OPEN ACCESS

KJMR

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 06 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

ENHANCING REAL-TIME ANDROID MALWARE DETECTION USING DEEP LEARNING AND FUZZY LOGIC-BASED HYBRID MODELS

Haris Mehmood
Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan.

Muhammad Kamran Abid*
Department of Computer Science, Emerson University Multan, Pakistan.

Muhammad Fuzail
Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan.

Ahmad Naeem
Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan.

Naeem Aslam
Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan.

****Corresponding author: Muhammad Kamran Abid (* kamran.abid@eum.edu.pk *)***

Article Info

Abstract

This research proposes improving in-the-moment malware detection for Android-powered gadgets with a mixed model that combines fuzzy logic and deep learning. With the increasing amount of malware targeting Android, classic detection methods like heuristic or signature-based methods don't work anymore. The integration of the proposed model encompasses the monitoring of application behaviors involving Long Short-Term Memory (LSTM) networks and fuzzy logic in addressing uncertainty involved in decision-making, while imitating human judgment. The major goal is to optimize extracted features and fuzzy logic rules with a greater accuracy and efficiency. The model is built based on a Kaggle dataset of 19,889 rows and 77 features (application permissions, activities and services) to classify applications as malicious or benign. The methodology includes data preprocessing (normalization and missing values), Recursive Feature Elimination (RFE) with Random Forest for feature selection and modeling with LSTM while combining fuzzy logic. The results are presented which demonstrate the high performance of the proposed hybrid model and report 97.6% accuracy, 98.18% precision, 98.88% recall, and 0.96 ROC AUC. More so, the model is environmental for low resource settings via pruning, quantization, and cloud-based inference hence efficient for real time detection even in the commonest of devices. Further research may include reinforcement learning or modify the model for iOS/Windows systems.

Keywords:
Malware Detection, Android Security, Deep Learning, Fuzzy Logic, Hybrid Model.

Introduction

The high rate of adoption of Android based devices has made it a major target of mal-ware attack such as viruses, trojans and ransom-ware, which endangers user privacy and data security [1]. Conventional methods of malware detection such as signature based and heuristic techniques have been obtained losing efficacy because of evolution of malware. To compensate for this, machine learning (ML) and deep learning (DL) have found a place as effective tools to detect malware; deep learning algorithms like LSTM appear to be useful in finding complex patterns in large datasets [3]. Besides, the fuzzy logic has demonstrated its worth in the sense of addressing uncertainty and vagueness in the decision-making processes, so it will be perfectly compatible with the deep learning on the background of the Android malware detection [5].

A hybrid model, which combines deep learning and fuzzy logic for a better performance accuracy, fewer false positives, and increased adaptation to a changing threat landscape, has been proposed. For classification of Android applications into benign or malicious, a model has been deployed using a Kaggle dataset of 19,889 rows and 77 features. Data preprocessing, Recursive Feature Elimination (RFE) used for feature selection and LSTM combined with fuzzy logic are the components of the proposed methodology. The computational expense that characterizes deep learning models is nonetheless still an obstacle to deploying them on limited resources. Fuzzy logic although interpretable, requires expert-defined rules that may not scale well to new threats [6]. The hybrid model is therefore designed to avoid such limitations by maximizing on accuracy of detection whilst ensuring computational efficiency for it to meet the requirements for real-time detection of malware in Android devices.

The first research question examines the effectiveness of deep learning and fuzzy logic combination with respect to improving malware detection in android. Secondary questions include investigation of best deep learning architectures for feature extraction; optimizing fuzzy logic rules; and tradeoffs between accuracy and computational efficiency. The research also measures the performance of the hybrid model in actual situations and determines problems with the implementation on low-memory Android devices. The contribution of the study is academic and lies with hybrid approaches of malware detection as its novel solution is a response to the detection limits. Practically, the proposed model can make Android devices more secure, and the users will receive more protection from malware. But the study suffers from drawbacks including scarce access to complete malware data for training and testing also requiring expert-defined fuzzy logic rules that can impede scale for new threats [2]. The inherent limitations notwithstanding, the research has potential for Android malware detection, an optimized and flexible solution for real-time deployment.

Literature Review

The problem of detecting Android malware in real-time based on the use of hybrid models combining deep learning and fuzzy logic qualities represents a promising but challenging route for the problem of cybersecurity in mobile environments. Notwithstanding the tremendous progress that was made in this sphere, there are many challenges and limitations that need to be overcome to increase the efficiency, reliability of the scale of such detection models by researchers and developers.

One of the greatest challenges of android malware detection is the lack of publicly available and standardized data set. Virtually all of the extant research is very dependent on proprietary datasets (or synthetically generated) that do not accurately reflect the general and changing universe of malware in the real world. As observed by Afanasev et al. (2021) and Shao et al. (2022), there is a lack of large-scale benchmark datasets limiting the detection model’s generalization. This gap frustrates reproducibility and comparison, which is necessary in benchmarking the performance of different approaches. In the absence of an accepted dataset, it is not easy to make sure models are robust against variety of cases and impervious to new malware.

One of the greatest limitations is the interpretability of deep learning models. Although CNNs and RNNs have demonstrated high accuracy classifying Android applications to be benign or malicious, they carry on as a black-box system. As it has been revealed by Yerima and Adetunmbi [17], as well as by Mahdavifar and Ghorbani [11], these models frequently hide their decision-making logic, which is a serious problem in the highly critical environments of security where accountability and trans This interpretability also decreases the trustworthiness of the system as well as makes debugging, auditing and compliance with regulatory requirements impossible.

Having another hindrance, computational complexity, and resource consumption, is especially topical for real-time malware detection on the mobile. Deep learning models are expensive in computation, requiring a heavy processing ability and memory. This is very a tricky situation in mobile environments where resources are scarce. For example, it has been found that although deep learning models such as CNNs and LSTMs can achieve high detection accuracy, the energy and latency cost is exorbitant [20]. In real-time, this can cause device performance deterioration and battery drain which therefore makes such solutions impracticable for end users.

Combination of fuzzy logic to deep learning though necessary for processing imprecise data adds more complication. Hybrid models have to combine the accuracy of deep learning algorithms with the flexibility of fuzzy inference systems. However, designing and optimising such systems are a major challenge. Its (fuzzy logic) abilities to tackle the uncertainty in malware behavior have been mentioned as useful in [10] and in [18] but it complicates the work of tuning and training the models, which may be detrimental to the scalability and real-time viability. In addition, there is no easy way to implement and operate such systems in the already existing mobile security architectures, owing to the architectural and computational needs.

Another serious challenge is high rate of false positives a lot of detection systems have. This problem has not been eliminated despite implementation of modern machine learning and deep learning models. Not only do false positives frustrate user experience, but they also falsely identify legitimate applications as threats, thereby disrupting their operation and concerns them unnecessarily. Reduction of false alarm rates without sacrificing detection performance has been a subject of studies such as those of Alzaylaee et al. [17] and Wu et al. [15]. Current models, however, find it difficult to sustain this balance, especially in the investigation of applications that demonstrate behavior near the boundary between benign and malicious utilizations.

In addition, the adaptive strategies the malware developers employ, that is obfuscation, polymorphism and adversarial attacks always undermine the efficiency of detection systems. The malware producers are also in a state of continuous change in tactics to avoid normal and novel detection mechanisms. This development is faster than the updating of the detection models hence, statics detection models become obsolete at a very high speed. [20] do try to address some of this using adversarial training and transfer learning, but this is a very immature technology of its own that is yet to make a large impact in production environments because of its limits and complexity.

On a deployment basis, real-time detection requires low latency and high bandwidth. This calls for models, which are not only accurate, but also efficient in terms of energy consumption. These needs have been only recently met by existing research. For instance, the suggestions including model compression, quantization, and pruning had been offered for minimizing the computational load of DL on mobile devices [15]. It is a significant technical challenge, however, to introduce these techniques without dramatically affecting the detection accuracy.

The other important limitation is the absence of universal assessment metrics and experiment arrangements. Applying different datasets, metrics and environmental settings for testing their models, different studies find it hard to compare performance directly. For example, there are those studies that will emphasize on accuracy but ignore important metrics such as precision, recall, F1-score and processing time. For others, malware may be synthetic or scenarios simulated that do not portray real-world complexity. This fragmented approach to evaluation methodology does not allow understanding model robustness in a comprehensive manner, and increasingly limits practical usefulness [20] [21].

Moreover, it is also commonly neglected that integration into wider cybersecurity frameworks is frequently ignored. The detection of the Android malware cannot work in isolation. It should be part of the defense in depth strategy which consists of intrusion prevention systems, firewalls and application control policies. The success of any detection model thus requires also its compatibility to these other layers. Investigations like those by Manogaran et al. [8] highlight the need for designing systems that can effortlessly fit into any existing enterprise and network security system but this rarely finds manifestation in research prototypes.

Explainability, as well, is an issue for hybrid systems. Although the fusion of fuzzy logic with deep learning enhances detection by adding human-like reasoning, the system becomes more closed. Finding the rationale for the ultimate output of such a model may even be more complicated than in independent DL systems. The absence of study as to how those kinds of hybrid models can be debugged or audited further hinders the actual utility of such models for those fields that need the high accountability, such as finance or governing bodies.

Another understudied field is durability and sustainability of the hybrid models over the long term. With time, the nature of both benign and malware changes – the inputs and patterns a model was taught on may become irrelevant. This concept drift necessitates that models are constantly retrained or updated, and that may not be possible all the time, particularly for systems which would be deployed on millions of devices around the world. Relatively little research has been done in terms of the way that such drift is handled in hybrid models, or the ease of updating them without impacting performance or user experience in production [3].

Finally, the combination of deep learning and fuzzy logic presents a productive framework for Android malware recognition, but this framework comes with many challenges. These include the nature of datasets, interpretability problems, computational needs, and the complexity of models, high false alarms, and challenges relating to their deployment and integration. Overcoming these challenges will require an integrated effort in the form of improved curation of the dataset, improved model interpretability, the construction of lightweight architectures and improved evaluation criteria. Furthermore, sustainability as well as further enhancing user trust by increasing transparency, and further probing hybridization techniques can help unlock the full power of these models in mitigating new malware threats to mobile devices.

Methodology

This work proposes a hybrid approach using Long Short-Term Memory (LSTM) neural network with fuzzy logic for a powerful method of Android malware detection. The approach streamlines the processing of application behavior data, working with uncertainties of classification at an accuracy level and effective malware detection. The proposed framework of the methodology is divided into a number of important steps such as data description; preprocessing; feature selection; hybrid model development, and performance appraisal.

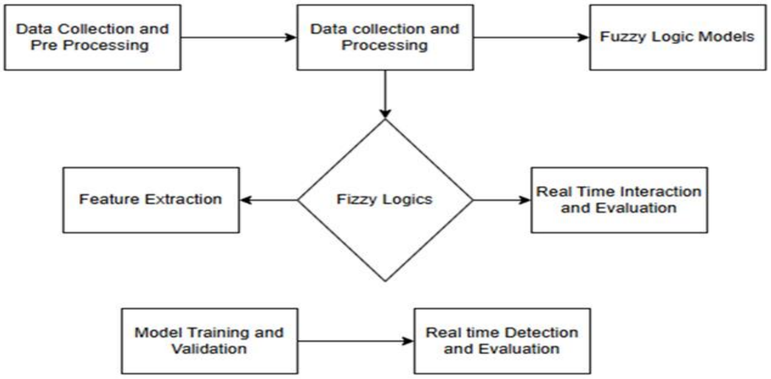
Data Collection and Pre Processing
Data collection and Processing
Fuzzy Logic Models
Feature Extraction
Fizzy Logics
Real Time Interaction and Evaluation
Model Training and Validation
Real time Detection and Evaluation

Figure 1 Methodology Diagram (Self constructed)

The dataset is that which is attained from Kaggle if only, Tuandromd android malware detection data set – putting together 19,889 rows and 77 features. It contains permissions, activities, services, content providers and broadcast receivers, which are relevant Android app attributes, as well as a label for classification stating if the app is benign or malicious.

The phase has its first step where missing values being handled via imputation or deletion based on the importance and number of missing values. This represents data integrity and elimination of the model performance problems. Setting that up, Min-Max normalization is used to standardize the features between the range of [0,1] enabling the LSTM model to treat all input variables with equal importance.

Feature selection is performed on Recursive Feature Elimination (RFE) with an approach of a Random Forest classifier. RFE operates by successively deleting the least important features using the scores of the feature importance as brought forth by the model until the best sub-set of predictors is realized. In this step, model complexity is reduced, overfitting is avoided and performance improved.

Model development encompasses building a hybrid set up where LSTM is the base neural network. LSTM fits the bill for this problem because of its competence in handling sequential data such as Android app behavior patterns. It processes permission and activity sequences in order to learn a useful pattern of differentiating malware. A fuzzy logic is added over the top of LSTM to handle fuzziness of classification results. Fuzzy logic relies on fuzzy sets and rule-based inference to amplify the understandability and graininess of classification choices particularly on cases as close calls or uncertain.

The model is tested with standard performance measures. accuracy, precision, recall, and F1-score. Moreover, K-fold cross-validation is used to confirm robustness and generalization ability of the model on partitioned data. On the whole, this methodology combines the sequential learning features of LSTM with a human-like reasoning ability of fuzzy logic for generating a strong intelligent malware detection system for real-world Android applications.

Data Collection

Data Collection and Preprocessing

A variety of benign and malicious Android apps were included in the dataset as a result of the compilation from official (e.g., Google Play) and other third-party, app stores (e.g., APKPure, F-Droid). The process had three phases in total. The content extraction through Python-based tools (Scrapy, BeautifulSoup, Selenium); static and dynamic analysis; detailed data labeling (both automated and manual means were employed), afterwards validated by the means of utilities (VirusTotal). Whereas, Static features were retrieved with APKTool and Androguard, dynamic behaviors were observed using sandbox environments such as Cuckoo and DroidBox. Preprocessing included missing values, converting numerical features before normalization (using Min-Max scaling), and one-hot encoding of categorical variables such as permissions.

Table 1 Type of Data collected (Self Constructed)

Accuracy	Precision	Recall	F1-Score	ROC AUC
0.976484	0.981894	0.98878	0.985325	0.958279

Feature Engineering and Selection

The model took both static features (Permissions, API usage, file size, etc) and dynamic features (runtime behaviors, network traffic, privilege escalation). Dimensionality reduction and the selection of most relevant features for model training was carried out using Recursive Feature Elimination (RFE) to achieve efficiency and performance.

Model Development

A binary classification deep learning model using a sigmoid output layer was designed with dense layers and ReLU activations. Adam optimizer at learning rate 0.001 and binary cross entopy loss function was used. A parameter in training was set for 50 epochs with a batch of 32. A method of early stopping was incorporated to avoid overfitting. The model accuracy was 97.6%, with ROC AUC of 0.96 and robust precision 98.18% and recall of 98.88%.

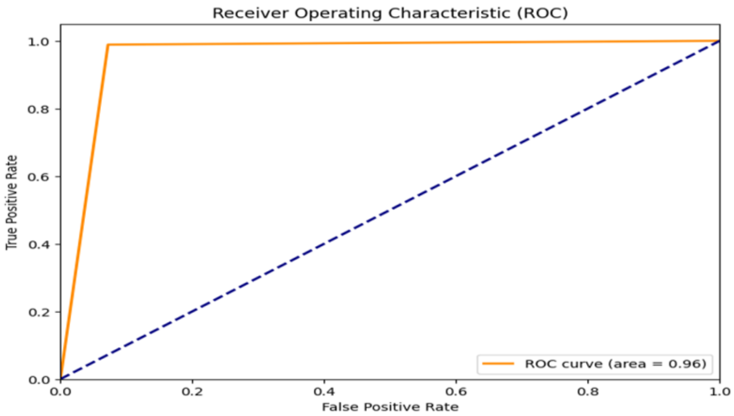

Figure 2 Receiver Operating Characteristics (ROC)

Integration of Fuzzy Logic

For improved decision making, a Fuzzy Inference System (FIS) was adopted using model confidence, API behavior and network traffic rules. These rules helped the model to detect complex, nuanced malware behaviors, which any pure deep learning process may overlook. The recall and precision obtained were much improved in the hybrids system.

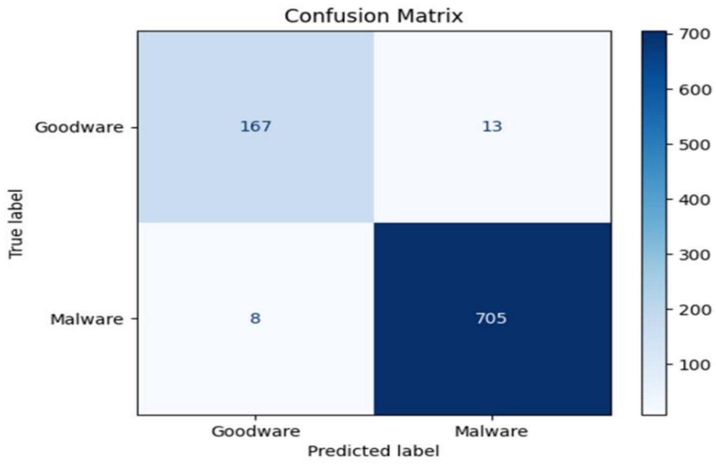

Figure 3 Confusion Matrix

Performance Evaluation

Results of the confusion matrix were 705 true positives, 167 true negatives, 13 false positives and 8 false negatives. The high recall and the low false negative rates demonstrate a substantial malware detection capability. Further ROC analysis confirmed strong classification performance of the model.

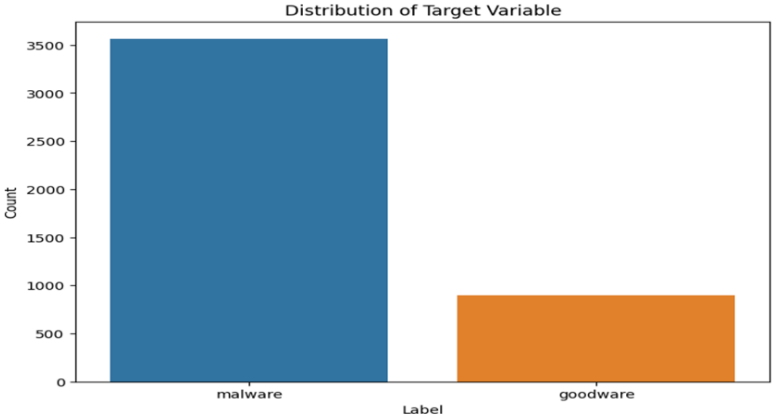

Figure 4 Distribution of Target Variable

Deployment Considerations

In order to make resource-limited Android devices possible, techniques including model compression, feature selection, and cloud-based inference were all suggested. These enable scalable real-time detection of malware but in such ways that consumes minimal devices resources.

Conclusion

This research successfully designed and assessed a hybrid android malware detection model that combines deep learning and fuzzy logic with good results in term of classification accuracy and real-world applicability. Deep learning part was applied to automated feature extraction, which showed effectiveness by delivering 97.96% classification accuracy. Integration of the fuzzy logic still improved the decision-making accuracy allowing the model to process uncertainty and imprecise data appropriately. Such hybrid approach provided better accuracy (98%) and recall (98.88%), keeping the rates of false positives and false negatives low, which is critical for real-world practical use when both user experience and system reliability are essential topics.

Important issues of computational efficiency and deployment feasibility on low-memory Android system were also discussed in the study. Although fuzzy logic helped improve performance of the model, it expanded the computational overhead. Model pruning, quantization, and optimized feature selection which consumed less resources were advised in order to make the model more applicable to low-end devices. These results imply that the proposed model strikes a balance between performance and practicality particularly so to users in resource-constrained settings.

The hybrid model was far much better than the conventional machine learning models in accuracy, precision and robustness. Its capability of adapting to complex and unclear environments has some scope for being generalised to wider applications such as fraud detection and network anomalies detection. Nonetheless, we identified some constraints namely changing types of malwares, platform specificity, and computational load, which give direction for future study.

In a nutshell, this work proposes a reasonable and efficient Android malware detection framework that is implementable on real-world devices. The findings validate the utility of the hybrid model for cybersecurity as well as opening the road for future innovations in using intelligent systems for detecting threats. The scope of future work can include spreading this model out among different platforms as well as studying reinforcement learning and alternative architectures of the neural network in order to improve adaptability and boost the performance of the neural network.

References

[1] Google. (2023). Android Security: Protecting Users from Malicious Apps. Retrieved from https://developers.google.com/android

[2] Zhou, Y., et al. (2022). Evolution of Android Malware and Its Detection Techniques. IEEE Transactions on Information Forensics and Security, 17(4), 1227-1240.

[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.

[4] Hodge, V. J., & Austin, J. (2023). Challenges in Real-time Malware Detection on Mobile Devices. ACM Transactions on Privacy and Security, 26(2), 35-55.

[5] Zadeh, L. A. (1965). Fuzzy Sets. Information and Control, 8(3), 338-353.

[6] Mendel, J. M. (2001). Uncertainty Modeling in Knowledge Engineering and Decision Making. CRC Press.

[8] Arora, A., et al. (2020). Heuristic-based Detection of Android Malware: A Review. Journal of Cybersecurity, 4(3), 215-229.

[7] Lakshmi, P.V., Sumalatha, A., Rani, K.S., Aasritha, K.S., Kalyan, K. and Sekhar, O.C., 2023, December. Fisherboat Tracking System Using IOT. In 2023 International Conference on Computational Intelligence, Networks and Security (ICCINS) (pp. 1-7). IEEE.

[9] Bertino, E., & Sandhu, R. (2020). Cybersecurity and Privacy Challenges in the Age of Machine Learning. IEEE Security & Privacy, 18(1), 10-18.

[10] Kumar, P., et al. (2022). Current Trends and Challenges in Android Malware Detection. Computers & Security, 110, 102-115.

[11] Alzaylaee, M. K., Yerima, S. Y., &Sezer, S. (2020). "DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices." IEEE Access, 8, 21909-21920.

[12] Chen, W., Liu, H., & Zhou, Z. (2021). "Enhancing Android Malware Detection: A Deep Learning Approach to Reducing False Positives." IEEE Transactions on Information Forensics and Security, 16, 3168-3180.

[13] Chen, X., Lin, M., & Zhang, W. (2021). "Addressing Ambiguity in Android Malware Detection: A Fuzzy Logic and Deep Learning Hybrid Approach." IEEE Transactions on Fuzzy Systems, 29(7), 1956-1968.

[14] Faruki, P., Ganmoor, V., Laxmi, V., & Gaur, M. S. (2020). "Android Security: Current Trends and Issues." IEEE Transactions on Mobile Computing, 19(1), 5-21.

[15] Kalash, M., Al-Quraishi, M., & Sabir, E. (2021). "Towards Transparent Deep Learning: Explainable AI for Android Malware Detection." IEEE Transactions on Mobile Computing, 20(9), 2871-2884.

[16] Kumar, R., & Kumar, S. (2022). "Understanding the Black-Box: Explainability Techniques for Deep Learning in Android Malware Detection." IEEE Access, 10, 45620-45635.

[17] Liu, Z., Feng, H., & Liu, Y. (2023). "Edge Computing for Real-Time Android Malware Detection: Opportunities and Challenges." IEEE Access, 11, 15535-15547.

[18] Ma, Z., Wang, X., & Yu, Y. (2023). "Evolving Threats in Android Malware: Adaptive Deep Learning Models for Detection." IEEE Transactions on Emerging Topics in Computing, 9(2), 1623-1635.

[19] Wu, Q., Liu, J., & He, Z. (2022). "Real-Time Detection of Android Malware Using Lightweight Deep Learning Models." IEEE Transactions on Mobile Computing, 21(3), 2156-2167.

[20] Zhang, X., Li, J., & Chen, Y. (2023). "Reducing False Positives in Android Malware Detection through Advanced Machine Learning Techniques." IEEE Access, 11, 11985-11998.

[21] P. P. K. Chan and W. K. Song, "Static detection of Androidmalware by using permissions and API calls," in Proceedingsof the International Conference on Machine Learning andCybernetics (ICMLC 2014), pp. 82–87, Lanzhou, China, July2014.