Explainable Hybrid Neural Model for Robust Android Malware Detection Via API Call Analysis

Nusum Yamuna[1], Neeli Prathibha[2], Indhurthi Meghana[3], DR. R. Madana Mohana[4]
[1,2,3,4]*Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, Telangana, India*

***Abstract*—Research is to use API call analysis to create an explainable hybrid neural model for reliable Android malware detection. By extracting API call sequences—which are crucial markers of malicious activity—the suggested method examines the behavioural patterns of Android applications. Explainable AI techniques are incorporated to provide transparency in the decision-making process, and a neural network-based detection model is used to categories applications as benign or malicious. by emphasising the most significant API characteristics that affect classification outcomes. We use a neural network-based detection model to sort apps into two groups: benign and malicious. We also use explainable AI techniques to make the decision-making process clear. By showing the API features that had the biggest impact on the classification results.**

***Index Terms*—Android Malware Detection, Explainable Artificial Intelligence (XAI), API Call Analysis, Machine Learning, Malware Classification, Android Applications.**

I. INTRODUCTION

The evolution of technology in relation to the use of smart phones and mobile applications has revolutionized communication and technology usage. Among all the mobile operating systems available in the market, the usage of the Android mobile operating system has gained immense popularity. This is because it is an open-source system that allows users to access a number of applications developed by various developers across the globe. However, this has also given rise to malware attacks on this system due to its popularity. Android malware are those applications that are programmed to cause damage to a particular device or those applications that misuse a particular user's information without their knowledge. Such applications are capable of damaging the normal functioning of a particular device. With an increase in the number of Android applications, the problem of controlling damaging applications has become a challenge to the field of mobile security.

Therefore, the problem of Android malware detection has received a great deal of attention. Earlier, the techniques that are used for detecting the malware are mainly signature-based techniques. The application is compared with the signature of the malware present in the database. The techniques are only effective if the pattern of the malware is already present in the database.

The techniques are not effective in detecting newly developed malware or changes in the malicious applications. Therefore, researchers are now focusing on developing more advanced techniques for detecting the malware. Recently, machine learning and deep learning techniques are used to improve the malware detection system. The machine learning and deep learning techniques are using the behavioral pattern of the application to learn from the data and identify the difference between malware applications and clean applications. The intelligent techniques are able to identify the suspicious behavior of the application after learning from the data. During the analysis of the Android application, various features are taken into consideration. The features include the permission required by the application, system calls, and API calls. Among the features, API calls are significant because they define how the application is interacting with the Android operating system. The pattern of API calls provides significant information regarding the behavior of the application. It can thus be said that it is quite clear that although the machine learning model and the neural network model are capable of producing high accuracy in terms of results obtained from the detection process, these two models are still

considered black box models since it is not easy to understand how a particular model has arrived. However, with the aim of resolving this problem, Explainable Artificial Intelligence (XAI) techniques have been introduced.

Explainable AI techniques are used for understanding the way a particular model has arrived at a particular decision and understanding the features affecting the model's prediction in the most significant way.

It can thus be said that the use of the neural network model along with the Explainable AI techniques has provided a new area of research with respect to the detection of Android malware. With the use of Explainable AI model, not only can the malware applications be detected in an effective way, but the results produced by the model can be trusted as well.

II. LITERATURE REVIEW

Several researchers have worked on various methods of detecting Android malware using machine learning, deep learning, and explainable artificial intelligence.

In a study, Liu et al. (2022) worked on the role of Explainable Artificial Intelligence in Android malware detection models. In this study, the researcher focused on understanding the decision-making process of machine learning models while classifying the application as malware or benign. The researcher used various explainability methods to identify the most influencing features while classifying the application as malware or benign. The study revealed that explainable AI helps in enhancing transparency in the model's prediction process.

Scan Savant, in 2024, developed a malware detection framework using a combination of machine learning and explainable AI. The framework analyzed the features and patterns of the applications, thereby enabling the detection of malicious activities in Android applications. Explainability techniques were also used to identify the features that are crucial in the classification process. The developed model ensured high accuracy in malware detection, and at the same time, the model's interpretability was enhanced.

In another study on malware detection in Android applications using explainable deep learning, the authors developed a model using a neural network to classify malware applications. The model extracted the features from the Android applications, and deep learning algorithms were applied in the malware detection process.

To increase the transparency of the model, explainable AI was used to examine the decision-making process. This ensured that the accuracy and reliability of the model were increased in the malware detection process. Additionally, the surveys on machine learning-based Android malware detection have examined different detection methods used in the detection of malware. These include static detection, dynamic detection, and the combination of the two. From the surveys conducted on machine learning-based Android malware detection, the results showed that machine learning algorithms such as SVM, RF, and NN have been effective in the detection of Android malware. In addition, the importance of API call analysis has been highlighted in the detection of malware. The API calls show how the android application interacts with the android operating system when carrying out different operations. It is possible to know the difference between the normal and malicious android applications by looking at the interaction. The malicious android applications use some of the API calls to access the information. Therefore, it is possible to know the malicious android applications and distinguish them from the normal android applications. This approach has been successful in enhancing the precision of the malicious android applications.

This means that there is no clear reason for the application being classified as malware or a benign application. In order to avoid the limitations of this system, it is necessary to incorporate the idea of using the XAI model in association with the Neural Network model and API call analysis for the detection of malware for the Android operating system. Further, researchers have also presented their findings on the effectiveness of deep learning models in detecting Android malware by analyzing behavioral patterns of Android applications. The researchers have presented their models that use neural network models to automatically learn complex patterns from features of Android applications. These models are capable of detecting unknown Android malware and improve detection accuracy for Android malware detection systems. Another significant contribution in this field of Android malware detection is the use of API call sequence analysis for Android malware detection systems. Several researchers have highlighted that API

calls provide useful insights into the runtime behavior of Android applications. By analyzing the sequence of API calls, it is possible to detect suspicious behaviors of Android applications, which are considered to be malware. Machine learning models are found to be promising in differentiating between benign and malicious applications.

Another significant aspect related to the analysis of social media is the application programming interface, which is used to obtain the required information from the social media platform. The application programming interface is used to obtain the information related to the user's activities, user engagement statistics, and content performance using the APIs offered by the social media platforms such as Facebook, Instagram, and YouTube. In the recent past, various researchers have used the application programming interface to design the analytics dashboards that can be used to analyze the performance of the social media platforms. However, the majority of the application programming interfaces developed can analyze the information from a single platform and not multiple platforms at the same time.

Additionally, the application of the explainable artificial intelligence has gained significant attention in recent times. The application of the explainability methods such as the feature importance analysis and the interpretation methods can be used to obtain the required information related to the factors that influence the malware classification decisions.

Another set of research focuses on the significance of feature engineering and the importance of feature selection methods in improving the performance of malware detection systems. As the number of features is quite high due to the nature of Android apps, it is essential to select the most significant features to build efficient models.

Overall, the above literature clearly proves that the integration of neural network models, API call analysis, and explainable AI can improve the performance of Android malware detection systems. These methods not only improve the accuracy of the malware detection system but also improve the transparency and interpretability of the system, which is essential for building efficient cybersecurity systems.

III. RESEARCH GAP

As per the analysis of the existing literature on Android Malware Detection using Machine Learning and Deep Learning approaches, some of the limitations are as follows:

Most of the existing literature on Android Malware Detection using Machine Learning and Deep Learning approaches are focused on enhancing the detection accuracy of malware, but not much importance is given to the transparency of the detection models. Most of the machine learning models, as well as deep learning models, are considered black-box models, as they do not provide transparency in the detection process, i.e., the models do not clearly explain to the security experts how they are arriving at a particular conclusion.

Although some of the recent literature has included the concept of Explainable Artificial Intelligence, this concept of transparency in the detection model has not been fully utilized, as this concept of transparency is used only after the prediction of the model, not during the prediction process.

Third, most studies have focused on general application features, such as permissions and static code features, in malware detection. Though such features are important, their use may not be effective in the detection process, especially in capturing dynamic patterns in malware behaviors. On the other hand, using API call analysis can offer deeper insights into the behaviors of applications in relation to the Android OS, though this has not been sufficiently integrated with the use of explainable neural networks.

Another area that has been neglected in most studies is the ability of malware detection systems to be effective in the presence of evolving malware, especially in dynamic malware behaviors, which are not seen in traditional malware.

Thus, the need for a reliable malware detection system that combines the use of hybrid neural networks, API call analysis, and explainable AI is crucial in ensuring the accuracy of malware detection, while at the same time offering deeper insights into the behaviors of malware applications.

Ref.	Method/Approach	Main Objective	Multiplatform	Prediction	Limitations
1	Machine learning based android malware detection	Detect malicious android applications using permission features	No	Yes	Limited explainability & feature transparency.
2	Deep learning for android malware analysis	Automatically learn patterns from API calls.	No	Yes	Requires large training data.
3	Static analysis-based malware detection.	Analyze apk files using permissions.	No	Partial	Cannot detect dynamical loaded malware.
4	Hybrid analysis approach	Combine static & dynamic analysis for detection	No	Yes	Complex implementation.
5	XAI based malware detection	Proposed interpretable results using SHAP	Partial	Yes	Explainability technique increase computational.

IV. PROPOSED METHODOLOGY

A. Data Collection

In this section, we collect datasets containing Android application information, including both benign and malicious applications. These datasets enable us to analyze the behavior of the applications. Researchers can also utilize public malware datasets to train and test the detection model.

B. Data Preprocessing

We preprocess the collected dataset prior to analysis. This involves cleaning the dataset and eliminating any unwanted information. Preprocessing the dataset improves the accuracy and efficiency of the detection model.

Within this system, users can link their social media profiles and obtain statistics on user engagements using various platforms such as Facebook, Instagram, YouTube, and GitHub.

C. API Call Extraction

The API call is a medium through which the Android application communicates with its operating system. In this section, we are going to extract the API calls from the Android application and analyze how it behaves. API calls are useful to identify malicious activities in Android applications.

D. Feature Extraction.

In this step, the features are identified based on the data gathered in the previous step. The features identified in this step include the frequency of the API calls, the permissions of the application, the behavioral pattern of the application, and so on. Feature extraction is the process of converting the raw or unstructured data of the application gathered in the previous step into a structured form. The structured form of the application data can be easily processed and analyzed by the machine learning algorithm.

The APIs are used to collect the required information related to user activity, content, and engagement metrics from different platforms, i.e., Facebook, Instagram, YouTube, GitHub, and so on. In this phase, duplicate values are removed, and inconsistent values are corrected. Additionally, some values are normalized to maintain consistency. After this phase, the data is structured to be utilized by the machine learning model.

V. EXPERIMENTAL SETUP

A. Dataset Collection

In this present research work, a dataset with samples of Android applications has been used. The dataset contains both benign and malicious applications. Various features are associated with the use of different types of permissions and API call attributes, which give an idea about the nature of the application. These features are used for further analysis for the purpose of detection. These features are used as input to the machine learning model to classify whether the application is of benign or malicious nature.

The dataset used in this research was collected from various publicly available Android malware datasets. The dataset used in this research consists of labeled samples, which are used in supervised learning, where each sample of the dataset represents an Android application along with its permission and API call information.

B. Data Preprocessing

Before training the model, the following data preprocessing activities were performed:

Data Cleaning –

Inconsistencies in the data, such as missing information in the data set, are removed.

Feature Selection –

Important features, such as permission features and API call features, are chosen to be the input features for the model.

Data Encoding –

Data encoding is performed in numerical form, which is required by the machine learning algorithm. Train Test Split – The data set is split into the training data set and the test data set to assess the model's performance. Typically, 80% and 20% are allocated for the training and test data sets, respectively.

C. Model Implementation

In this study, a deep learning-based model, specifically a neural model, is implemented in the detection of Android malware. The model is based on the characteristics of Android applications, such as API and permissions, to classify Android applications as benign or malware. The model is implemented using the Python programming language and some of the most commonly used libraries in machine learning, such as:

NumPy for numerical calculations

Pandas for data handling and preprocessing Scikit-learn for evaluation metrics and utilities SHAP for Explainable AI

The model is trained using the training dataset, and then it is tested using the testing dataset.

D. Evaluation Metrics

While assessing the performance of the proposed model, various evaluation metrics were taken into consideration: Accuracy – Assesses the overall accuracy of the model.

Precision –

Assesses the number of malware samples correctly identified by the model.

Recall –

Assesses the number of malware samples correctly identified by the model.

F1-score –

A combination of precision and recall.

directly from the platforms and provide real-time insights to users.

E. Explainability Analysis

To enhance the interpretability of the model, the SHAP (SHapley Additive exPlanations) method was applied to the model. It helps in identifying the features that are highly influential in the prediction made by the model.

The analysis done using the SHAP method identifies the top API call and permission features that are highly influential in the prediction done by the model. The analysis done using the SHAP method increases the transparency level in the decision-making done by the model for security analysts to understand the application identified as malware.

Confusion Matrix – It provides detailed information regarding the performance of the model.

VI. RESULTS AND DISCUSSION

The proposed model was assessed using a dataset comprising Android applications with a mix of benign and malicious applications. In addition, the performance of the proposed model was assessed using the standard evaluation parameters of accuracy, precision, recall, F1 score, and confusion matrix.

A. Model Performance

From the results obtained from the experiment, the proposed model has an accuracy of 93%. This indicates that the system is capable of effectively differentiating between benign and malicious Android applications. In addition, the precision value of 0.8657 indicates that the proposed system is correct in most of its predictions of malicious applications. Moreover, the value of the recall parameter of 0.8056 indicates that the proposed system is successful in detecting malicious applications in the dataset. This is because the F1- score is a balanced measure of precision and recall. Therefore, the result shows that the model has a good balance in detecting malware as well as avoiding false alarms. Based on the results obtained above, it is

evident that the proposed method is effective in the execution of Android malware detection tasks.

B. Classification Analysis

From the classification report above, the results obtained show the performance of the model in classifying different classes. In the case of the class representing benign applications (Class 0), the results obtained show that the precision was 0.94 and the recall was 0.96. This means that the majority of the legitimate applications were correctly classified by the system. In the case of the class representing malicious applications (Class 1), the results obtained show that the precision was 0.87 and the recall was 0.81. This means that the system is effective in the detection of malware as well. Although there was a small number of malware instances that were misclassified by the system, the results obtained show that the system is effective in the detection of malware. This is because the number of malwares is less compared to the number of benign applications.

C. Confusion Matrix Interpretation

The confusion matrix gives us insight into the performance of the model in making its predictions. Out of 247 benign applications, 238 were classified correctly, whereas 9 were classified incorrectly as malware. Out of 72 malware applications, 58 were classified correctly, whereas 14 were classified incorrectly as benign.

The results show that the model is capable of making a high number of accurate classifications, with low false positives and false negatives. The low number of false classifications may be due to similarities in permission usage between benign and malware applications.

D. Feature Importance and Explainability

In order to improve the interpretability of the model, the feature importance was carried out using the SHAP (SHapley Additive exPlanations) tool. The SHAP analysis helped in determining the top 20 features in the prediction model. The analysis identified the top 20 Android permissions and API calls that are highly significant in the decision-making process.

The top features identified in the model are the permissions such as SEND_SMS, SYSTEM_ALERT_WINDOW, READ_PHONE_STATE, CHANGE_WIFI_STATE, and ACCESS_COARSE_LOCATION. These permissions are usually linked with suspicious activities such as sending messages without the user's consent, accessing device information, and changing the network state.

The feature importance visualization using the SHAP tool has shown the significance of the permissions in the detection of malware.

E. Discussion

As indicated by the experimental results, the proposed model of explainable neural networks was able to accurately identify Android malware while at the same time providing insights into the features used in the process of prediction. The application of deep learning models coupled with the incorporation of explainability tools such as SHAP not only helped improve the accuracy of the detection process but also improved trust in the process.

As indicated, the experimental results show that the proposed model was effective in detecting malicious Android applications based on API call features, thus being a potential solution in enhancing mobile security.

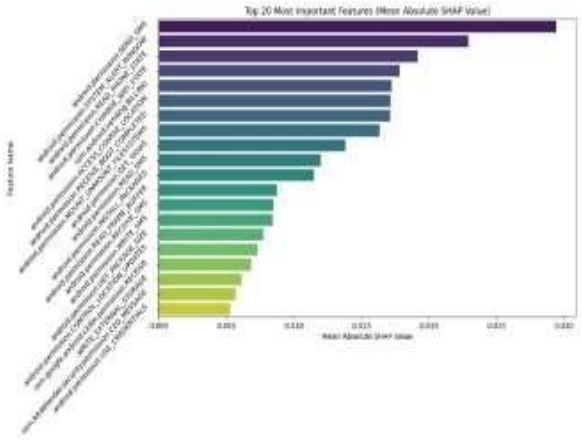
```
F1-Score: 0.8345
Precision: 0.8657
Recall: 0.8056

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95       247
           1       0.87      0.81      0.83        72

    accuracy                           0.93       319
   macro avg       0.91      0.88      0.89       319
weighted avg       0.93      0.93      0.93       319


Confusion Matrix:
[[238   9]
 [ 14  58]]
```



VII. CONCLUSION AND FUTURE WORK

The number of Android devices and applications is increasing day by day. This has resulted in the increase of malware attacks. Therefore, the development of efficient detection techniques is very important for the security of Android devices. The reference papers highlighted the importance of machine learning and deep learning techniques in the detection of Android malware. Various studies were conducted to prove the effectiveness of the analysis of application behavior based on API call patterns in distinguishing between malicious and benign applications.

However, the machine learning approach is based on the concept of a black box system. Therefore, it is very difficult to understand the decision-making process. Recent studies were conducted on the integration of the concept of explainable artificial intelligence in improving the transparency of the machine learning approach. This approach is effective in identifying the most important features in the classification results. From the analysis of the reference papers, it is evident that the integration of the neural network approach and API call analysis is effective in the detection of Android malware. Moreover, the integration of the concept of explainable AI is effective in improving the reliability and trustworthiness of the detection system. Therefore, the proposed approach of explainable AI-based robust Android malware detection using API call analysis is effective. Based on the analysis of the reference papers, it is evident that the combination of the neural network model and API call analysis is a robust approach for the detection of Android malware.

Another area that can be explored in the future is the development of advanced explainable artificial intelligence techniques for the development of malware detection

systems. Although the current focus is on the determination of the most important features that affect the system's prediction, in the future, the system can be enhanced to provide explanations about the different classifications. This will increase the level of trust in the system for the detection of malware. In the future, the system for the detection of Android malware can be enhanced by using advanced deep learning techniques to understand the complex relationships that exist between API calls and application behaviors. This will improve the accuracy of the system in the detection of malware because the system will be able to understand the complex behaviors of the applications, thereby increasing the chances of detecting newly emerging malware that may not be detected by the system.

In the future, the system can be enhanced through the application of a combination of static and dynamic analysis for the development of efficient malware detection systems. Static analysis refers to the study of application code and permissions, whereas dynamic analysis refers to the study of application behaviors.

REFERENCES

[1] Y. Liu, C. Tantithamthavorn, and L. Li, "Explainable AI for Android Malware Detection: Towards Understanding Why the Models Perform So Well," IEEE International Symposium on Software Reliability Engineering, pp. 169–180, 2022.

[2] M. N. AlJarrah, Q. M. Yaseen, and A. M. Mustafa, "A Context-Aware Android Malware Detection Approach Using Machine Learning," Information, vol. 13, no. 12, 2022.

[3] S. Aurangzeb and M. Aleem, "Evaluation and Classification of Obfuscated Android Malware Through Deep Learning Using Ensemble Voting Mechanism," Scientific Reports, vol. 13, 2023.

[4] N. Aslam et al., "Explainable Classification Model for Android Malware Analysis Using API and Permission-Based Features," Computers, Materials & Continua, vol. 76, no. 3, pp. 3167–3188, 2023.

[5] A. Muzaffar, H. R. Hassen, H. Zantout, and M. A. Lones, "Investigating Feature and Model Importance in Android Malware Detection Using Machine Learning Methods," 2023.

[6] C. Palma, A. Ferreira, and M. Figueiredo, "Explainable Machine Learning for Malware Detection on Android Applications," Information, vol. 15, no. 1, 2024

[7] S. Nazarinezhad, N. Khosrojerdi, and A. R. Shafieesabet, "Android Malware Detection by XGBoost Algorithm," Journal of Artificial Intelligence, Applications and Innovations, vol. 1, no. 3, pp. 31–37, 2024.

[8] W. Sun, "Malicious Software Identification Based on Deep Learning Algorithms and API Feature Extraction," EURASIP Journal on Information Security, 2025.

[9] S. Zhang et al., "A Malware-Detection Method Using Deep Learning to Fully Extract API Sequence Features," Electronics, vol. 14, no. 1, 2025.

[10] P. G. Bringas et al., "Understanding the Black Box: Android Malware Detection Through Explainable AI," Logic Journal of the IGPL, 2026.