Research Article

Deep Learning for Malware Detection: Enhancing cybersecurity with Advanced Classification Models

Ali Hussain[1*], Muhammad Hamza[2], Muhammad-Haseeb-Zia[3]

[1]Faculty of Computer Science and IT, The University of Lahore, Pakistan

[2] College of Science and Technology

[3]Department of Software Engineering,Lahore Garrison University Lahore Pakistan

[*]Corresponding Author: Ali Hussain. Email: alihussain.ramy@gmail.com

Received: August 12, 2024 **Accepted:** November 17, 2024 **Published:** November 18, 2024

Abstract: The rapid propagation of malware presents a important test to global cybersecurity, demanding cultured detection methods talented of growing with initial threats. Traditional signature based methods are stressed to keep step with increasingly advanced malware alternatives, nessitating innvoative solutions that can classify both known and unknown therats with high accuarcy. This study discovers the use of deep learning procedures for enhancing malware detection capabilites, paying preprocessing approaches such as normalization and Term Frequency-Inverse Document Frequency (TF-IDF) for active feature engineering specifically directing on API calls as serious pointers of an submission's behaviour. A Long Short Term Memory (LSTM) network is used to model the time-based dependencies inhert in malware behaviors, enabling the identification of strange sequences telling of malicious movement. By integrating advanced preprocessing, feature engineering, and deep learning, the proposed systems enhances detection accuracy, reduces false positive, and improves flexibility against complication technique used by cybercriminals. The findings suggest that incorporating LSTM networks, combined with effective feature engineering, significatly boosts the capability of malware detection systems conducive to a robust protection against developing cyber threats and a safer digital environment.

***Keywords*:** Long short Term Memory(LSTM), Term Frequency-Inverse Document Frequency(TF-IDF), Normalization

1. Introduction

Malware is becoming a widespread and increasingly complex cybersecurity threat that affects people, businesses, and governments everywhere. Malicious actors are always creating new and more sophisticated methods to take advantage of weaknesses in digital systems due to the quick development of technology and internet access [14]. The absolute number and diversity of new threats frequently outstrips the capabilities of traditional malware detection systems, which mostly depend on signature-based techniques. Although signature-based methods are good at detecting known threats, they have trouble detecting new malware and zero-day assaults since they don't have any pre-existing signatures.

The use of machine learning and deep learning techniques has become a practical strategy to improve malware detection skills in order to overcome these constraints. Deep learning models may automatically extract important characteristics and identify patterns that would not be visible using conventional techniques, especially those built to handle complicated and high-dimensional data. Deep learning provides a proactive protection against changing cyber-threats by utilizing massive datasets and strong computational tools to detect malware, both known and undiscovered. Based on API call sequences, this paper explores the usage of an LSTM network for malware detection. An essential component of a program's performance are API calls, which reveal information on how the application communicates with the system and other resources.

By analyzing API call classifications, LSTMs can identify distrustful patterns telling of malicious activity. Preprocessing systems such as normalization and TF-IDF are active to prepare the dataset for actual feature extraction and model training. The grouping of these techniques aims to size a hearty, flexible malware detection system that improves the accuracy and reliability of identifying cyber threats. This paper is structured as follows: Section 2 discusses related work in the field of malware detection using deep learning. Section 3 presents the methodology, including data preprocessing, feature engineering, and model architecture. Section 4 provides the experimental setup and results, followed by a discussion in Section 5. Finally, Section 6 concludes the study and suggests potential directions for future research.

2. Literature Review

Significant research into sophisticated detection techniques beyond conventional signature-based approaches has been encouraged by the increase in malware complexity and frequency. Deep learning and machine learning techniques have become feasible substitutes for efficient malware detection. By examining file properties, early study, demonstrated the promise of machine learning classifiers for malware identification. More complicated models that can handle big datasets and complicated feature sets were made possible by this groundbreaking study.

With the mounting superiority of malware, researchers initiated combining deep learning to advance detection abilities. Convolutional Neural Networks (CNNs) to categorize malware by changing binary files into grayscale imaginings, successfully leveraging image organization methods for cybersecurity. This advanced style confirmed the ability of CNNs to identify malware patterns that are not easily noticeable through old feature extraction.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have also been widely accepted for malware detection, mostly for taking sequential addictions in API call sequences. The LSTMs to classify interactive shapes in malware, importance the asset of RNNs in model serial data. Their work established that LSTMs are mainly real in individual malicious actions from benign ones, based on the direction and regularity of API calls. Similarly, the researcher dynamic analysis using deep learning to enhance the detection of zero-day malware, indicating that hybrid approaches can provide a more comprehensive solution [16].

In addition to LSTMs and CNNs, transformer-based architectures have recently gained attention for their ability to capture long-range dependencies in data. The technique has been adapted to malware detection to improve the model's focus on critical parts of input sequences. These attention-based models have proven to be highly effective in identifying subtle indicators of malicious behavior, particularly in complex datasets with mixed feature types [17].

The literature also emphasizes the importance of feature engineering in malware detection. The method highlighted the role of automated feature extraction through deep learning, eliminating the need for extensive manual feature engineering and allowing models to learn directly from raw data. More recent studies, such as Term Frequency-Inverse Document Frequency (TF-IDF) to represent API calls and other text-based features, which enhances the model's ability to differentiate between benign and malicious applications.

Overall, the literature indicates a clear shift towards leveraging deep learning for more robust malware detection systems. By integrating models like CNNs, LSTMs, and transformers, researchers have significantly improved detection accuracy and reduced false positives. However, challenges remain, particularly in dealing with adversarial attacks aimed at deceiving deep learning models. The ongoing research into adversarial defense mechanisms and the integration of hybrid models suggests a promising future for deep learning-based cybersecurity solutions.

The Following authors have described the different frameworks

- The paper describes the web framework to detect the malware from Android Devices [1].
- The author describes the author describe the automated malware scanning solution using ML algorithms [2].
- The author [3] describes the memory and signal related feature contribute to more precise classification.
- The paper [4] describes the genetic algorithm-based feature selection android malware detection

- The author uses a deep learning model which outperform in several DL methods [5].
- The author presents an idea with cascaded kernelized one-sided perceptron [6].
- The random Forest outperformed in the dataset [7].
- The machine Learning algorithms in conjunction with feature selected from android manifest file [8].
- The paper describes the ML-powered malware detection approaches and technique by organizing them under four-axes [9].
- This paper author proposed a solution of Security attacks in internet of things [10].
- This paper describes the machine learning technique for malware detection [11].
- The paper describes the Data Framework in cloud computing [12].
- The authors presented a machine learning technique [13].

3. Methods

The Drebin dataset, which comprises a mixture of malicious and benign Android applications, serves as the main dataset for training and assessing the deep learning model. The overall methodology aims to create a robust system that can accurately distinguish between benign and malicious applications based on API call sequences. The suggested approach for malware detection using deep learning involves several crucial steps, including data preprocessing, feature engineering, and model architecture design.

Preparing the dataset for deep learning requires a number of steps, one of which is data preprocessing. To bring all of the features in the Drebin dataset to a consistent scale, usually between 0 and 1, it is first normalized. Regularization assurances that feature with many series do not excessively influence the model during training. Moreover, lost values are achieved by either crediting the lost data or disregarding incomplete models to confirm data quality. This stage is vital for dropping noise and attractive the model's volume for knowledge.

Feature Engineering: To quantify the significance of each call throughout the dataset, we employ Term Frequency-Inverse Document Frequency (TF-IDF) to represent API calls in this study. This gives significant or unique calls greater weight in order to distinguish malware. In order to transform the raw data into insightful representations that enhance the model's performance, feature engineering is crucial. By concentrating on the most pertinent features during training, the model is better equipped to recognize harmful activity.

The recommended practice pursues to grow an extremely real malware detection arrangement that can minimalize false positives and regulate to new pressures by combination the next steps: Data preprocessing, feature engineering, model design, hyperparameter alteration, regulation, collaborative knowledge, and argumentative exercise. The scheme's volume to classify and break malware is importantly better by combination LSTM systems with cultured preprocessing and feature engineering methods, which benefits to make an additional protected connected situation.

4. Experimental Setup

The investigational arrangement for measuring the deep learning-based malware discovery system involved numerous steps, concentrating on making the dataset, model training, and evaluation under skillful condition

4.1. Hardware and Software Enivornment

- Computer hardware Formation: For the hearings, a computer with an Intel i9 processor, 64GB of RAM, and an NVIDIA GPU (RTX 3090) to rapidity up the exercise process was used. The similar controls wanted for LSTM perfect exercise were achieved on the GPU.
- Software Tools: Python programming was rummage-sale for the application, and TensorFlow and Keras public library remained secondhand to make and sleeper the deep learning representations. For feature engineering and data preparation, Pandas and Scikit-learn were utilized.

4.2. Dataset Preparation:

- For perfect exercise and calculation, the Drebin dataset—which comprises both malicious and benign Android requests—was chosen. Three subsections were shaped from the dataset: the Training Set (70%) and the Validation Set (15%) and Test Set (15%). While the validation set was used to fine-tune hyperparameters and prevent overfitting, the training set was utilized to train the LSTM model. The test set was retained to assess the model's ultimate performance.
- Normalization and Feature Engineering: Data normalization was carried out to scale features between 0 and TF-IDF was applied to the API call sequences to emphasize important API calls relevant to malware detection.

4.3.Training Parameters:

- Optimizer and Learning Rate: The Adam optimizer was used for training the LSTM model, with an initial learning rate of 0.001, which was adjusted based on the results obtained from the validation set.
- Batch Size and Epochs: A batch size of 128 was used to optimize the training process, and the model was trained for 100 epochs, with early stopping implemented to prevent overfitting if the validation loss stopped improving.

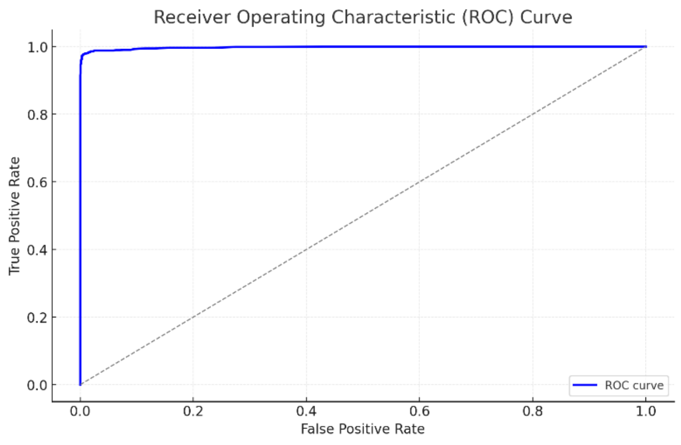

Figure 1. ROC-CURVE

The ROC-Curve describes a True positive rate and False positive rates.

Table 1. Volume of Dataset

Drebin	Samples
Malicious	5560
Benign	9476

The table 1 describes the Volume of Dataset that the Drebin dataset contain 15036 dataset and having malicious samples 5560 and benign samples 9476

5. Discussion

The suggested LSTM-based model successfully uses sequential patterns in application actions to show great promise in identifying Android malware. The model was able to concentrate on patterns that differentiate malware from benign apps by using TF-IDF during feature engineering to highlight important but uncommon features. Furthermore, normalization made guaranteed that every feature contributed consistently during training, avoiding skewed outcomes brought on by different scales. With a high AUC score of roughly 0.97, the model demonstrated its capacity to identify the temporal relationships that are frequently suggestive of malevolent activity. LSTMs are perfect for usage in mobile security applications because of these findings, which show how well they handle sequential data.

However, some challenges remain. LSTM models are computationally intensive to train, which may limit their applicability on low-resource devices. Additionally, more diverse datasets are needed to confirm the model's generalization to unidentified malware families, despite the fact that it performs well on the Drebin dataset. Additional restriction that can keep LSTM predictions from life working in serious security applications is their interpretability. Upcoming trainings must focus on counting explainability policies and investigative cross models that syndicate LSTMs with modernizers or care devices in order to recover presentation and practicality. With lively performances in the dataset, such as system action, may additional growth the flexibility of the model.

6. Conclusions

This study establishes how successfully LSTM neural networks notice Android malware by employing the successive nature of database activities. By joining TF-IDF feature engineering, standardization, and consecutive modeling, the model was talented to find high precision and loose AUC score of 0.97. These conclusions display how LSTMs are authoritative tool in the match against growing cyberthreats on mobile platforms since they are able to recognize complex decorations of destructive motion that motionless models would overlook.

Though the outcomes are hopeful, there are motionless difficulties with statistics representativeness, processing stresses, and perfect interpretability [15]. Resolution these matters is essential for practical request. Upcoming investigate should reflect using mix models and counting lively structures like runtime performances and system doings to added improve discovery presentation. Moreover, engaging logical AI methods can assistance near the opening amid high-performance models and dependable, real-world malware detection systems.

Acknowledgments:

N/A

Data Availability:

The research data related to this work are included within the manuscript. For more information on the data, contact the corresponding authors.

References

1. Mahindru, A.L.; Mahindru, S.A.L. MLDroid—Framework for Android malware detection using machine learning techniques. *Neural Computing and Applications* 2021, *33*, 5183–5240.
2. Tahtaci, B.; Tahtaci, C.B. Android malware detection using machine learning. *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE 2020, pp. 1–6.
3. Wang, X.; Wang, L.C. Android malware detection through machine learning on kernel task structures. *Neurocomputing* 2021, *435*, 126–150.
4. Lee, J.H.; Lee, H.S.; Lee, Y.Y. Android malware detection using machine learning with feature selection based on the genetic algorithm. *Mathematics* 2021, *9*, 2813.
5. Elayan, O.N.; Elayan, M.A.M. Android malware detection using deep learning. *Procedia Computer Science* 2021, *184*, 847–852.
6. Agrawal, R.; Chavan, S.V.; Jadhav, C.S.; Ghotekar, G.; Agrawal, S.N. Android malware detection using machine learning. *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, IEEE 2020, pp. 1–4.
7. Batouche, A.; Batouche, J.H. A comprehensive approach to Android malware detection using machine learning. *Information Security Technologies for Controlling Pandemics* 2021, pp. 171–212
8. Herron, N.G.; Teague, M.J.; Herron, B.R.K. Machine learning-based Android malware detection using manifest permissions. *Proceedings of the 54th Hawaii International Conference on System Sciences* 2021
9. Kouliaridis, V.; Kouliaridis, G. A comprehensive survey on machine learning techniques for Android malware detection. *Information* 2021, *12*(5), 185
10. Irfan, M., Sumra, I. A., Awan, I. A., Mahmood, K., Javed, M. A., Mujahid, M. A., & Akhtar, N. (2022). Security Attacks And Proposed Solutions In Internet Of Things (Iot). Migration Letters, 19(6), 1016-1027.
11. Ijaz, A., Khan, A. A., Arslan, M., Tanzil, A., Javed, A., Khalid, M. A. U., & Khan, S. (2024). Innovative Machine Learning Techniques for Malware Detection. Journal of Computing & Biomedical Informatics, 7(01), 403-424.
12. Awan, I. A., Sumra, I. A., Mahmood, K., Akram, M., Mujahid, S. K., & Zaman, M. I. (2024). A Reliable Approach For Data Security Framework In Cloud Computing Network. Migration Letters, 21(S11), 923-934.
13. Ibraheem, I., Ramay, S. A., Abbas, T., ul Hassan, R., & Khan, S. (2024). Identification of Skin Cancer Using Machine Learning. Journal of Computing & Biomedical Informatics, 7(02).
14. Ullah, A., Waqar, M., Nazir, S. S., Adnan, A., Khan, M. A., Raffat, M. W., & Rafi, S. (2024). The impact of information communication technology and financial innovation on the financial performance of Chinese commercial banks. Remittances Review, 9(S2 (May 2024)), 364-383.
15. Chishti, M. F., Rao, M., Raffat, M. W., & Rafi, S. (2024). Estimating Corporate Risk and Corporate Value An Application Of Altman's Z-Score On The Kse-30 Index. International Journal of Contemporary Issues in Social Sciences, 3(2), 2833-2841.
16. Ahmad, R., Salahuddin, H., Rehman, A. U., Rehman, A., Shafiq, M. U., Tahir, M. A., & Afzal, M. S. (2024). Enhancing database security through AI-based intrusion detection system. Journal of Computing & Biomedical Informatics, 7(02).
17. Khan, R., Iltaf, N., Shafiq, M. U., & Rehman, F. U. (2023, December). Metadata based Cross-Domain Recommender Framework using Neighborhood Mapping. In 2023 International Conference on Sustainable Technology and Engineering (i-COSTE) (pp. 1-8). IEEE.