares, this study has thus, investigated the performance of the random forest classifier and the support vector machine model on a dataset obtained from Kaggle incorporation. To select the most correlative features from the dataset, the correlation metric feature technique was applied before the dataset was scaled and split into 70:30 trains to test proportion and was then fed to the two adopted machine learning models. The performance of the models was evaluated using precision, recall, f1_score, accuracy score, and confusion matrix, the result revealed the random forest surpassed the support vector machine with an accuracy score of 98%. Hence, from the statistical result from the two models concerning the evaluation metrics also, it can be deduced that the random forest classifier performs more effectively in the detection of malicious malware from the dataset sourced from the Kaggle machine learning repository.

REFERENCES

[1] Agu, E. O., Ejiofor V. E., Moses T., (2017) A Hybrid Model For Remote Dynamic Data Auditing (RDDA) On Cloud Computing. Journal of Computer Science and Application (JCSA). Vol. 24, No. 1, pp 96-116.

[2] Francisca N.O. and Edward O. A, (2015) A Mitigation Technique for Internet Security Threat of Toolkits Attack, International Journal of Computer Science and Security (IJCSS). ww.cscjournals.org/manuscript/Journals/IJCSS/Volume9/Issue5/IJCSS-1134.pdf. Vol.9, pp225-237

[3] Amira, B., and Sallow, E. (2020). An Investigation for Mobile Malware Behavioural and Detection Techniques Based on Android Platform. IOSR Journal of Computer Engineering (IOSR-JCE), 22(4), 14-20.

[4] Liu, Y., Guo, K., Huang, X., Zhou, Z., and Zhang, Y. (2018). Detecting Android Malware with High-Efficient Hybrid Analyzing Methods. Mobile Information Systems, 2018, 1–12. Doi: 10.1155/2018/1649703.

[5] Sabhadiya, S., Barad, J., and Gheewala, J. (2019). “Android Malware Detection using Deep Learning,” in 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 1254–1260, DOI: 10.1109/ICOEI.2019.8862633.

[6] Shen, F., Vecchio, J.D., Mohaisen, A., Ko, S.Y., and Ziarek, L. (2019). Android Malware Detection Using Complex Flows. Journal of IEEE, 8(6), 1231–1245, DOI: 10.1109/TMC.2018.2861405.

[7] Mahindru, A., & Sangal, A. L. (2021). MLDroid—Framework for Android malware detection using machine learning techniques. Neural Computing and Applications, 33(10), 5183-5240.

[8] Liu, L., Wang, B. S., Yu, B., & Zhong, Q. X. (2017). Automatic malware classification and new malware detection using machine learning. Frontiers of Information Technology & Electronic Engineering, 18(9), 1336-1347.

[9] Miramirkhani, N., Appini, M.P., Nikiforakis, N., and Polychronakis, M. (2017). Spotless Sandboxes: Evading Malware Analysis Systems Using Wear-and-Tear Artifacts. 2017 IEEE Symposium on Security and Privacy (SP), 2017, 1009-1024, Doi: 10.1109/SP.2017.42.

[10] Bat-Erdene, M., Park, H., Li, H., Lee, H., and Choi, M.S. (2017). Entropy analysis to classify unknown packing algorithms for malware detection. Information Journal of Information Security, 16(3),227–248. https://doi.org/10.1007/s10207-016-0330-4

[11] Yeo, M., Koo, Y., Yoon, Y., Hwang, T., Ryu, J., Song, J., Park, C. (2018). Flow-based malware detection using a convolutional neural network. 2018 International Conference on Information Networking (ICOIN), 2018, 910-913, Doi: 10.1109/ICOIN.2018.8343255.

[12] Dubey, A., Narang, S., Kumar, A., Sasubilli, S., & García Díaz, V. (2020, November 27). Performance Estimation of Machine Learning Algorithms in the Factor Analysis of COVID-19 Dataset. Computers, Materials and Continua, 66, 1921-1936. https://doi.org/10.32604/cmc.2020.012151