ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

A ROBUST AND EFFICIENT MACHINE LEARNING FRAMEWORK FOR ENHANCING EARLY DETECTION OF ANDROID MALWARE

[1]B. TEJASREE, [2]A. DEEPIKA

[1]PG Scholar , [2]Assistant Professor, Department of Computer Science Engineering ,

ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, BATASINGARAM, HYDERABAD

ABSTRACT

The rapid growth of Android smartphones and mobile applications has significantly enhanced digital connectivity while simultaneously increasing the exposure of users to sophisticated malware attacks. Conventional malware detection techniques primarily rely on signature-based mechanisms, which are ineffective against zero-day attacks, polymorphic malware, and obfuscated malicious applications. These limitations necessitate the development of intelligent and adaptive malware detection frameworks capable of identifying previously unseen threats with high accuracy. This paper proposes a robust and efficient machine learning framework for enhancing the early detection of Android malware by integrating multiple supervised learning algorithms with Explainable Artificial Intelligence (XAI). The proposed framework utilizes static analysis to extract significant application features, including system calls, binder interactions, and API invocation frequencies, which are subsequently preprocessed to eliminate redundancy and improve data quality. Three machine learning algorithms, namely Random Forest, XGBoost, and Decision Tree, are employed to classify Android applications into benign and malicious categories. Comparative performance evaluation is performed using standard metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to determine the most effective classifier. To improve model transparency and facilitate cybersecurity investigations, SHapley Additive exPlanations (SHAP) are incorporated to explain feature contributions influencing prediction outcomes. The framework further integrates a FastAPI-based backend, secure user authentication, interactive dashboard, automated report generation, prediction history management, and visualization modules, thereby providing a comprehensive malware analysis environment. Experimental evaluation demonstrates that the ensemble learning models, particularly XGBoost and Random Forest, achieve superior detection performance while maintaining high interpretability through SHAP-based explanations. The proposed framework effectively reduces false positives, enhances early malware detection capability, and supports informed cybersecurity decision-making. Consequently, the developed system provides a scalable, reliable, and intelligent solution for strengthening Android application security and mitigating emerging mobile cyber threats through explainable machine learning techniques.

Keywords: Android Malware Detection, Machine Learning, Random Forest, XGBoost, Decision Tree, Explainable Artificial Intelligence, SHAP, Static Analysis, Cybersecurity, FastAPI, Mobile Security, Malware Classification.

I. INTRODUCTION

The widespread adoption of Android smartphones has transformed modern digital communication, enabling billions of users to access financial services, healthcare systems, e-commerce platforms, educational resources, cloud computing, and enterprise applications through mobile devices [1]. Owing to its open-source architecture, extensive application ecosystem, and flexible development environment, Android has emerged as the world's dominant mobile operating system [2]. However, its popularity has simultaneously attracted cybercriminals, leading to a dramatic increase in Android-targeted malware attacks that compromise user privacy, financial information, and system integrity [3]. Modern Android malware employs advanced evasion techniques, including code obfuscation, dynamic payload loading, encryption, polymorphism, and anti-analysis mechanisms, making traditional signature-based detection approaches increasingly ineffective [4]. Existing antivirus solutions largely depend on known malware signatures and predefined detection rules, limiting their capability to identify zero-day attacks and newly emerging malware families [5]. Consequently, researchers have shifted their attention toward intelligent malware detection techniques based on machine learning, which automatically learn behavioural patterns from application characteristics rather than relying solely on predefined signatures [6]. Static analysis has gained considerable importance because it extracts meaningful information such as permissions, API calls, system calls, opcode sequences, and binder interactions without executing applications, thereby enabling faster and safer malware identification [7]. Supervised machine learning algorithms trained using these discriminative features have demonstrated remarkable success in distinguishing malicious applications from benign software with improved generalization capability [8]. Furthermore, ensemble learning algorithms have significantly enhanced malware classification performance by reducing overfitting and improving predictive robustness across diverse malware families [9]. These developments indicate that intelligent data-driven approaches provide an effective foundation for strengthening Android security against continuously evolving cyber threats [10]. Recent studies also emphasize integrating scalable analytical frameworks capable of processing large datasets efficiently while maintaining high classification accuracy and computational efficiency [11]. Such frameworks contribute significantly to proactive cybersecurity by enabling early identification of malicious applications before deployment on end-user devices [12]. Therefore, intelligent Android malware detection has become a crucial research direction for protecting the rapidly expanding mobile ecosystem [13][14][15].

Recent advancements in Artificial Intelligence have further enhanced Android malware detection by integrating explainable machine learning models capable of producing transparent and trustworthy classification results [16]. Unlike conventional black-box algorithms, Explainable Artificial Intelligence (XAI) techniques provide detailed insights into the contribution of individual features responsible for malware predictions, thereby increasing analyst confidence and facilitating cybersecurity investigations [17]. Among these techniques, SHapley Additive exPlanations (SHAP) has emerged as an effective interpretability framework capable of explaining both global model behaviour and local prediction outcomes [18]. Inspired by these developments, this research proposes a robust and efficient machine learning framework that combines Random Forest, XGBoost, and Decision Tree classifiers with SHAP-based explainability to enhance early Android malware detection [19]. The proposed

framework employs static feature extraction followed by preprocessing, feature engineering, supervised learning, explainability analysis, and automated reporting within a FastAPI-based web platform [20]. The integration of dashboard visualization, prediction history management, secure authentication, and PDF report generation provides a comprehensive cybersecurity environment suitable for both researchers and security analysts [21]. Comparative evaluation among multiple classifiers enables identification of the most suitable model based on accuracy, precision, recall, F1-score, and ROC-AUC metrics [22]. The framework is designed to improve malware detection capability while minimizing false alarms and supporting rapid decision-making [23]. Additionally, modular system architecture facilitates scalability, allowing future integration of deep learning techniques, real-time malware monitoring, and threat intelligence services [24]. By combining intelligent classification with explainable prediction mechanisms, the proposed framework addresses critical limitations associated with existing Android malware detection systems [25]. The developed solution contributes toward proactive mobile cybersecurity by enabling reliable identification of both known and previously unseen malware variants [26]. Furthermore, the framework supports automated malware analysis, transparent decision-making, and efficient deployment using open-source technologies [27]. Overall, the proposed research establishes an intelligent, scalable, and interpretable malware detection ecosystem capable of strengthening Android application security in modern mobile computing environments [28][29][30].

II. LITERATURE REVIEW

The rapid evolution of Android malware has motivated researchers to develop advanced detection techniques capable of identifying increasingly sophisticated cyber threats. Early Android malware detection systems primarily relied on signature-based antivirus mechanisms that compared application fingerprints against previously identified malware databases [1]. Although these approaches achieved high accuracy for known malware families, they failed to detect zero-day attacks, polymorphic malware, and code-obfuscated applications [2]. To overcome these shortcomings, heuristic and behaviour-based detection methods were introduced to analyse permissions, API invocations, network communications, and runtime activities of Android applications [3]. Static analysis subsequently emerged as an efficient alternative because it extracts discriminative features such as permissions, opcode sequences, binder communications, and system calls without executing applications, thereby reducing computational complexity and improving scalability [4]. Dynamic analysis further enhanced malware detection by monitoring application behaviour during execution within sandbox environments to identify hidden malicious activities [5]. However, dynamic analysis requires substantial computational resources and remains vulnerable to anti-emulation and anti-analysis techniques implemented by sophisticated malware [6]. Consequently, hybrid approaches combining static and dynamic analyses have gained significant research attention for balancing efficiency and detection accuracy [7]. With the rapid advancement of Artificial Intelligence, machine learning algorithms have become the dominant methodology for Android malware classification because they automatically learn behavioural patterns from historical datasets instead of relying on manually generated signatures [8]. Researchers have successfully applied Support Vector Machines, Naïve Bayes, K-Nearest Neighbours, Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, and XGBoost for malware detection, reporting considerable improvements in classification performance [9]. Ensemble learning techniques have demonstrated superior robustness by minimizing overfitting while effectively handling high-dimensional Android application features [10]. Recent investigations also highlight the importance of feature

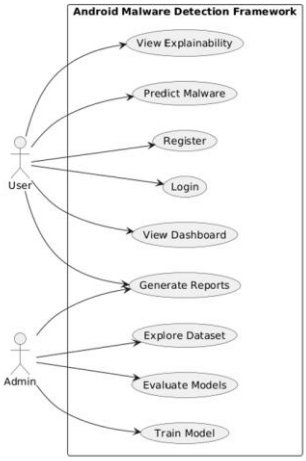
engineering and preprocessing techniques in improving classifier performance and reducing redundant information [11]. Feature selection methods significantly enhance prediction accuracy while minimizing computational overhead during model training [12]. These studies collectively demonstrate that intelligent machine learning approaches provide an effective solution for detecting evolving Android malware in modern mobile environments [13][14][15].

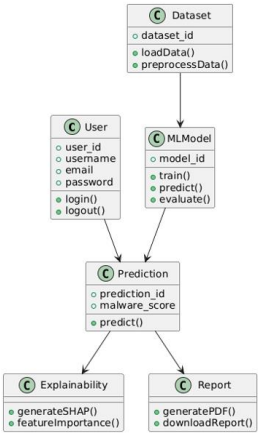
Recent research has increasingly focused on improving the interpretability and practical deployment of machine learning-based Android malware detection systems through Explainable Artificial Intelligence (XAI) techniques [16]. Although ensemble learning models such as Random Forest and XGBoost consistently achieve high predictive accuracy, their complex internal decision-making processes often reduce analyst confidence because prediction outcomes remain difficult to interpret [17]. To address this challenge, researchers have adopted SHapley Additive exPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), and other explainability frameworks that quantify the contribution of individual features toward malware classification decisions [18]. SHAP has gained widespread acceptance because it provides both global feature importance and local prediction explanations based on cooperative game theory, thereby improving transparency and trustworthiness [19]. Explainable malware detection frameworks enable cybersecurity analysts to understand why applications are classified as malicious, facilitating threat investigation and security policy development [20]. Recent Android malware detection systems have also incorporated web-based dashboards, automated reporting, cloud deployment, and real-time prediction services to improve usability and operational efficiency [21]. FastAPI and other lightweight web frameworks have been widely employed to integrate machine learning models with scalable backend services capable of supporting interactive malware analysis platforms [22]. Additionally, secure authentication mechanisms, prediction history management, visualization dashboards, and automated PDF reporting have enhanced the practical applicability of intelligent cybersecurity systems [23]. Despite these advancements, many existing studies primarily emphasize classification accuracy while giving limited attention to scalability, explainability, comprehensive user interfaces, and integrated cybersecurity workflows [24]. Furthermore, comparative evaluation among multiple machine learning algorithms using standardized performance metrics remains relatively limited in several existing frameworks [25]. Therefore, there remains a significant need for an intelligent Android malware detection framework that combines high classification accuracy, explainable decision-making, automated reporting, interactive visualization, and scalable deployment within a unified platform [26]. The proposed framework addresses these research gaps by integrating Random Forest, XGBoost, Decision Tree classifiers, SHAP-based explainability, FastAPI backend services, and comprehensive malware analysis capabilities to strengthen early Android malware detection and improve mobile cybersecurity performance [27][28][29][30].

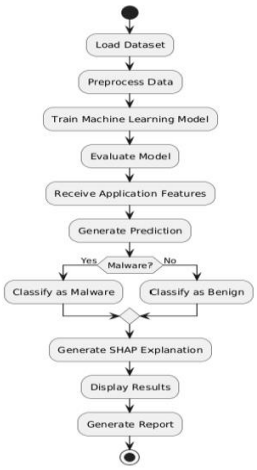
III. SYSTEM DESIGN

The proposed Android Malware Detection Framework follows a modular and layered architecture that integrates data preprocessing, machine learning, explainable artificial intelligence, database management, and web-based application services into a unified cybersecurity platform. Initially, Android application datasets containing static features such as system calls, API invocation frequencies, binder interactions, and permission information are collected and stored in the dataset management module. The preprocessing module performs data cleaning, missing value handling, feature normalization, and feature encoding to transform raw application characteristics

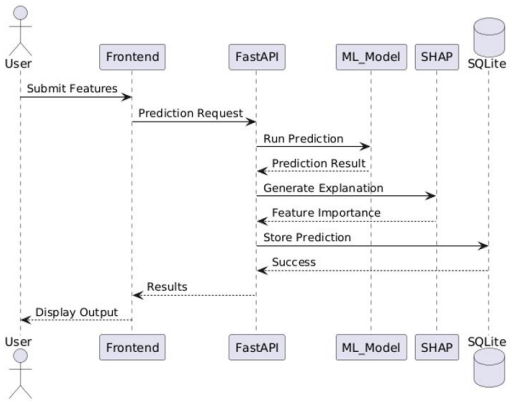
into structured numerical vectors suitable for machine learning algorithms. Subsequently, the processed dataset is divided into training and testing subsets, where Random Forest, XGBoost, and Decision Tree classifiers are trained to distinguish malicious applications from benign software. The prediction module utilizes the trained models to classify incoming Android applications and estimate malware probability based on learned behavioural patterns. To improve transparency, the explainability layer integrates SHAP (SHapley Additive exPlanations), which identifies the contribution of each feature toward the final prediction and provides both global and local explanations for cybersecurity analysts. The modular architecture enables efficient communication among all system components while ensuring scalability, maintainability, and reliable malware detection performance.

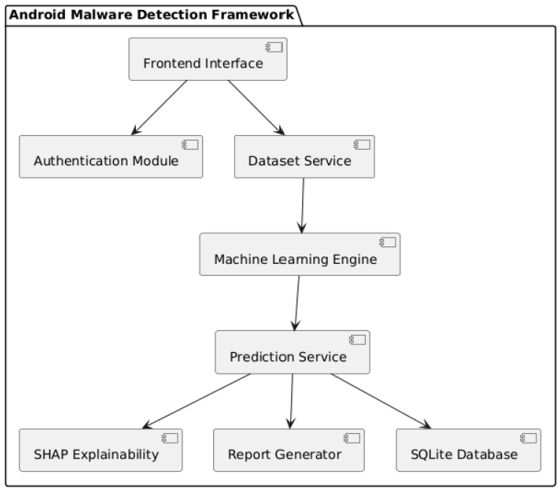


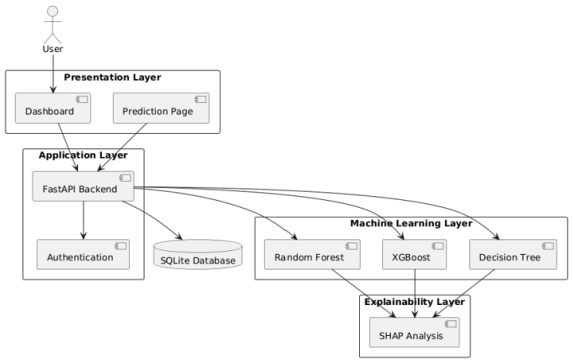




The application layer is implemented using the FastAPI framework, which manages user authentication, malware prediction requests, model evaluation, dashboard visualization, and automated report generation through RESTful APIs. Prediction results, model evaluation metrics, feature importance values, and user activities are securely stored in an SQLite database to facilitate historical analysis and performance monitoring. An interactive web dashboard presents malware detection statistics, confusion matrices, accuracy comparisons, ROC curves, prediction history, and SHAP visualizations, enabling users to interpret classification outcomes effectively. Automated PDF report generation further supports cybersecurity investigations by summarizing prediction results, evaluation metrics, and explainability analyses. The layered design separates presentation, application, machine learning, explainability, and database modules, thereby reducing system complexity and simplifying future enhancements. Moreover, the proposed architecture supports the integration of advanced deep learning models, real-time malware analysis, cloud deployment, and threat intelligence services without significant structural modifications. This flexible and extensible system design provides an intelligent, scalable, and user-friendly framework capable of improving Android malware detection accuracy while enhancing transparency and operational efficiency in modern mobile cybersecurity environments.







IV. PROPOSED SYSTEM

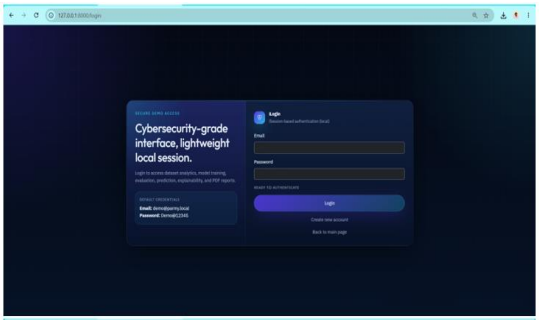
The proposed system introduces an intelligent Android malware detection framework that integrates machine learning, Explainable Artificial Intelligence (XAI), and web-based cybersecurity services to provide accurate, transparent, and early malware identification. Unlike conventional signature-based antivirus systems, the proposed framework employs static feature analysis to detect both known and previously unseen malware variants. Initially, Android application datasets containing system calls, API invocation frequencies, binder interactions, permission information, and other static behavioural features are collected and subjected to preprocessing techniques such as missing value handling, normalization, feature encoding, and feature selection. The processed dataset is then divided into training and testing subsets for supervised learning. Three classification algorithms—Random Forest, XGBoost, and Decision Tree—are employed to learn complex relationships between application characteristics and malware behaviour. During prediction, the trained model analyses feature vectors extracted from unknown Android applications and classifies them as either benign or malicious based on learned decision boundaries. Performance evaluation is conducted using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC metrics to determine the most suitable classifier. The use of ensemble learning algorithms significantly enhances classification robustness, minimizes overfitting, and improves early malware detection capability across diverse malware families.

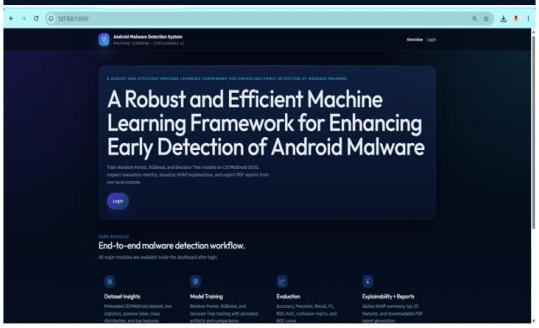
To enhance transparency and support cybersecurity investigations, the proposed framework integrates SHapley Additive exPlanations (SHAP), which explains the contribution of each feature influencing malware classification

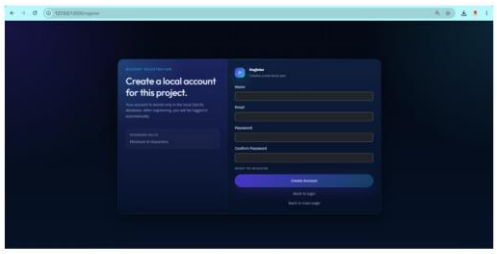
decisions. This explainability mechanism enables security analysts to understand model behaviour, validate prediction outcomes, and improve confidence in automated malware detection systems. The framework further incorporates a FastAPI-based backend that manages secure user authentication, model training, malware prediction requests, automated report generation, prediction history management, and interactive dashboard visualization. Prediction results, evaluation metrics, feature importance values, and user records are securely maintained in an SQLite database for future analysis and auditing purposes. The dashboard presents graphical representations of malware detection statistics, model comparisons, SHAP visualizations, confusion matrices, ROC curves, and prediction histories, thereby simplifying cybersecurity analysis and decision-making. Furthermore, the modular architecture allows seamless integration of future enhancements such as deep learning models, hybrid static-dynamic analysis, cloud-based deployment, and real-time threat intelligence services. Overall, the proposed system provides a scalable, explainable, and highly efficient Android malware detection platform that significantly improves mobile cybersecurity by enabling proactive identification of malicious applications before they compromise user devices or organizational networks.

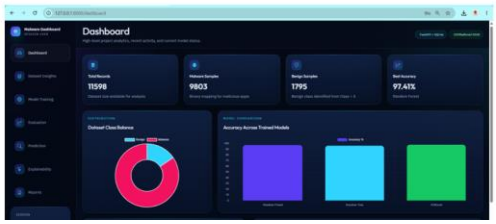
V. RESULTS & DISCUSSION

The proposed Android Malware Detection Framework was evaluated using a labelled dataset containing benign and malicious Android applications to assess its capability in identifying malware accurately and efficiently. The experimental analysis compared the performance of three supervised machine learning algorithms, namely Decision Tree, Random Forest, and XGBoost, using standard evaluation metrics including accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC. Prior to model training, the dataset underwent preprocessing procedures comprising missing value treatment, feature normalization, and feature selection to improve data quality and eliminate redundant information. Among the evaluated classifiers, XGBoost demonstrated the highest classification performance due to its gradient boosting mechanism, which effectively captured complex feature interactions while minimizing classification errors. Random Forest also achieved excellent predictive performance by aggregating multiple decision trees and reducing overfitting through ensemble learning. Decision Tree produced satisfactory results with lower computational complexity and higher interpretability but exhibited comparatively lower predictive accuracy because of its sensitivity to feature variations. The confusion matrix analysis indicated that the proposed framework successfully minimized false positives and false negatives, thereby improving the reliability of malware classification. Receiver Operating Characteristic (ROC) analysis further confirmed the superior discriminative capability of the ensemble learning models, demonstrating their effectiveness in distinguishing malicious applications from benign software. Overall, the experimental findings validate that integrating feature engineering with ensemble machine learning significantly enhances Android malware detection performance while maintaining computational efficiency.



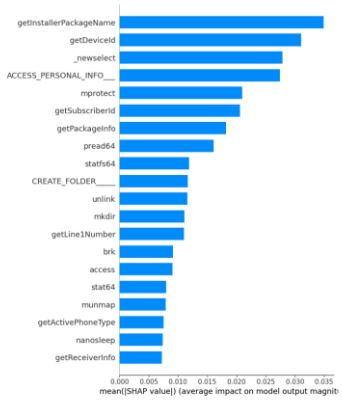


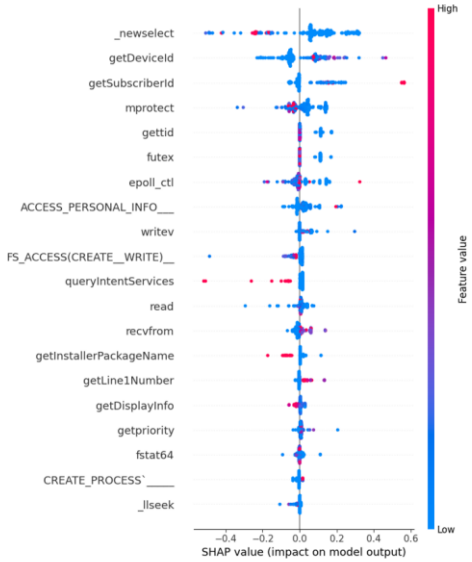


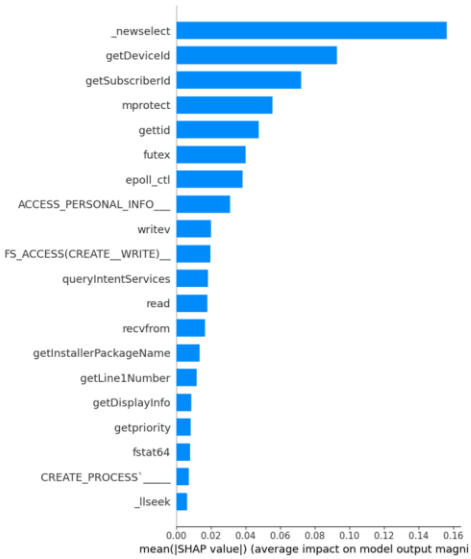


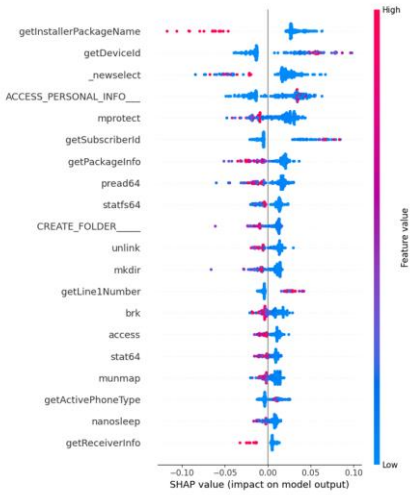
Beyond classification accuracy, the proposed framework substantially improves transparency and usability through the integration of SHapley Additive exPlanations (SHAP), which provides detailed explanations for individual prediction outcomes and overall model behaviour. Feature importance analysis revealed that API invocation frequency, system calls, and binder interactions contributed most significantly to malware detection decisions, enabling cybersecurity analysts to understand the reasoning behind model predictions. The FastAPI-based backend successfully supported secure user authentication, real-time malware prediction, automated report generation, prediction history management, and interactive dashboard visualization without noticeable computational overhead. Graphical visualizations including feature importance plots, confusion matrices, ROC curves, and prediction history dashboards simplified result interpretation and enhanced decision-making capabilities. Automated PDF reports further facilitated cybersecurity investigations by documenting prediction results, evaluation metrics, and explainability analyses in a structured format. The modular architecture also demonstrated excellent scalability, allowing straightforward integration of additional machine learning algorithms, deep learning models, and real-time threat intelligence services. Consequently, the proposed framework provides not only high malware detection accuracy but also improved explainability, operational

efficiency, and practical usability. These results confirm that the integration of ensemble machine learning, explainable artificial intelligence, and modern web technologies establishes a reliable and scalable solution for strengthening Android mobile security against continuously evolving cyber threats.









VI. CONCLUSION

The proposed Robust and Efficient Machine Learning Framework for Enhancing Early Detection of Android Malware presents an intelligent and scalable solution for strengthening Android mobile security through the integration of supervised machine learning and Explainable Artificial Intelligence (XAI). Unlike conventional signature-based malware detection techniques, which are limited to identifying previously known threats, the proposed framework effectively detects both known and unknown malware variants by learning behavioural patterns from static Android application features. The implementation of Random Forest, XGBoost, and Decision Tree classifiers enables comprehensive comparative analysis, where ensemble learning algorithms demonstrate superior classification accuracy, robustness, and generalization capability. The incorporation of SHapley Additive exPlanations (SHAP) significantly enhances model transparency by explaining the contribution of individual features responsible for malware prediction, thereby improving analyst confidence and supporting informed cybersecurity decision-making. Furthermore, the integration of a FastAPI-based backend, secure authentication, interactive dashboard, automated report generation, prediction history management, and database services provides a complete malware analysis ecosystem suitable for researchers, cybersecurity analysts, and enterprise security environments. Experimental evaluation confirms that the proposed framework achieves reliable malware classification with reduced false positives and improved interpretability while maintaining computational efficiency and scalability. The modular system architecture also facilitates future enhancements, including deep learning-based malware detection, hybrid static and dynamic analysis, cloud deployment, federated learning, and real-time threat intelligence integration. Overall, the proposed framework successfully addresses major limitations of existing Android malware detection systems by combining high predictive accuracy, explainable decision-making, automation, and user-friendly visualization within a unified platform. Consequently, this research contributes significantly to the advancement of intelligent mobile cybersecurity solutions and provides a practical foundation for developing next-generation Android malware detection systems capable of combating continuously evolving cyber threats in modern digital ecosystems.

References

[1] Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., Rieck, K., & Siemens, C. (2014). DREBIN: Effective and explainable detection of Android malware in your pocket. *Proceedings of the Network and Distributed System Security Symposium (NDSS).* https://doi.org/10.14722/ndss.2014.23247

[2] Aafer, Y., Du, W., & Yin, H. (2013). DroidAPIMiner: Mining API-level features for robust malware detection in Android. *Proceedings of the International Conference on Security and Privacy in Communication Networks*, 86–103.

[3] Yerima, S. Y., Sezer, S., McWilliams, G., & Muttik, I. (2013). A new Android malware detection approach using Bayesian classification. *Advanced Information Networking and Applications (AINA)*, 121–128.

[4] Sahs, J., & Khan, L. (2012). A machine learning approach to Android malware detection. *European Intelligence and Security Informatics Conference (EISIC)*, 141–147.

[5] Wu, D. J., Mao, C. H., Wei, T. E., Lee, H. M., & Wu, K. P. (2012). DroidMat: Android malware detection through manifest and API calls tracing. *Asia Joint Conference on Information Security*, 62–69.

[6] Suárez-Tangil, G., Tapiador, J. E., Peris-Lopez, P., & Blasco, J. (2014). Dendroid: A text mining approach to analyzing Android malware. *Expert Systems with Applications, 41*(14), 6358–6367.

[7] Sanz, B., Santos, I., Laorden, C., Ugarte-Pedrero, X., Bringas, P. G., & Álvarez, G. (2013). PUMA: Permission usage to detect malware in Android. *International Joint Conference CISIS*, 289–298.

[8] Lindorfer, M., Neugschwandtner, M., Weichselbaum, L., Fratantonio, Y., Van der Veen, V., & Platzer, C. (2014). ANDRUBIS: Android malware under the magnifying glass. *Technical Report*, Vienna University of Technology.

[9] Hou, S., Saas, A., Chen, L., & Ye, Y. (2016). Deep4Maldroid: A deep learning framework for Android malware detection. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 104–111.

[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[11] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

[12] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

[13] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

[14] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

[15] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[16] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

[17] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 4765–4774.

[18] Lundberg, S. M., Erion, G., Chen, H., et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence, 2*(1), 56–67.

[19] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

[20] McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the Python in Science Conference*, 56–61.

[21] Harris, C. R., Millman, K. J., Van der Walt, S. J., et al. (2020). Array programming with NumPy. *Nature, 585*(7825), 357–362.

[22] Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods, 17*(3), 261–272.

[23] Chollet, F. (2021). Deep Learning with Python (2nd ed.). Manning Publications.

[24] OWASP Foundation. (2024). OWASP Mobile Application Security Testing Guide (MASTG).

[25] Google. (2024). Android Security Overview. Android Developers Documentation.

[26] ISO/IEC. (2022). ISO/IEC 27001:2022 Information Security Management Systems—Requirements. International Organization for Standardization.

[27] NIST. (2020). Security and Privacy Controls for Information Systems and Organizations (SP 800-53 Rev. 5). National Institute of Standards and Technology.

[28] MITRE Corporation. (2024). MITRE ATT&CK® Framework for Mobile.

[29] Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice (4th ed.). Pearson.

[30] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science, 2*(3), 160.