Protecting Androids from Malware Menace Using Machine Learning And Deep Learning

C Siva Kumar[1*], S Mohan Krishna[2], V Ebinazer[3],N Narasimha Naidu[4], P Pawan Kalyan[5]

[1]Professor, Department of DS, Mohan Babu University(Erstwhile Sree Vidyanikethan Engineering College), Tirupati, India
[2,3,4,5] UG Scholar, Department of Computer Science and Systems Engineering, Sree Vidyanikethan Engineering College, Tirupati, India

[*1]Svkumar650@gmail.com
[2]mohankrishna0882@gmail.com
[3]ebinezarv@gmail.com
[4]narasimhanettem123@gmail.com
[5]pawankalyan2029@gmail.com

Abstract—Mobile devices have become integral to our lives. Among operating systems, Android holds the largest market share, making it a prime target for attackers. While various solutions exist for Android malware detection, there remains a need for effective attribute selection methods. In this work, we introduce an Android malware detection technique that employs machine learning to distinguish between safe and dangerous applications. By reducing the feature vector dimension, training time decreases, and real-time malware detection becomes feasible. A number of multiple linear regression techniques are evaluated, including support vector machines, decision trees, Naïve Bayes, and k-nearest neighbors. Additionally, we employ the stacking classifier method an ensemble learning technique to enhance classification performance. This stacking classifier combines Random Forest and Sequential Neural Networks and performs testing. Our results surprisingly show good performance with linear regression models, eliminating the need for excessively complicated techniques.

Keywords: Ensemble Learning, SNN, Random Forest, Stacking Classifier, Permissions.

1 Introduction

In an era dominated by mobile technology, Android devices have become ubiquitous companions in our daily lives. These pocket-sized marvels empower us with a multitude of functionalities, from communication and entertainment to financial

K. R. Madhavi et al. (eds.), *Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024)*, Advances in Computer Science Research 112,
https://doi.org/10.2991/978-94-6463-471-6_28

transactions and location tracking. Unfortunately, malevolent actors who take advantage of weaknesses for personal benefit are also drawn to its widespread adoption.

Malware, a persistent threat in the digital landscape, poses significant risks to Android users. Cybercriminals continually churn out new variants, targeting unsuspecting victims through seemingly innocuous apps. As the popularity of Android devices soars, safeguarding system integrity and user privacy becomes paramount. This research paper delves into the realm of Android malware detection. Our goal is to develop an effective framework capable of identifying malicious apps and thwarting potential threats. We focus on dynamic analysis, a technique that scrutinizes app behavior during runtime. By extracting critical features through static analysis, we construct a robust model using machine learning algorithms. The voyage entails investigating several linear regression methods. Additionally, we embrace the power of ensemble learning by employing the stacking classifier method. Through rigorous experimentation on a vast dataset of over 500,000 Android apps, we unveil promising results.

A hybrid model that combines various machine learning techniques, including anomaly detection and feature engineering, can effectively detect malware by analyzing code patterns and behavioral anomalies. This approach enhances detection accuracy while providing unique insights into potential threats, By examining code structures and behavior, this model can accurately identify malicious activities. Through careful integration of algorithms and feature selection techniques, it enhances detection accuracy while providing unique insights into potential threats.

2 Literature Review

To demonstrate the importance of this work, a survey was done on different research papers to understand various techniques. A few of them are discussed here.

In this work, privacy and feature extraction were prioritized while using semi-supervised learning to detect Android malware.Their approach incorporated federation learning to improve detection algorithms, specifically targeting Android apps [1].This paper focused on deep learning and computational modeling for static malware detection in Android apps, employing tree-based machine learning methods for classification [2]
This paper proposed a method for Android malware detection combining machine learning and Siamese shot learning techniques, emphasizing security and efficiency with deep learning and TensorFlow implementation [3].This paper focused on deep learning and data mining techniques for Android malware detection, specifically targeting abnormal usage of sensitive data within applications[4].This research underscored the significance of security and static analysis in mobile communication by emphasizing the use of deep learning for feature extraction and malware detection in Android applications [5].This article focused on deep learning, parallel processing, and security on Android platforms by proposing a dynamic malware detection method using

temporal convolutional networks [6].This study prioritized performance assessment and technical innovation while focusing on machine learning and ensemble learning approaches for malware identification in Android devices[7].This study used machine learning and natural language processing methods—more especially, support vector machines and deep learning—to identify malware vulnerabilities on Android systems [8],This article focused on deep learning, generative adversarial networks, and security in Internet of Things settings, proposing a solution for[9] IoT-based Android malware detection using graph neural networks with adversarial defense

3 Methodology

The suggested system comprises two primary stages. The initial step is to investigate various machine-learning and deep-learning methodologies. The second stage is to create an ensemble learning stacking classifier that utilizes multiple regression techniques to improve malware detection accuracy

FIRST STEP:
INPUT: A dataset containing labeled android permissions, categorized as either benign or malware. The data set consists of 30000 rows and 215 attributes.
OUTPUT: Models from machine learning as well as deep learning that are specifically designed to identify malware on Android

```
STEPS:
  Initialize Models:
    - Here Models are KNN, NB, RF, DT, SNN, DNN.
    - Load the pre-trained models, leveraging prior training on the malware dataset.
  Fine-Tuning:
    - Fine-tune the entirety of all models on the dataset of android malware.
  Training:
    - Train the adapted all models using the designated training set.
  Validation:
    - Analyze the model's output using the validation set.
    - Monitor for signs of overfitting and assess its ability to generalize.
  Testing:
    - Assess all model's performance on the testing set, obtaining metrics that object
    tively measure its effectiveness.
  Results and Interpretation:
     -Scrutinize and interpret the accuracy of the trained model.
  -Identify potential areas for refinement or enhancement.
  End Algorithm
```

SECOND STEP:
Ensemble Stacking Classifier:

The process of creating a more reliable and accurate prediction model by combining multiple different models, or "base models," is known as ensemble learning. Using the diversity and complementary qualities of several models, ensemble techniques leverage better performance than a single model.

Why to use ensemble learning?

Hybrid Approach: Combine static and dynamic features. Ensemble models benefit from both perspectives. Achieve better accuracy and robustness.

This approach combines both Sequential Neural Networks and Random Forest. This mode of learning is used for hybrid purpose. This has accuracy of 98.9%.

Data Preparation: Preparing the dataset for modeling is the first stage. This usually entails obtaining the dataset, cleaning it up, creating features, and then dividing it into two sets: one for training and one for testing. This dataset is then separated into sets using the train test split function from the sklearn_model selection module. A random state and a chosen test size (20% in this example) are utilized for repeatability.

Random Forest Classifier: This classifier is trained using the fit strategy on the training set, where it is taught to predict using the input features (X_train) and target labels (Y_train).

Sequential Neural Network: This neural network design is comprised of three fully connected layers: employing a sigmoid activation function, a hidden la-yer with 32 neurons, an output layer with one neuron, and an input layer with 64 neurons. The network is assembled, with accuracy serving as the evaluation measure with the help of Adam optimizer as well as binary cross-entropy loss function,

Prediction and Integration: Predictions are made on the testing set using both the Random Forest classifier (random_forest.predict) and the Sequential Neural Network (sequential_nn.predict). For the neural network predictions, the output is thresholded at 0.5 to obtain binary predictions. These individual predictions are then combined using a simple averaging approach to form the hybrid predictions.

Evaluation: The accuracy of the hybrid model is ascertained by comparing the hybrid predictions with the true labels (Y_test) using the accuracy_score function from the sklearn.metrics module. Finally, the console publishes the accuracy of the hybrid model.

4 Results and Discussions

The accuracy ratings provided for various machine learning algorithms demonstrate how effectively they perform in distinguishing between malicious and benign applications or activities. This area often employs the following techniques: Sequential Neural Networks, K-Nearest Neighbors, Logistic Regression, Decision Trees,, and Support Vector Machines (both linear and with polynomial or radial basis function kernels).

Decision trees are excellent for detecting patterns suggestive of the existence of malware because of their capacity to handle categorical data and nonlinear connections. Similar to this, since they can recognize complex correlations in the data, Support Vector Machines with different kernels—including all types of functions which are linear, polynomial, and radial basis functions—perform well for binary classification tasks like malware detection.

Suitable for binary classification problems like malware detection, Logistic Regression is a simple but effective technique that predicts the likelihood of a sample falling into a certain class depending on its characteristics. The process of training the logistic regression model involves identifying the ideal parameters θ that reduce the cost function. Typically, the logistic regression cost function is defined as follows:

$$J(\delta)=-1/m\sum i=1m[y(i)\log(h\theta(x(i))) + (1-y(i))\log(1-h\theta(x(i)))]$$

In the case of Android malware detection, various elements (such permission requests, API calls, etc.) extracted from the Android app would be represented by X, while Y would be the label indicating whether the application is malicious or benign.

Sequential Neural Networks for capturing sequential patterns in Android malware behavior, such as sequences of API calls or system events. These networks excel at learning from sequential data and detecting subtle patterns indicative of malicious activity.

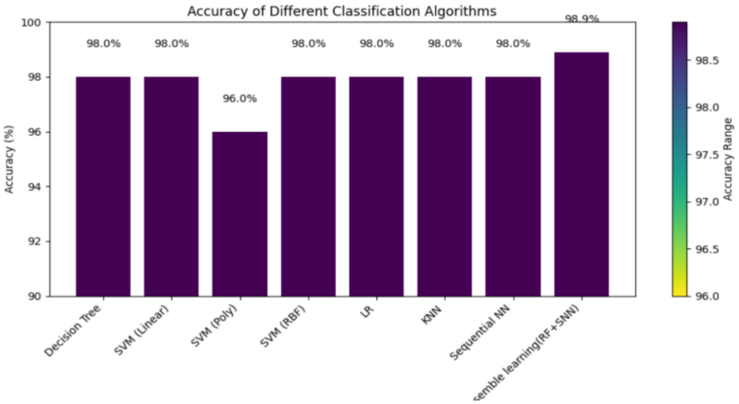

Fig1: Accuracy of all models

Fig1 detailing the comparisons of all models used in this model.

Ensemble learning, represented here as a combination of Random Forest (RF) and Sequential Neural Network (SNN), leverages the strengths of both models to improve overall predictive performance. Random Forest offers interpretability and robustness, while Sequential Neural Networks excel at capturing sequential patterns. The ensemble technique achieved the greatest accuracy of 98.9%, indicating the efficiency of these approaches in detecting Android malware and helping to identify and mitigate possible dangers to users' mobile devices. The observed high accuracies across all algorithms support this claim.

5 Conclusion

The performance of ensemble approaches surpasses that of conventional machine learning algorithms. One of the newest and lightest algorithms is called Stacking Classifier. In order to determine the optimal prediction model for identifying Android malware, we have investigated a number of machine learning algorithms in this work. We have also employed feature selection techniques to lower the many features, maximizing resources and time. Based on our findings, out of all the ensemble approaches, Stacking Classifier gets the highest accuracy of 98.9%. whereas Stacking classifier optimal feature count is 100, with an accuracy of 98.9%. Now, Stacking Classifier may be a better option when it comes to accuracy, but Random Forest provides more reliable results when it comes to efficiency or feature count.

6 References

1. A. Mahindru, S. K. Sharma, "YarowskyDroid: Semi-supervised based Android malware detection using federation learning," 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 380-385, doi: 10.1109/InCACCT57535.2023.10141735.
2. F. A. Almarshad, M. Zakariah, "Detection of Android Malware Using Machine Learning and Siamese Shot Learning Technique for Security," in IEEE Access, vol. 11, pp. 127697-127714, 2023, doi: 10.1109/ACCESS.2023.3331739.
3. S. Guan and W. Li, "EnsembleDroid: A Malware Detection Approach for Android System based on Ensemble Learning," 2022 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, USA, 2022, pp. 1-5, doi: 10.1109/URTC56832.2022.10002213.
4. X. Su, D. Zhang, W. Li, "A Deep Learning Approach to Android Malware Feature Learning and Detection," 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 2016, pp. 244-251, doi: 10.1109/TrustCom.2016.0070.
5. Kumar, Voruganti Naresh, U. Sivaji, Gunipati Kanishka, B. Rupa Devi, A. Suresh, K. Reddy Madhavi, and Syed Thouheed Ahmed. "A FRAMEWORK FOR TWEET CLASSIFICATION AND ANALYSIS ON SOCIAL MEDIA PLATFORM USING FEDERATED LEARNING." Malaysian Journal of Computer Science (2023): 90-98.
6. A.Joomye, Yau, "Dynamic Android Malware Detection Using Temporal Convolutional Networks," 2023 IEEE International Conference on Computing (ICOCO), Langkawi, Malaysia, 2023, pp. 317-322, doi: 10.1109/ICOCO59262.2023.10397844.
7. M. K. Alzaylaee, S. Sezer, “DL-Droid: deep learning based android malware detection using real devices,” Computers & Security, vol. 89, Article ID 101663, 2020.
8. Avanija, J., K. E. Kumar, Ch Usha Kumari, G. Naga Jyothi, K. Srujan Raju, and K. Reddy Madhavi. "Enhancing Network Forensic and Deep Learning Mechanism for Internet of Things Networks." (2023).R. Saidi, N. Essoussi, “Hybrid feature selection method based on the genetic algorithm and Pearson correlation coefficient,” Machine Learning Paradigms: Theory and Application, vol. 801, pp. 3–24, 2018.
9. Faruki, P., Bharmal, A., V., L.. Android Security: A Survey of Issues, Malware Penetration, and Defenses. IEEE Communications Surveys & Tutorials 2015;17(2):998–1022.
10. Rosenberg, I., Shabtai, A., Rokach, L., Elovici, Y.: Generic black-box end-to-end attack against state of the art api call based malware classifiers. In: 21st International Symposium, RAID 2018. Springer (2018).

11. C. C, P. K. Pareek, and D. Gupta, "Improved Domain Generation Algorithm To Detect Cyber-Attack With Deep Learning Techniques," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-8, doi: 10.1109/MysuruCon55714.2022.9972526.
12. Kumar, DNS Ravi, N. Praveen, Hari Hara P. Kumar, Ganganagunta Srinivas, and M. V. Raju. "Acoustic Feedback Noise Cancellation in Hearing Aids Using Adaptive Filter." International Journal of Integrated Engineering 14, no. 7 (2022): 45-55.
13. P. Kumar and A. Mahmood Shakir, "A Robust Long Short-Term Memory Model for Classification of Malware Analysis," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-5, doi: 10.1109/AIKIIE60097.2023.10390338.
14. Madhavi, K. Reddy, K. Suneetha, K. Srujan Raju, Padmavathi Kora, Gudavalli Madhavi, and Suresh Kallam. "Detection of COVID 19 using X-ray Images with Fine-tuned Transfer Learning." Journal of Scientific and Industrial Research (2023): 241-248.
15. Z. Namrud, and C. Talhi, "Deep-Layer Clustering to Identify Permission Usage Patterns of Android App Categories," in IEEE Access, vol. 10, pp. 24240-24254, 2022, doi: 10.1109/ACCESS.2022.3156083.