IJARST

International Journal of Advanced Research in Science and Technology

https://doi.org/10.62226/ijarst20241332

ISSN 2319 – 1783 (Print)
ISSN 2320 – 1126 (Online)

Enhancing Malware Detection through Machine Learning: A Comparative Analysis of Random Forest and Naive Bayes Classification Systems

D.Asir[1], Natheesh A[2], Shakeel Ahmed A[3], Manoj K[4]
[1,2,3,4] Department of Computer Science and Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, Tamilnadu, India

ARTICLEINFO

Article history:
Received 18 Mar 2024
Accepted 21 Mar 2024
Available online 26 Mar 2024

Keywords:
Malware Detection System (MDS),
F1-Score,
Machine Learning,
Random Forest,
Naïve Bayes,
Classification.

ABSTRACT

Malware, a type of malicious software encompassing viruses, worms, Trojans, backdoors, and spyware, poses a grave threat to the confidentiality, integrity, and functionality of computer systems, given their integral role in everyday life. To combat the escalating sophistication of malware attacks, deep-learning-based Malware Detection Systems (MDSs) have emerged as indispensable components of both economic and national security. Utilizing a dataset sourced from a repository, our research focuses on classifying observations into benign and malicious software for Android devices, employing machine learning algorithms such as Random Forest and Naïve Bayes. The dataset comprises 100,000 observations with 35 features, and our evaluation metrics encompass accuracy, precision, recall, and F1-score. This study underscores the significance of MDSs in safeguarding against evolving cyber threats, utilizing cutting-edge machine learning techniques to bolster defense mechanisms.

I. INTRODUCTION

In recent years, the widespread adoption of Android devices has rendered them a primary target for malicious entities aiming to exploit vulnerabilities for malicious intents. As malware sophistication continues to advance, traditional signature-based detection methods struggle to adapt to the rapidly evolving threat landscape. In response, integrating machine learning techniques has emerged as a promising avenue to bolster the detection and mitigation of Android malware.

This study investigates the application of machine learning, specifically leveraging the innovative Siamese Shot Learning technique, for Android malware detection. Siamese Shot Learning, a derivative of Siamese Networks, excels in security applications by learning robust data representations and effectively discerning between similar and dissimilar instances. The proposed methodology involves training a machine learning model using a diverse dataset of benign and malicious Android applications, extracting meaningful features such as permissions, API calls, and code patterns to distinguish legitimate from malicious behavior.

Malware, a term encompassing viruses, worms, Trojans, backdoors, and spyware, poses a significant threat to the confidentiality, integrity, and functionality of computer systems, given their ubiquitous role in modern life. With an estimated one in four computers in the U.S. affected by malware, the threat extends beyond emotional distress to significant financial losses, with up to one billion dollars stolen from financial institutions globally in recent years due to malware attacks.

Conventional signature-based detection methods, employed by anti-malware software products like Comodo, Symantec, and Kaspersky, rely on unique byte strings to identify known malware instances, but they are susceptible to evasion techniques such as encryption and polymorphism.

Additionally, the rapid proliferation of malicious files, reaching thousands per day, undermines the effectiveness of this signature-based approach. To address these challenges, intelligent malware detection techniques leveraging data mining and machine learning have garnered attention, with deep learning models demonstrating superior performance, particularly in analyzing extended sequences of system calls.

As malware detection remains a dynamic and evolving field in cybersecurity, the exploration of advanced techniques, including image processing for enhanced data visualization and decision-making, becomes increasingly imperative in the era of Big Data.

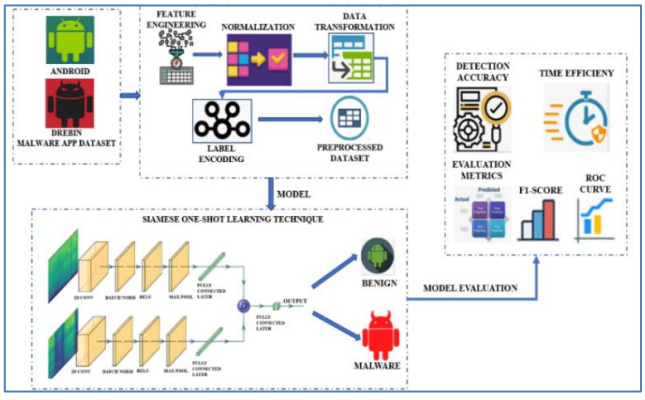


Figure 1: Overview of Proposed Model

Moreover, Siamese Shot Learning introduces the concept of "shots," allowing the model to effectively learn from a limited number of labeled examples. This feature proves particularly advantageous in the realm of Android malware detection, where obtaining labeled data can be challenging or costly.

The primary contributions of this study are as follows:

1. Development of a robust machine learning model tailored for Android malware detection.
2. Integration of the Siamese Shot Learning technique to improve the model's generalization capability, especially with limited labeled data.
3. Evaluation of the proposed approach using real-world datasets, showcasing its effectiveness in identifying both known and zero-day malware threats.
4. Discussion of practical implications and potential applications of the proposed methodology within the cybersecurity domain.

II. LITERATURE REVIEW

AMalNet: Introducing a novel deep learning framework called AMalNet, based on graph convolutional networks, to tackle the rising threat of malware in Android apps. Traditional detection systems often suffer from computational limitations or lack robustness. AMalNet overcomes these challenges by leveraging multiple embedding representations for malware detection and family attribution. It utilizes Graph Convolutional Networks to model high-level graphical semantics, Independently Recurrent Neural Networks to decode deep semantic information, and autonomously extracts important features without human intervention. While its complexity is high, experimental results demonstrate its significant outperformance compared to state-of-the-art techniques.

DL4MD: This paper explores a deep learning framework, DL4MD, for intelligent malware detection, focusing on Windows API calls extracted from Portable Executable (PE) files. The framework employs stacked Auto Encoders (SAEs) for unsupervised feature learning, followed by supervised fine-tuning. While it demonstrates feasibility in learning higher-level concepts, it exhibits lower accuracy in malware prediction.

Multi-level Deep Learning System: Addressing the escalating threat of sophisticated malware attacks, this system employs multi-level deep learning to enhance malware detection. Unlike traditional single deep learning models, this system recognizes that different sample subspaces may have unique data distributions. While it performs better than shallow learning models, it incurs lower time consumption and effectively captures complex malware data distributions.

Efficient Malware Detection with Multi-layered Random Forest Ensemble: Proposing a robust and efficient hybrid ensembling approach for malware detection, this method outperforms both machine learning and deep learning models with a remarkable accuracy of 98.91%. The adaptive nature of the model allows it to handle small-scale and large-scale data efficiently while consuming fewer computational resources and time compared to deep neural networks.

III. METHODOLOGY

In this system, the malware dataset sourced from repositories like the UCI repository serves as the input. Subsequently, feature selection techniques are applied to identify the most relevant features from the partitioned data. Following this, classification algorithms such as Random Forest and Naïve Bayes are implemented for the analysis. The experimental evaluation reveals performance metrics including accuracy, precision, and recall, demonstrating the effectiveness of the approach.

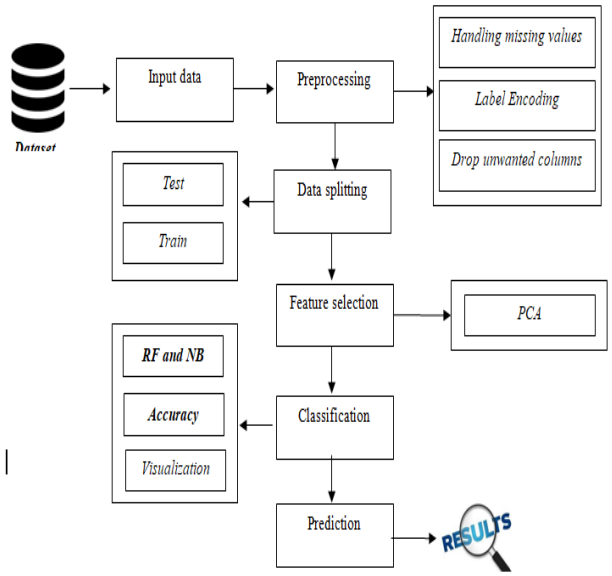


Figure 2: Architecture of Proposed methodology

Data Collection: The input data utilized in this project was gathered from a dataset repository, specifically focusing on malware detection. This dataset contains crucial information such as classification labels (malware or benign) and host details. Leveraging the pandas library in Python, the dataset, stored in a '.csv' file format, was read and prepared for further analysis.

Data Preprocessing: Data preprocessing involves the removal of irrelevant information and the transformation of the dataset into a format suitable for machine learning algorithms. One aspect of preprocessing involves handling missing data, where null values are replaced with zeros. Additionally, categorical data encoding is performed to convert variables with finite label values into a numerical format.

Data Splitting: To evaluate the performance of the machine learning algorithm, the dataset is split into training and testing subsets. In this project, 70% of the dataset is designated for training purposes, while the remaining 30% is reserved for testing. This partitioning facilitates the development and evaluation of predictive models, ensuring a comprehensive assessment of algorithm performance.

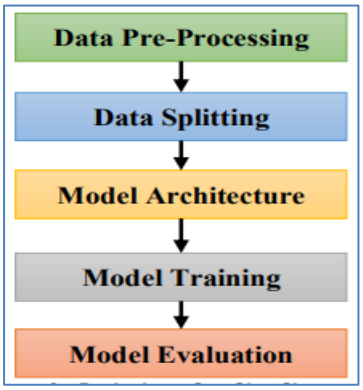


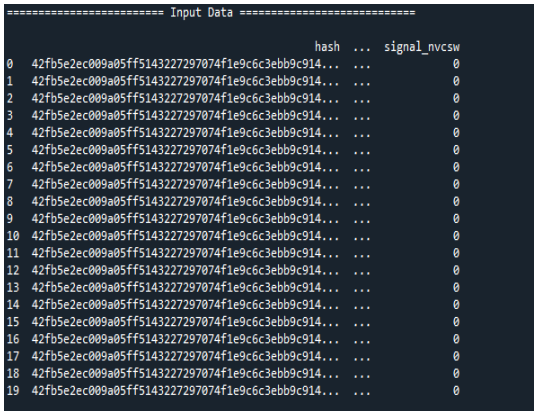
Figure 3: Steps for Designing One-shot Siamese Network

Feature Selection: In our workflow, feature selection is conducted employing techniques such as Principal Component Analysis (PCA). PCA is an unsupervised learning algorithm utilized for dimensionality reduction in machine learning. It employs a statistical process to transform correlated features into a set of linearly uncorrelated features through orthogonal transformation.
Classification: In our methodology, two machine learning algorithms are implemented: Random Forest and Naïve Bayes. Random Forest is a classification algorithm composed of multiple decision trees. It leverages bagging and feature randomness in constructing individual trees to create an ensemble of uncorrelated trees, enhancing prediction accuracy through committee voting.

Result Generation: The final outcome is derived based on the collective classification and prediction process. The performance of the proposed approach is evaluated using various measures, including accuracy. Accuracy reflects the classifier's capability to predict class labels correctly, calculated by the ratio of true positives and true negatives to the total number of predictions.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Data Collection and Labeling: Researchers typically compile a dataset of Android applications containing both benign and malicious samples. This dataset undergoes meticulous curation and labeling to ensure its suitability for training and testing in malware detection experiments.

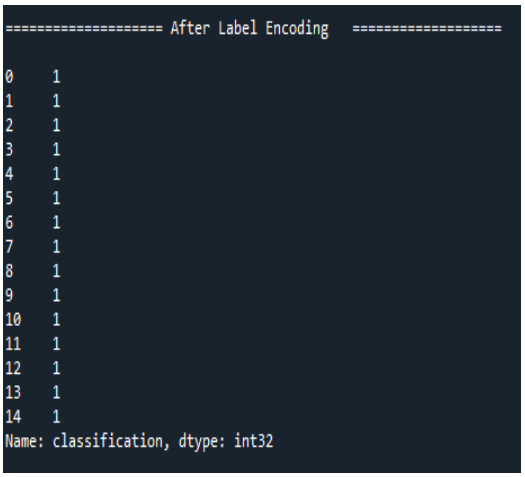


```
======================== Input Data ==========================

                                                hash  ...  signal_nvcsw
0   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
1   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
2   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
3   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
4   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
5   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
6   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
7   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
8   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
9   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
10  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
11  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
12  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
13  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
14  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
15  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
16  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
17  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
18  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
19  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
```

Figure 4: Figure Data Selection

Feature Extraction: Android applications can yield a plethora of features, including requested permissions, API calls, system calls, code structure, and more. These diverse features play a pivotal role in training machine learning models for malware detection.



```
=================== After Label Encoding  ===================

0    1
1    1
2    1
3    1
4    1
5    1
6    1
7    1
8    1
9    1
10   1
11   1
12   1
13   1
14   1
Name: classification, dtype: int32
```

Figure 5: Pre-Processing

Model Training: The extracted features are utilized to train machine learning models, which may include traditional classifiers like Support Vector Machines (SVM) or Random Forests, as well as advanced techniques such as deep learning models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs).

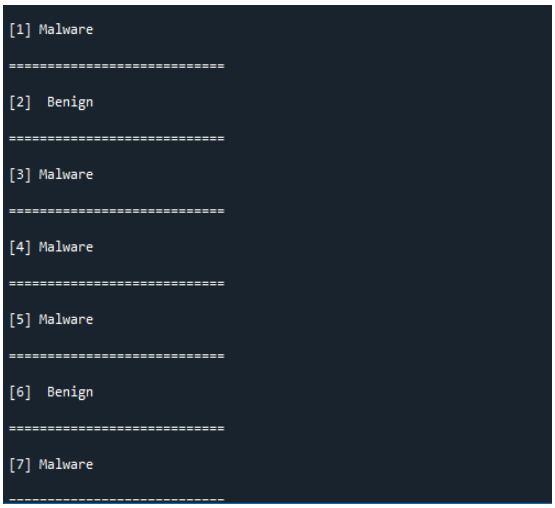
Siamese Shot Learning Technique: Siamese Shot learning, based on Siamese networks, offers a method to discern patterns in Android malware behavior and distinguish them from benign behavior, enhancing the detection capabilities.

Evaluation: Trained models are rigorously evaluated using a variety of metrics, including accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) curves. Cross-validation techniques are applied to ensure the reliability and robustness of the models.

Results and Analysis: The results are thoroughly analyzed to assess the effectiveness of both machine learning and Siamese Shot learning techniques in detecting Android malware. Researchers delve into false positives, false negatives, and any discernible patterns or trends uncovered during the analysis.

Comparison with Existing Methods: The proposed approach's performance is compared with existing methods for Android malware detection to gauge its efficacy and potential for improvement, providing valuable insights into its strengths and limitations.

Publication and Dissemination: The experiment findings are disseminated within the academic community through publication in journals or presentation at conferences, fostering knowledge sharing and further advancements in the fields of cybersecurity and machine learning.



```
[1] Malware

============================

[2]  Benign

============================

[3] Malware

============================

[4] Malware

============================

[5] Malware

============================

[6]  Benign

============================

[7] Malware

----------------------------
```

Figure 6: Classification

With the proliferation of smartphones and the continuous evolution of technology, the usage of Android applications has surged. However, this growth has also brought forth heightened concerns regarding security vulnerabilities inherent in these applications, making them susceptible to exploitation by malicious actors. Such vulnerabilities pose significant challenges for both researchers and developers tasked with ensuring the security of these applications. To mitigate potential security risks within this ecosystem, machine and deep learning techniques are deployed to proactively detect malicious attempts targeting Android applications. This research endeavors to bolster cybersecurity efforts by detecting anomalies within signature databases and empowering the system to recognize unknown threats through the innovative Siamese one-shot learning approach.

V.CONCLUSION

In summary, the utilization of machine learning and Siamese Shot learning techniques for Android malware detection presents a promising avenue for bolstering cybersecurity in mobile environments. Through training on meticulously curated datasets comprising benign and malicious Android applications, machine learning models exhibit prowess in accurately classifying and identifying malware based on extracted features. The amalgamation of traditional classifiers and deep learning algorithms further fortifies the development of robust detection systems. The incorporation of Siamese Shot learning techniques introduces an additional layer of sophistication, enabling systems to discern nuanced patterns and variations in malicious behavior. Ultimately, this approach furnishes a machine-learning based methodology for detecting malware attacks within software, delivering commendable performance across various algorithms such as random forest and logistic regression. One avenue for prospective research lies in integrating advanced feature extraction methodologies. As malware grows in complexity and sophistication, novel features such as dynamic behavioral analysis, opcode sequences, or semantic information derived from code structures could bolster the discriminative capabilities of detection models. Additionally, exploring ensemble learning techniques offers potential for combining the strengths of multiple classifiers, thereby enhancing detection systems' resilience against diverse malware families and evasion tactics. Techniques like stacking, bagging, or boosting could mitigate overfitting effects and enhance generalization performance across varied datasets and scenarios. Moreover, with the increasing interconnectedness of mobile devices and IoT ecosystems, future investigations may extend machine learning-based malware detection methodologies to broader contexts such as IoT device security, mobile app ecosystems, and cloud-based services. By embracing a comprehensive and proactive approach to cybersecurity, researchers can contribute to crafting adaptive defense mechanisms that fortify the integrity and privacy of digital ecosystems amidst evolving threat landscapes.

REFERENCES

[1] M. Alazab, S. Venkatraman, P. Watters, M. Alazab, and A. Alazab, ‘‘Cybercrime: The case of obfuscated malware,’’ in Global Security, Safety and Sustainability & e-Democracy (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 99, C. K. Georgiadis, H. Jahankhani, E. Pimenidis, R. Bashroush, and A. Al-Nemrat, Eds. Berlin, Germany: Springer, 2012.

[2] M. Alazab, ‘‘Profiling and classifying the behavior of malicious codes,’’ J. Syst. Softw., vol. 100, pp. 91–102, Feb. 2015.

[3] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, and J. Yearwood, ‘‘Hybrids of support vector machine wrapper and filter based framework for malware detection,’’ Future Gener. Comput. Syst., vol. 55, pp. 376–390, Feb. 2016.

[4] E. Raff, J. Sylvester, and C. Nicholas, ‘‘Learning the PE header, malware detection with minimal domain knowledge,’’ in Proc. 10th ACM Workshop Artif. Intell. Secur. New York, NY, USA: ACM, Nov. 2017, pp. 121–132.

[5] C. Rossow, et al., ‘‘Prudent practices for designing malware experiments: Status quo and outlook,’’ in Proc. IEEE Symp. Secur. Privacy (SP), Mar. 2012, pp. 65–79. [11] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas. (2017). ‘‘Malware detection by eating a whole exe.’’

[6] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, ‘‘Malware images: Visualization and automatic classification,’’ in Proc. 8th Int. Symp. Vis. Cyber Secur. New York, NY, USA: ACM, Jul. 2011, p. 4

[7] L. Nataraj and B. S. Manjunath. (2016). ‘‘SPAM: Signal processing to analyze malware.’’

[8] D. Kirat, L. Nataraj, G. Vigna, and B. S. Manjunath ‘‘SigMal: A static signal processing based malware triage,’’ in Proc. 29th Annu. Comput. Secur. Appl. Conf. New York, NY, USA: ACM, Dec. 2013, pp. 89–98.

[9] X. Glorot, A. Bordes, and Y. Bengio, ‘‘Deep sparse rectifier neural networks,’’ in Proc. 14th Int. Conf. Artif. Intell. Statist., Jun. 2011, pp. 315–323

[10] E. Raff et al., ‘‘an investigation of byte n-gram features for malware classification,’’ J. Comput. Virology Hacking Techn., vol. 14, no. 1, pp. 1–20, 2018.

[11] F. Alswaina and K. Elleithy, “Android malware family classification and analysis: Current status and future directions,” Electronics, vol. 9, no. 6, p. 942, 2020.

[12] W. Wang et al., “Constructing features for detecting android malicious applications: issues, taxonomy, and directions,” IEEE access, vol. 7, pp. 67602–67631, 2019.

[13] V. J. Raymond and R. J. Retna Raj, “Investigation of Android Malware Using Deep Learning Approach.,” Intell. Autom. \& Soft Comput., vol. 35, no. 2, 2023.

[14] N. Xie, Z. Qin, and X. Di, “GA-StackingMD: Android Malware Detection Method Based on Genetic Algorithm Optimized Stacking,” Appl. Sci., vol. 13, no. 4, p. 2629, 2023.

[15] D. Smith, S. Khorsandroo, and K. Roy, “Supervised and Unsupervised Learning Techniques Utilizing Malware Datasets,” in 2023 IEEE 2nd International Conference on AI in Cybersecurity (ICAIC), 2023, pp. 1–7.

[16] B. A. Mantoo and S. S. Khurana, “Static, dynamic and intrinsic features based Android malware detection using machine learning,” in Proceedings of ICRIC 2019: Recent Innovations in Computing, 2020, pp. 31–45.

[17] S. Yen, M. Moh, and T.-S. Moh, "Detecting compromised social network accounts using deep learning for behavior and text analyses," Int. J. Cloud Appl. Comput., vol. 11, no. 2, pp. 97–109, 2021.

[18] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A review of android malware detection approaches based on machine learning," IEEE Access, vol. 8, pp. 124579–124607, 2020.

[19] P. Bhat and K. Dutta, "A survey on various threats and current state of security in aanndroid platform," ACM Comput. Surv., vol. 52, no. 1, pp. 1–35, 2019.

[20] T. Sharma and D. Rattan, "Malicious application detection in android—a systematic literature review," Comput. Sci. Rev., vol. 40, p. 100373, 2021.

[21] M. A. Azad, F. Riaz, A. Aftab, S. K. J. Rizvi, J. Arshad, and H. F. Atlam, "DEEPSEL: a novel feature selection for early identification of malware in mobile applications," Futur. Gener. Comput. Syst., vol. 129, pp. 54–63, 2022.

[22] V. Kouliaridis and G. Kambourakis, "A comprehensive survey on machine learning techniques for android malware detection," Information, vol. 12, no. 5, p. 185, 2021.

[23] L. Cai, Y. Li, and Z. Xiong, "JOWMDroid: Android malware detection based on feature weighting with joint optimization of weight-mapping and classifier parameters," Comput. \&Secur., vol. 100, p. 102086, 2021.