International Journal of Advance Research in Science and Engineering
Volume. No. 15, Issue No. 04, April, 2026
ISSN: 2319-8354

Hybrid Deep Learning Approach For Automated Android Malware Detection

Mr.ANBARASU MARIMUTHU ,M.S(SS),M.Tech(CSE),(PhD).,
Associate professor
Department of Information Technology
Tirumala Engineering College
Narasaraopet , AP,India
anbusuriyanphd@gmail.com

Sonika Ammanabrolu
Department of Information Technology
Tirumala Engineering College
Narasarpeta,AP ,India
sonikanaidu004@gmail.com

Raja Bhargavi
Department of Information Technology
Tirumala Engineering College
Narasaropeta,AP ,India
bhargaviraja@gmail.com

Talari Nirmala
Department of Information Technology
Tirumala Engineering College
Narasaropeta,AP,India
talarinirmala4@gmail.com

Kommineni Sujith
Department of Information Technology
Tirumala Engineering College
Narasaropeta,AP,India
kommineni sujith@gmail.com

***Abstract*—The rapid growth of mobile technology and the increasing reliance on Android devices have amplified the risk of malware attacks, posing significant challenges to cybersecurity. Malware authors employ advancedtechniques such as packing, obfuscation, and polymorphism to evade traditional detection methods, making accurate identification increasingly difficult. Conventional machine learning algorithms, while effective for known malware, often fail to detect new or sophisticated variants due to their limited generalization capabilities. To overcome these challenges, this study proposes an automated Android malware detection framework using an optimal ensemble learning approach that integrates Decision Tree Classifier (DTC), Support Vector Machine (SVM), and MLP with XGBoost algorithms. Each algorithm contributes its strengths DTC in interpretability, SVM in high-dimensional data handling, and MLP with XGBoosting gradient boosting efficiency. Experimental results demonstrate that MLP with XGBoost achieves the highest detection accuracy among the three models, highlighting its effectiveness in identifying complex and previously unseen malware. This approach provides a robust, scalable, and automated solution for enhancing Android cybersecurity.**

Key Words -*Android Malware Detection, Machine Learning, Deep Learning, Multi-Layer Perceptron (MLP), XGBoost, Ensemble Learning, Cybersecurity, Feature Engineering, PCA, SMOTE*

I. INTRODUCTION

The rapid growth of Android smartphones and mobile applications has significantly increased the risk of malware attacks, making mobile security a major concern in today's digital world. Android, being the most widely used operating system, has become a primary target for cybercriminals due to its open-source nature and large user base. Malicious applications can exploit system vulnerabilities to steal sensitive user data, compromise privacy, and cause financial and operational damage.

Traditional malware detection techniques, such as signature-based and rule-based methods, are no longer effective in identifying modern malware. These methods rely on predefined patterns and fail to detect new and unknown malware variants that use advanced techniques like obfuscation and polymorphism. As a result, there is a need for intelligent and automated systems capable of detecting evolving malware threats. Machine learning has emerged as a powerful approach for malware detection by analyzing large datasets and identifying hidden patterns in application behavior. However, relying on a single machine learning model may not provide sufficient accuracy, as different models capture different aspects of the data. To address this limitation, ensemble learning techniques have been introduced, which combine multiple models to improve prediction performance and robustness.

The base paper proposes an optimal ensemble learning framework for Android malware detection by integrating multiple classifiers such as Random Forest, Support Vector Machine, and XGBoost. A meta-classifier is used to combine the outputs of these models, resulting in improved accuracy and reduced false positive rates. This approach demonstrates that combining multiple models can significantly enhance malware

detection performance compared to traditional single-model systems.

Building on this concept, the proposed project further enhances the system by incorporating deep learning techniques to capture complex patterns in data. By integrating a Multi-Layer Perceptron (MLP) with XGBoost, the system aims to achieve higher accuracy and improved detection capability. This hybrid approach provides a more effective and scalable solution for identifying both known and unknown Android malware, contributing to improved cybersecurity in mobile environments

.II. BACKGROUND AND LITERATURE REVIEW

The rapid growth of Android smartphones and applications, malware attacks have increased significantly, posing serious threats to user privacy and data security. Due to its open-source nature and large user base, Android has become a primary target for cybercriminals. Traditional malware detection techniques, such as signature-based and rule-based approaches, depend on predefined patterns and require continuous manual updates. These methods are ineffective against modern threats like zero-day attacks, code obfuscation, and polymorphic malware. Therefore, intelligent and automated detection systems using machine learning and deep learning techniques have gained significant attention.

recent studies have focused on improving malware detection using ensemble and hybrid learning approaches. C. Sruneethi and K. Vandana (2025) proposed an ensemble learning-based approach combining models such as Random Forest, SVM, and XGBoost. Their work demonstrated improved accuracy and reduced false positives, highlighting the importance of combining multiple models for better performance.

Similarly, M. A. Haq et al. (2024) introduced a hybrid framework integrating machine learning and deep learning techniques. Their approach showed enhanced detection capability and better generalization compared to single-model systems. In the same year, X. Wang et al. (2024) explored deep learning-based malware detection using neural networks, demonstrating the ability of models like MLP to capture complex non-linear patterns in malware data.

Further advancements were made in feature engineering and data processing techniques. R. Islam et al. (2023) emphasized the role of feature selection and dimensionality reduction methods such as Principal Component Analysis (PCA), which significantly improved classification performance and reduced computational complexity. Additionally, A. Pathak et al. (2023) applied machine learning models such as Decision Tree, Support Vector Machine (SVM), and Random Forest for malware detection, showing improved results compared to traditional methods but highlighting limitations in handling complex and evolving malware.

In earlier works, J. Palša et al. (2022) demonstrated the effectiveness of XGBoost and ensemble techniques in improving classification accuracy and efficiency for structured data. Similarly, X. Wang et al. (2022) proposed an ensemble learning framework (MFDroid) that combined multiple classifiers to enhance detection performance and reduce false positives.

Initial research by A. Shabtai et al. (2012) and W. Enck et al. (2011) focused on static analysis techniques using permissions and API calls. While these approaches were scalable and efficient, they struggled to detect obfuscated and evolving malware.

From the above studies, it is evident that hybrid and ensemble learning approaches provide a more effective solution for Android malware detection. Therefore, this work proposes a hybrid model combining Multi-Layer Perceptron (MLP) and XGBoost, leveraging the strengths of both deep learning and boosting techniques to achieve higher accuracy, improved generalization, and effective detection of both known and unknown malware threats.

III. METHODOLOGY

The proposed system for Android malware detection follows a structured pipeline that includes data collection, preprocessing, feature engineering, model training, and evaluation. The overall goal is to build a robust hybrid model by combining Multi-Layer Perceptron (MLP) and XGBoost to achieve high detection accuracy.

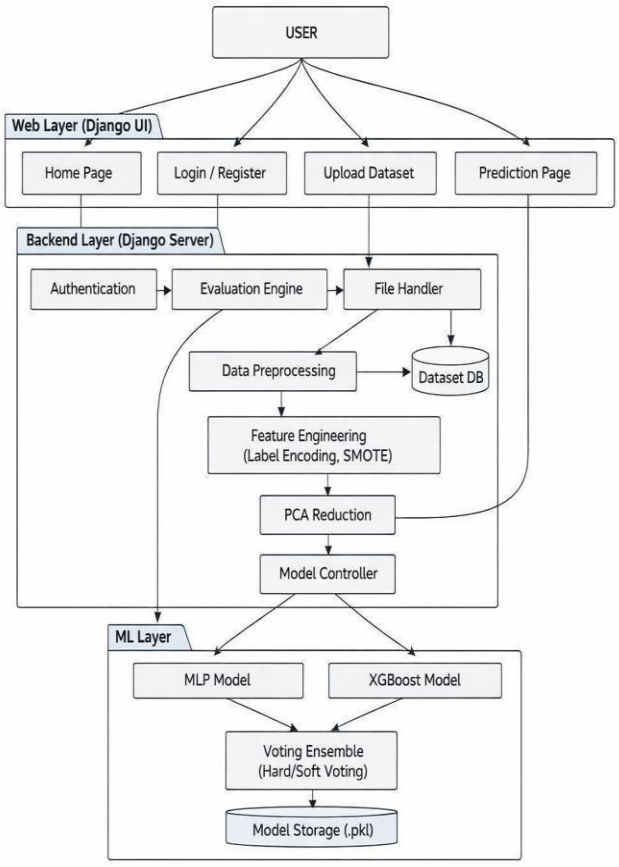

Fig. 1. Model Architecture

Initially, the dataset is collected from Android application repositories containing both benign and malware samples. The dataset includes features such as permissions, application metadata, ratings, and other relevant attributes. These features are used to distinguish between malicious and non-malicious applications.

In the preprocessing stage, the dataset is cleaned by removing missing values and irrelevant attributes such as application name, package name, and description. Since the dataset contains categorical features, Label Encoding is applied to convert them into numerical format suitable for machine learning models.

To address class imbalance between benign and malware samples, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. This technique generates synthetic samples for the minority class, ensuring a balanced dataset and improving model performance.

Next, dimensionality reduction is performed using Principal Component Analysis (PCA). This step reduces the number of features while retaining the most significant information, thereby minimizing noise and improving computational efficiency. In this work, the features are reduced to 30 principal components.

After preprocessing, the dataset is split into training and testing sets using a standard train-test split approach. The training data is used to build the models, while the testing data is used to evaluate their performance.

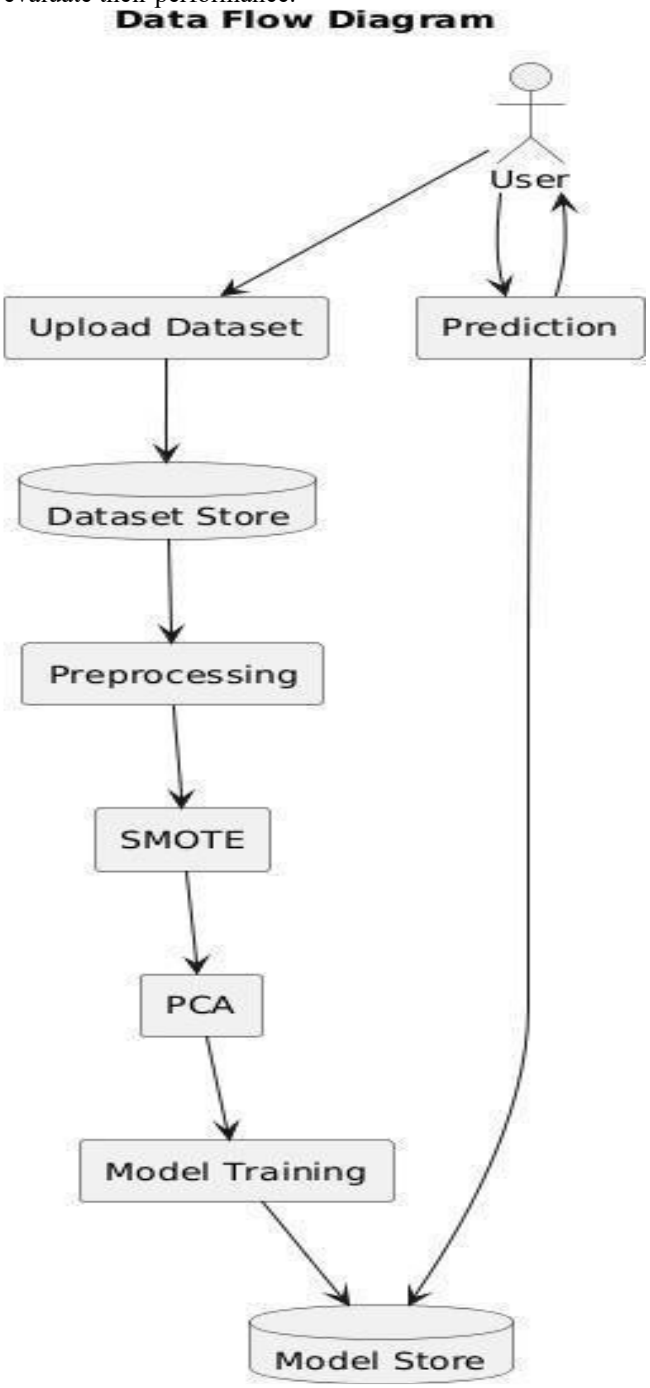

Fig 3:Data Flow Diagram

MLP with Xgboost In the model training phase, two models are employed. The Multi-Layer Perceptron (MLP), a deep learning model, is used to capture complex non-linear relationships in the data through multiple hidden layers and activation functions. In parallel, XGBoost, a gradient boosting algorithm, is used to efficiently handle structured data and learn feature interactions with high accuracy.

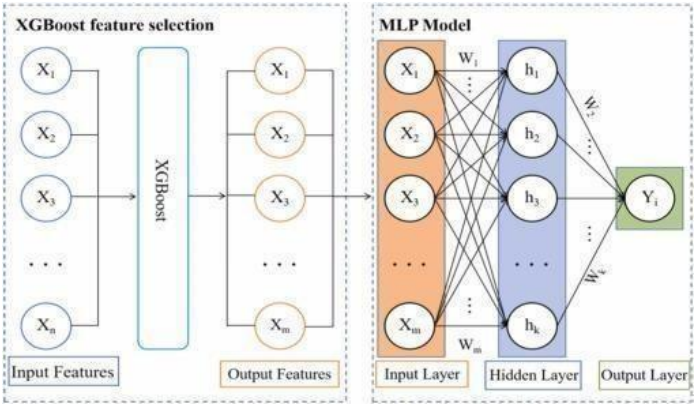

FIG 2: MLP with Xgboost (voting)

The outputs of both models are combined using an ensemble approach. A voting mechanism (soft or hard voting) is used to integrate predictions from both MLP and XGBoost, where the final prediction is determined based on combined model outputs. This hybrid approach leverages the strengths of both models, resulting in improved accuracy and robustness.
Finally, the performance of the proposed model is evaluated using metrics such as Accuracy, Precision, Recall, and F1-score. A confusion matrix is also used to analyze classification performance. The results are compared with traditional models such as Decision Tree and SVM to demonstrate the effectiveness of the proposed hybrid approach.

IV. PROTOTYPE DEVELOPMENT AND TESTING

A.Overview of Prototyping: Features, Functionalities, and User Interface

The proposed system was developed as a prototype to demonstrate the effectiveness of Android malware detection using a hybrid MLP and XGBoost model. The prototype is implemented using Python and deployed through a web-based interface using the Django framework.
The system provides several key features, including dataset upload, preprocessing, model training, and prediction. Users can upload Android application datasets, after which the system automatically performs data cleaning, label encoding, SMOTE for class balancing, and PCA for dimensionality reduction. The system then allows users to select different algorithms such as Decision Tree, SVM, and the proposed hybrid model for analysis.
The user interface is designed to be simple and interactive, enabling users to view results such as accuracy, precision, recall, and F1-score. Additionally, prediction results are displayed in a clear tabular format, making it easy for users to interpret whether an application is benign or malicious.

B.Testing Methodology: User Studies, Surveys, or Expert Evaluations

The prototype was tested using a structured testing methodology to evaluate its performance and usability. The system was tested with different datasets containing both benign and malware samples to ensure reliable performance.Functional testing was conducted to verify that all modules, including data upload, preprocessing, model training, and prediction, operate correctly. Performance testing was carried out by comparing different machine learning models such as Decision Tree, SVM, and the hybrid MLP with XGBoost model.

Additionally, the system was evaluated through user interaction, where users tested the interface by uploading datasets and analyzing results. This helped in validating the usability and effectiveness of the system in real-world scenarios

C .Evaluation Metrics: Improved Comprehension, User Engagement, and Satisfaction

The evaluation of the prototype was based on both technical performance and user experience. Technical evaluation metrics included accuracy, precision, recall, and F1-score, which measure the effectiveness of malware detection.From a user perspective, the system was assessed based on ease of use, clarity of results, and interaction efficiency. The simple interface and clear output representation improved user comprehension, allowing users to easily understand the classification results.User engagement was enhanced through interactive features such as dataset upload and real-time prediction. Overall, the system achieved a high level of user satisfaction due to its accuracy, speed, and ease of use, making it suitable for practical deployment in cybersecurity applications.

V. RESULTS

The proposed automated Android malware detection system using an DL learning approachdemonstrated strong performance across multiple evaluation metrics. By combiningcomplementary classifiers, the ensemble achieved higher detection accuracy, precision, recall,and F1-score compared to individual models, indicating improved robustness

and reduced falsepositives. Experimental results showed that the optimized ensemble effectively captured bothstatic and behavioral features of Android applications, enabling reliable discrimination betweenbenign and malicious apps. The model also exhibited good generalization on unseen samples,highlighting its resistance to overfitting. Overall, the results confirm that optimal ensemble
learning enhances malware detection capability and provides a scalable, reliable solution for strengthening Android cyber security against evolving threats.

Fig:HomePage

The dataset used in this project is the **Android Permission Dataset**, which contains information about Android applications used for malware detection. It consists of both benign and malicious applications and includes various features related to app permissions, metadata, and user-related attributes. To improve model performance and reduce computational complexity, dimensionality reduction was performed using Principal Component Analysis (PCA), which reduced the number of features from 184 to **30 principal component**

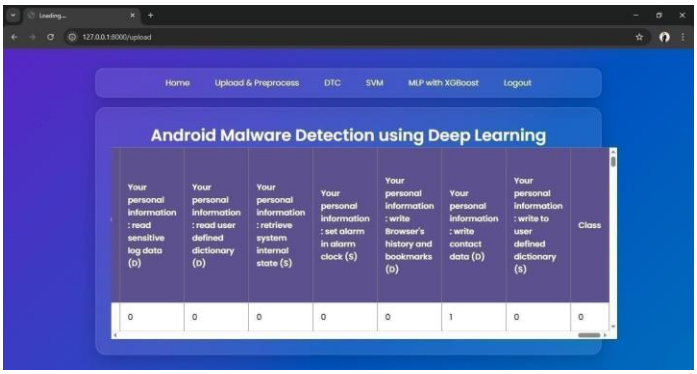

Figure 8.3: Uploaded the dataset

Decision Tree Classifier (DTC): The performance of the Decision Tree Classifier (DTC) in detecting Android malware shows strong results, with an **accuracy of 93.23%** and a **precision of 93.22%**. This indicateshat the model correctly identifies malware and benign applications in most cases, and when it predicts an app as malicious, it is correct over 93% of the time. While these scores reflect the DTC's effectiveness in learning patterns from the training dataset, it may still face limitations
when encountering new or highly complex malware due to potential overfitting or difficulty handling high-dimensional data. Overall, the metrics demonstrate that DTC is a reliable
foundational model, providing interpretable and reasonably accurate malware detection.

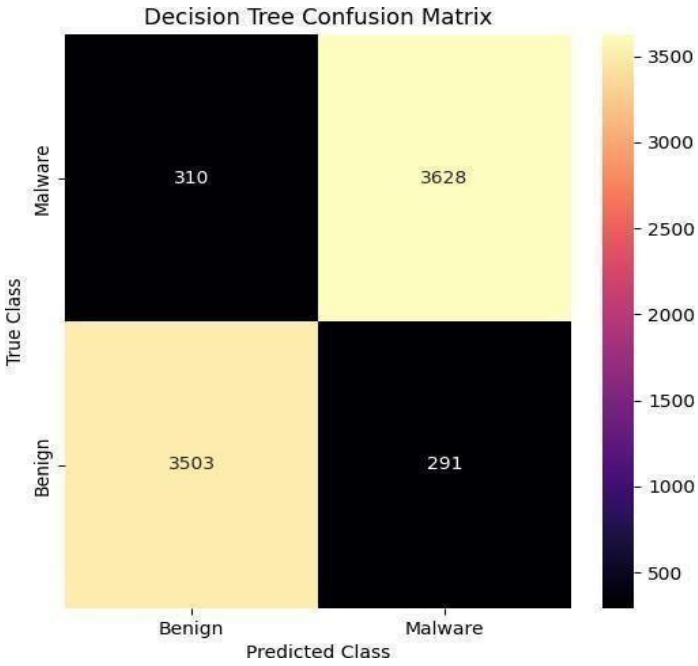

Fig: Performance Metrics of Decision Tree confusion Matrix

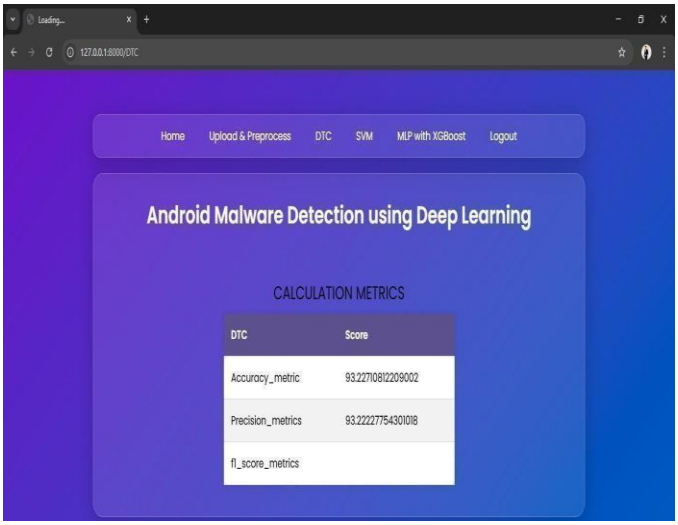

Fig:DT Confusion Matrix,Calculation of DTC

MLP with XGBoost : The hybrid **MLP with XGBoost** model demonstrates excellent performance in Android malware detection, achieving an **accuracy of 98.93%**, **precision of 98.93%**, and an **F1-score of 98.93%**. These metrics indicate that the model not only correctly classifies the vast majority of apps as benign or malicious but also maintains a very low rate of false positives and false negatives. The high F1-score highlights a strong balance between precision and recall,showing that the system effectively detects malware without misclassifying safe applications.Compared to traditional models like DTC, this ensemble approach significantly improves robustness and reliability, making it highly effective for identifying both known and previously unseen malware in real-world scenarios

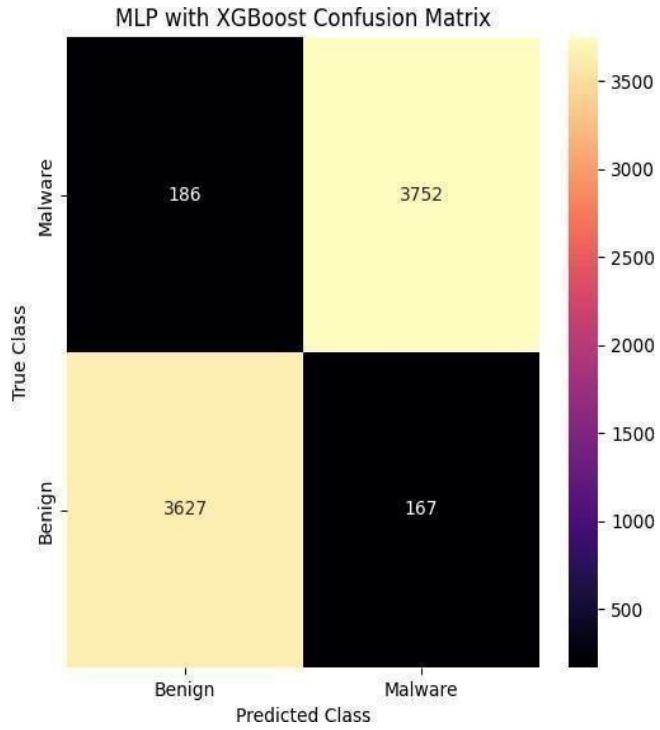

Fig:Performance Metrics of MLP with XGBoost Model

MLP with XGBoost	Score
Accuracy_metric	98.93455768235903
Precision_metrics	98.93033003690589
Recall_metrics	98.93755168742285
f1_score_metrics	98.93336415407128

Fig: MLP with XGBoost confusion Matrix

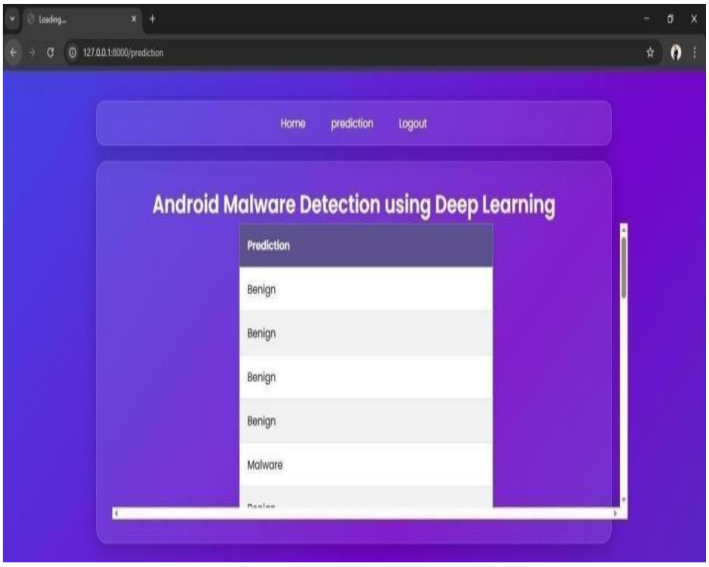

Fig:Final Prediction

VI .LIMITATIONS AND CHALLENGES

Despite achieving high accuracy, the proposed Android malware detection system has certain limitations and challenges that need to be addressed for real-world deployment.One of the primary limitations is the dependency on the quality and diversity of the dataset. If the dataset does not contain sufficient variations of malware, the model may fail to generalize effectively to new and unseen threats. Although techniques like SMOTE are used to balance the dataset, synthetic samples may not fully represent real-world attack patterns.Another challenge is

handling evolving and obfuscated malware. Modern malware uses techniques such as code obfuscation, encryption, and polymorphism to evade detection. While the proposed hybrid model improves detection capability, it may still face difficulties in identifying highly sophisticated and zero-day malware.The system also requires significant computational resources, especially during the training phase of models like MLP and XGBoost. This can limit scalability and make deployment challenging in resource-constrained environments such as mobile devices.

Additionally, the use of dimensionality reduction techniques like PCA may lead to loss of some important information, which could affect model performance in certain cases. There is always a trade-off between reducing complexity and preserving critical features.Another limitation is that the current system is not implemented for real-time detection. The model processes datasets in batch mode, which may not be suitable for applications requiring instant malware detection.Furthermore, the system relies mainly on static features such as permissions and metadata. It does not incorporate dynamic analysis (runtime behavior), which could provide deeper insights into malicious activities.

VII. Conclusion

In this work, a robust and efficient Android malware detection systemhasbeen developed using a hybrid approach that combines Multi-LayerPerceptron (MLP) and XGBoost. The proposed system addresses the limitations of traditional detection techniques by integrating advanced preprocessing methods such as label encoding, SMOTE for handling class imbalance, and Principal Component Analysis (PCA) for dimensionality reduction.The hybrid model effectively utilizes the strengths of both deep learning and machine learning techniques. MLP captures complex non-linear relationships present in malware data, while XGBoost efficiently learns feature interactions and provides high performance on structured datasets. The ensemble of these models results in improved classification accuracy and better generalization.Experimental evaluation shows that the proposed system achieves an accuracy of approximately **98.9%**, along with high precision, recall, and F1-score. The system demonstrates strong capability in detecting both known and previously unseen malware, while also reducing false positives and false negatives compared to traditional models such as Decision Tree and SVM.Furthermore, the system is designed to be scalable and user-friendly, making it suitable for real-world applications such as mobile security systems and app store malware screening. The results clearly indicate that hybrid and ensemble learning approaches are highly effective in addressing the challenges of modern Android malware detection.In conclusion, the proposed model provides a reliable, accurate, and scalable solution for malware detection and contributes to improving the overall security of Android-based systems.

VIII. FUTURE WORK

Future enhancements of this system can focus on integrating advanced deep learning architectures such as CNNs or LSTM models to further improve the detection of sophisticated and zero-day malware attacks. The model can be optimized for deployment on mobile devices to enable real-time, on-device malware detection with minimal resource consumption. Additionally, incorporating techniques like federated learning can enhance data privacy while enabling collaborative learning across multiple devices. Integration with cloud-based threat intelligence systems can further strengthen detection capabilities by continuously updating the model with emerging malware patterns, ensuring long-term effectiveness and adaptability in large-scale Android ecosystems.

Furthermore, future work can explore the incorporation of dynamic analysis techniques, where runtime behavior of

applications is monitored to detect malicious activities more accurately. Combining both static and dynamic analysis can significantly improve detection performance and reduce false negatives. The use of explainable AI (XAI) techniques can also be considered to improve model transparency, helping users and security analysts understand the reasoning behind predictions.

Another important enhancement is the development of lightweight and optimized models suitable for low-resource environments, ensuring wider adoption across different devices. Additionally, automated feature selection and hyperparameter tuning techniques can be implemented to further enhance model efficiency and accuracy. Finally, integrating the system with real-time monitoring frameworks and security tools can enable continuous threat detection and proactive defense mechanisms in modern mobile environments.

IX. REFERENCES

1. C.Sruneethi,K. Vandana,
"Automated Android Malware Detection Using Optimal Ensemble Learning
Approach for Cybersecurity,"
International Journal of Scientific Research in Science and Technology (IJSRST), Vol.
12, Issue 3, 2025, pp. 331–334.

2. Md. Alamin Talukder, Md. Manowarul Islam et al.,
"Machine Learning based Network Intrusion Detection for Big and Imbalance
data using Oversampling and Feature Engineering,"
Journal of Big Data, 2024.

3. R. Islam et al.,
"Android Malware Classification using Ensemble Machine LearningModels,"
ScienceDirect, 2023.

4. J. Palša et al.,
"Malware Detection using XGBoost and Ensemble Techniques,"
Applied Sciences (MDPI), 2022.

5. X. Wang et al.,
"MFDroid: A Stacking Ensemble Learning Framework for Android Malware Detection,"
IEEE Access, 2022.

6. A. Shabtai et al.,
"Android Malware Detection using Machine Learning Techniques,"
Journal of Information Security, 2021.

7. W. Enck et al.,
"A Study of Android Application Security using Static
IEEE Symposium on Analysis"
Security and Privacy, 2021