Holistic Smartphone Data Protection System Integrating Android App Analysis And Secure Metadata Tracking

R.Sowmiya[1], S.Ahamed Firaz[2],M.Sanjay Kumar[3], T.P.Mohamed Sowban[4], S.Mohamed imran[5]
[1]Assist prof, Dept of Artificial Intelligence and Data Science
[2, 3, 4, 5] Dept of Artificial Intelligence and Data Science
[1, 2, 3, 4, 5] Mohamed Sathak Engineering College, Kilakarai, Tamil Nadu, India

Abstract- *This project aims to design a *Comprehensive Smartphone Data Security System* to enhance security of Android phone by smart malware detection and metadata management. With the increased use of mobile devices for critical applications such as telephony and banking, traditional signaturebased security tools are no longer effective to identify advanced types of malware like zero-day and polymorphic attacks. Our system addresses these issues by employing a machine learning based model using XGBoost algorithm to detect malware Android apps by considering both static and dynamic features, such as permissions, API calls and usage patterns. This enables efficient detection of malicious apps with minimal false positives. In addition, we implement a blockchain-like JSON-based module to ensure data integrity and transparency, by storing the detection results and information about the model. This guarantees integrity and allows pre-installation authentication of apps. Our system also incorporates other important components such as a Malware Level Indicator and a proactive Blocker to prevent the installation or distribution of critical applications. Overall, our system provides a scalable, real-time and efficient system to improve mobile security against the ever-evolving threats.*

Keywords: AI, Sign Language Recognition, ISL, Trans former, Real-Time System, Gesture Recognition, TTS, Speech Recognition.

I. INTRODUCTION

With the growing use of Android smartphones for communication, financial transactions and storing personal data, privacy and security are at risk of being compromised by malware attacks. The growing trend of distributing harmful apps from different channels has prompted a need for more efficient and effective protection. This project is motivated by the shortcomings of existing malware detection systems, which are unable to effectively identify new and advanced malware and may generate false positives or negatives. This leaves vulnerabilities that may result in data leaks and financial loss. The goal of this project is to develop an intelligent system for Android malware detection, using machine learning to accurately classify malware and blockchain to securely and reliably confirm the results of the analysis. The result of the system is a reliable, efficient and secure system offering realtime malware detection, blocking the installation of malicious applications, and secure metadata tracking, contributing to enhanced security for mobile devices.

II. LITERATURE REVIEW

Several studies have been conducted in the field of malware detection of Android using different methods. Hanxun Zhou and Xinlin Yang (2020) came up with a system to analyze malware using fewer metrics based on the similarity of additional features and GRU models by integrating Static Components Analysis and Comparing model, namely SIMGRU-based. The method improves the accuracy of detection much higher than traditional GRU models; however, it is not a dynamic analysis alternative, and is readily influenced by code obfuscation diminishing its effectiveness to zero-day attacks.

Similarly, Pengbin Feng and Jianfeng Ma (2019) have suggested a dynamic malware detection method, which involves ensemble learning. Their system relies on system traces to extract features and stacking techniques of classification. This method is quite accurate and can be used to overcome obfuscation better than the analysis of the static code, but requires more tools and less efficient in combating networkbased attacks.

In addition, Hanqing Zhang and Senlin Luo (2019) developed a behavioral semantic analysis method, which analyzes the relationships between API calls on the method level. High accuracy and quicker analysis are possible in their approach though they are sensitive to obfuscated malware and require proper API extraction.

Moreover, Lu Huang and Jingfeng Xue (2023) came up with API order-based malware detection algorithm whereby API clustering techniques were employed. This method enhances the classification of behavior and provides competitive outcomes, yet it has weaknesses of avoiding methods and needs dynamic analysis assistance.

Lastly, Abhinandan Banik and Jyoti Prakash Singh (2023) suggested a permission-based detection algorithm with the FPGrowth algorithm and neural networks. Their method is very accurate and makes good use of real permission patterns at the cost of being very costly to compute and necessitating fine tuning of neural network parameters.

III. RELATED WORK

A plethora of approaches have been explored for Android malware detection. Initial approaches mainly focused on signature-based detection, which involves matching applications with a set of known malware signatures. While these methods are effective in detecting known malware, they are incapable of detecting zero-day and polymorphic malware, as they rely on patterns.

To address these challenges, machine learning approaches have been proposed for malware detection. T. Reddy Gadekallu et al. developed a deep learning approach for hand gesture recognition based on optimised convolutional neural networks, showcasing the potential of smart models in pattern recognition. Likewise, several research works have used techniques like Decision Trees, Random Forest and Support Vector Machines for Android malware detection and classification with better accuracy than conventional approaches.

More recent work has explored ensemble methods such as XGBoost that offer improved accuracy by training a strong learner from a collection of weak learners. XGBoost has gained popularity for its effectiveness in dealing with large datasets, preventing overfitting, and capturing complex feature relations, making it suitable for malware classification.

Apart from machine learning, feature extraction is also essential in finding out malicious apps. Research has highlighted the role of static and dynamic analysis, such as permissions, API usage and behavioural patterns. These features assist in building a complete behavioral profile of applications, aiding in their classification.

Recently, blockchain has been used to improve the security and integrity of malware detection. The use of decentralized ledgers for storing malware and detection outcomes has been investigated to ensure immutability and tamper-proof mechanisms. The combination of Blockchain with machine learning offers a secure way to authenticate machine learning models and securely log application analysis results.

Although these approaches have made significant progress, there still remain issues with computational efficiency, realtime detection, and the lack of integration between secure data management and intelligent detection. The proposed system integrates machine learning for malware detection with XGBoost, and blockchain for secure metadata monitoring, offering scalable, precise and tamper-proof Android security.

IV. PROPOSED METHODOLOGY

The suggested system will be an AI-based two-way system of communication allowing a smooth interaction between deaf and hearing people. The methodology incorporates sign language recognition, speech processing and avatar based sign synthesis to maintain real-time and correct communication. The total workflow includes three large modules, i.e., gesture recognition, speech-to-text conversion, and sign animation generation.

A. System Overview

The proposed system includes the analysis of Android apps (APK files) by extracting important static features and employing a machine learning model to classify them. To enhance security and maintain trust, blockchain is used to

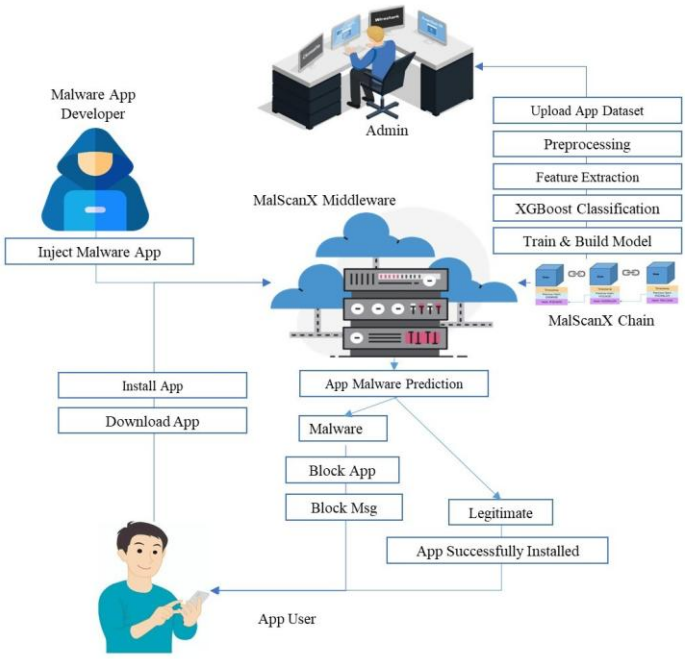

Fig. 1. proposed methodology

securely store the trained model and prediction results. The integrated approach improves detection and ensures security data integrity.

B. Data Collection

We obtain a large dataset of benign and malicious Android apps from trusted sources. The dataset contains APKs and their feature vectors. The samples are properly labeled for supervised learning. The dataset is carefully prepared to maintain diversity and to be balanced.

C. Data Preprocessing

The dataset is preprocessed before training to enhance the quality of data and training outcomes. This includes:

Filtering of inconsistent and missing data Scaling of numeric features Encoding of categorical attributes Division of the data into training, validation and testing subsets

These processes prepare clean and structured data for the model

D. Feature Extraction

Feature extraction plays a critical role in identifying malware behavior. The system extracts key static features from APK files, including Permission usage API call patterns Manifest components Intent structures Code-level behavioral indicators

The features come together to form a pattern of behaviour that is malicious.

E. Machine Learning Model

The features feed a potent machine learning model built using XGBoost. This is an efficient, scalable and interactive model.

It is trained to detect patterns of types of goodware and malware. We optimise the model using hyperparameter tuning to improve the accuracy and prevent overfitting.

F. Training and Testing

The model is built and validated using the dataset with measurements such as:

Accuracy Precision Recall F1-score Testing is to ensure good detection and low false alarms.

G. Store the Model in Blockchain

The model is stored in a secure Blockchain in a JSON form.

The hash of the model is also stored.

This mechanism guarantees:

Tamper-proof model storage Transparent verification Secure access to the model The model's integrity is monitored with every prediction request using the hash

H. APK Analysis and Prediction

Once an application is installed or uploaded by a user, the following process takes place:

APK validation Feature extraction Model verification via blockchain Classification of malware by the model

This allows the app to be labelled as malicious or not.

I. Log result on blockchain

Once the application is predicted, the classification is stored on the blockchain, which contains:

Application identifier Classification result Confidence score Timestamp

This allows verification in the future and avoids re-analysing the same applications.

J. Risk Level Assessment

The classification results are then used to map the risk level such as:

Low Medium High Critical

It's easier for users and administrators to understand the threats.

K. Malware Blocking Mechanism

When an app is detected as malicious, a blocking mechanism is triggered which:

Stops it installing on devices Prevents developers from publishing malicious apps

This improves system security.

L. Alert and Notification System

It offers instant notifications to users and administrators of security threats. The alerts provide

information about the threat and recommended actions to be taken.

M. Continuous Model Update

To address dynamic threat landscape, it allows regular updates:

New datasets are incorporated Model is retrained Updated model is securely stored on blockchain
This ensures ongoing effectiveness and performance.

V. IMPLEMENTATION DETAILS

This is a modular design of machine learning and blockchain secured Android malware detection system.

A. Development Environment

It's made with Python 3.8, Flask for the web app and

MySQL database. The Machine Learning model is built with XGBoost, NumPy and Pandas libraries are used. Storage is done using a Blockchain (JSON based).

B. Feature Extraction and Model Implementation

Users will submit APKs which need to be analysed statically. The system will analyse the permission, API calls and components extracted from the app's manifest and generate features.

The features are then used to classify the app as malicious or benign using the pre-trained XGBoost model. It is trained to be accurate as well as have a low false positive rate.

C. Blockchain Integration

The data and machine learning model is saved to the blockchain in JSON format. But the model is checked by computing the hash of the model. The model is verified on the blockchain for predictions.

D. Prediction and Result Handling

The system can predict Malware APKs. The results are:

Stored into the database for user's reference Logged in blockchain for verification

And also the risk factor (Low, Medium, High, Critical) is mentioned.

E. Security and Notification

Malware is prevented from installing. Security admins and users are also given warnings and notifications.

VI. PERFORMANCE METRICS

The traditional machine learning measures used for classification problems have been applied for the proposed Androidbased malware detector. They are used for measuring effectiveness and accuracy of the XGBoost model.

A. Confusion Matrix

Confusion matrix of the classifier is:

True Positive (TP): Infected apps True Negative (TN): Correctly detected benign apps False Positive (FP): Benign apps incorrectly detected False Negative (FN): Malware apps incorrectly detected

B. Accuracy

Accuracy is the degree of correctness.

Accuracy = TP + TN$\frac{TP}{+TN+FP+FN}$

Measures the number of correct predictions The closer to 1 the better

C. Precision

The proportion of malware that is predicted as malware.

Precision = TP$\frac{TP}{+FP}$

Important to reduce false alarms Increasing precision → reduce false positives

D. Recall (Sensitivity)

Recall is the probability of detecting malware.

Recall = TP$\frac{TP}{+FN}$

Important for security systems High recall → few missed malware

E. F1-Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Used when data is unbalanced Measures how well the model works

F. ROC Curve and AUC

ROC (Receiver Operating Characteristic) - graph of True

Positive Rate vs False Positive Rate
AUC (Area Under Curve) - model performance
A high AUC (close to 1) → good model
Very low AUC (close to 0.5) → poor model

G. Performance Significance

Accuracy ensures overall correctness Precision avoids false alarms Recall ensures malware is not missed F1-score is a mixture of precision and recall

These establish the accuracy and detection relationship of the proposed malware detection algorithm.

VII. RESULT AND DISCUSSION

A. Experimental Evaluation

A variety of Android applications with benign and malicious apps were used to test the proposed system using a diverse dataset of Android applications. XGBoost model was used to classify the features used to classify.

The experimental findings have shown that the model has a high classification efficiency with:Accuracy: 96

Precision: Malware prediction has high consistency. Recall: Good skill with identifying bad applications. F1-Score: Good performance in all classes.

As the confusion matrix displays, most cases of the malware are appropriately recognized, and there are few instances of misclassification.

B. Comparative Analysis

Compared to conventional methods of detection:

Signature-based approaches- do not identify unknown (zeroday) malware. Rule-based systems* need to be updated on an ongoing basis. Simple ML models* indicate reduced accuracy as they learn few features.

The proposed system:

Makes inferences about the data. As well, adjusted to new malware types. Gives a more accurate classification results.

C. Effectiveness of Feature-Based Detection

Detection performance is greatly enhanced by utilizing static features like permissions, API calls, and manifest components. The attributes have been used to detect insidious activities even of a legitimate application.

This implies that feature engineering is an important part in enhancing the predictive power of the model.

Blockchain in System Reliability.

The implementation of Blockchain means a new layer of credibility and safety:

Assures that the model trained is not distorted. Keeps unchangeable records of detectives. Allows checking of applications that are analyzed earlier.

This makes them less reliant on centralized systems and enhances transparency.

Discussion of Findings.

The experimental results emphasize that the suggested system:

secure and tamper-proof authentication. Upholds real-time malware detection and prevention.

Nevertheless, there are still some obstacles:

The analysis at the run time might not be captured. Malware that is highly obfuscated may decrease the detection accuracy. The quality of the dataset affects the performance of a system.

TABLE I
MALWARE DETECTION SYSTEM COMPARISON

System	ML-Based	Real-Time Detection	Blockchain Security	Zero-Day Detection	Accuracy
Signature-Based (2020)	No	Yes	No	No	Low
Static Analysis (2021)	Partial	Yes	No	Limited	Medium
Dynamic Analysis (2022)	Yes	No	No	Yes	Medium
Basic ML Model (2023)	Yes	Yes	No	Partial	Medium
DL-Based Detection (2024)	Yes	Yes	No	Yes	High
Proposed System	Yes	Yes	Yes	Yes	High

VIII. CONCLUSION

This paper suggests an Android malware detection system that's safe as well as smart because of the machine learning approach and blockchain verification. The system employs the XGBoost computer learning algorithm to analyse the features of apps and then determines whether it is benign or malware.

The experimental outcomes demonstrate the efficiency of the proposed system in terms of high accuracy of detection with the least false negatives and false positives. The propertybased analysis method helps the system to identify complex patterns of malwares, such as those never encountered before, and overcomes the limitations of the traditional signaturebased systems.

Furthermore, Blockchain assures the security of the system, by ensuring that the trained model and the result of the detection process are unmodified. This ensures secure verification and boosts user trust in the malware detection system. It also offers real-time security through automated forecasting, risk gauge and a preventive mechanism that limits the downloading and installation of harmful applications in the solution. The suggested solution can be applicable in mobile security applications.

In conclusion, the proposed method is a scalable, precise and secure method of Android malware detection. This machine intelligence-verification hybrid strategy of decentralized private peer is efficient to handle the ever-evolving mobile threats.

REFERENCES

[1] Z. Z. Jundi and H. Alyasiri, "Android malware detection based on grammatical evaluation algorithm and XGBoost," in *Proc. IEEE AlSadiq Int. Conf. Communication and Information Technology (AICCIT)*, 2023.

[2] H. Bakır and R. Bakır, "DroidEncoder: Malware detection using autoencoder based feature extractor and machine learning algorithms," *Computers and Electrical Engineering*, vol. 110, p. 108804, 2023.

[3] S. E. Antony *et al.*, "Android malware detection using an AI-powered hybrid XGBoost-RF model for enhanced cybersecurity," *EPJ Web of Conferences*, vol. 354, p. 02007, 2026.

[4] H. Zhu, Y. Li, L. Wang, and V. S. Sheng, "A multi-model ensemble learning framework for imbalanced Android malware detection," *Expert Systems with Applications*, vol. 234, p. 120952, 2023.

[5] M. Mittal and P. K. G. Pandian, "Deep learning approaches to malware detection and classification," *International Journal of Multidisciplinary Innovative Research Methodology*, vol. 3, no. 1, pp. 70–76, 2024.

[6] S. Kumari *et al.*, "Decentralized malware attacks detection using blockchain," *ITM Web of Conferences*, vol. 53, p. 03002, 2023.

[7] R. Yumlembam, B. Issac, S. M. Jacob, and L. Yang, "IoT-based Android malware detection using graph neural network with adversarial defense,"*IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 8432–8444, 2023.

[8] Rana *et al.*, "Performance evaluation of machine-learning models for Android malware detection on the Ethereum blockchain," *Journal of Information Security*, 2019.

[9] S. Homayoun *et al.*, "A blockchain-based framework for detecting malicious mobile applications in app stores," *IEEE Access*, 2019.

[10] Fuji *et al.*, "Blockchain-based malware detection method using shared signatures of suspected malware files," *Journal of Computer Virology and Hacking Techniques*, 2020.