# ANDROID MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES

S. Saiful Islam
*KV SubbaReddy Engineering College, Kurnool, A.P, India*
J. Lokesh Reddy, S. Suhel Ahmad, Sk. Farhan, G Emmanuel Raju
*KV SubbaReddy Engineering College, Kurnool, A.P, India*

***Abstract***: Current technological advancement in computer systems has transformed the lives of humans from real to virtual environments. Malware is unnecessary software that is often utilized to launch cyberattacks. Malware variants are still evolving by using advanced packing and obfuscation methods. These approaches make malware classification and detection more challenging. New techniques that are different from conventional systems should be utilized for effectively combating new malware variants. Machine learning (ML) methods are ineffective in identifying all complex and new malware variants. The deep learning (DL) method can be a promising solution to detect all malware variants. This project presents an Automated Android Malware Detection using Optimal Ensemble Learning Approach for Cybersecurity (AAMDOELAC) technique. The major aim of the AAMD-OELAC technique lies in the automated classification and identification of Android malware. To achieve this, the AAMD- OELAC technique performs data preprocessing at the preliminary stage. For the Android malware detection process, the AAMD-OELAC technique follows an ensemble learning process using three ML models, namely Least Square Support Vector Machine (LS- SVM), kernel extreme learning machine (KELM), and Regularized random vector functional link neural network (RRVFLN). Finally, the hunter-prey optimization (HPO) approach is exploited for the optimal parameter tuning of the three DL models, and it helps accomplish improved malware detection results. To denote the supremacy of the AAMD-OELAC method, a comprehensive experimental analysis is conducted.

**Keywords** : Android Malware Detection, Ensemble Learning, Deep Learning, Hunter-Prey Optimization, Cybersecurity

## I. INTRODUCTION

In our quest to fortify the static detection of Android malware, we introduce a novel subset of features meticulously curated from diverse categories. These seven additional feature sets are carefully selected for their potency in discerning between benign and malicious Android applications. Our feature selection spans permissions, API call sequences, manifest attributes, resources, code characteristics, behavioral patterns, and metadata. Each feature set contributes unique insights into the application's behavior, empowering our detection model to scrutinize every facet of an app's structure and operation.

To evaluate the robustness of our approach, we employ a vast dataset comprising over 500,000 Android applications, encompassing both benign and malicious samples. Notably, our dataset boasts the largest malware sample set compared to existing methodologies, ensuring comprehensive coverage of contemporary threats. We meticulously curate this dataset to include samples collected over recent years, imbuing our model with time-awareness and relevancy to the latest malware trends. Rigorous cross-validation techniques and performance metrics such as precision, recall, F1-score, and ROC-AUC are employed to assess the stability and efficacy of our approach.

In tandem with our feature-rich framework, we deploy six classifier models encompassing a spectrum of machine learning algorithms. Decision Trees, Random Forests, Support Vector Machines, Logistic Regression, Gradient Boosting Machines, and Neural Networks are among the models harnessed to scrutinize our feature sets and discern malicious from benign applications. Additionally, we employ a Boosting ensemble learning approach, specifically AdaBoost, coupled with a Decision Tree base learner. This ensemble technique amalgamates the predictive prowess of individual models, amplifying our detection capabilities and fortifying our resilience against evolving malware threats. Through this holistic approach, we establish a formidable defense against Android malware, anchored in advanced feature engineering, cutting-edge machine learning, and ensemble learning strategies.

In today's rapidly advancing digital landscape, the threat of cyberattacks has become more sophisticated and pervasive than ever before. As mobile technology flourishes, Android—the world's most widely used mobile

operating system—has become a prime target for cybercriminals who exploit vulnerabilities to deliver malicious payloads. These malware threats can compromise user privacy, steal sensitive data, damage devices, and even create large- scale botnets. With the exponential increase in Android application downloads and the dependency of users on mobile services, ensuring cybersecurity has never been more critical.

Cybersecurity is no longer a luxury; it is a necessity. The ability to detect and prevent malicious behavior in real-time plays a vital role in maintaining the integrity, confidentiality, and availability of digital systems. Traditional anti-virus methods often fall short in detecting new and obfuscated threats, necessitating the integration of intelligent, automated systems powered by machine learning.

**Android Malware Detection (AMD)** is designed to address this challenge by providing an intelligent, scalable, and proactive approach to Android malware detection. By analyzing URLs and leveraging multiple machine learning models, this system enhances the ability to identify and mitigate threats before they cause harm. It ensures safe browsing and protects both users and infrastructures from potential cyberattacks.

## II. LITERATURE SURVEY

AUTOMATED ANDROID MALWARE DETECTION USING OPTIMAL ENSEMBLE LEARNING APPROACH FOR CYBER SECURITY

Current technological advancement in computer systems has transformed the lives of humans from real to virtual environments. Malware is unnecessary software that is often utilized to launch cyberattacks. Malware variants are still evolving by using advanced packing and obfuscation methods. These approaches make malware classification and detection more challenging. New techniques that are different from conventional systems should be utilized for effectively combating new malware variants. Machine learning (ML) methods are ineffective in identifying all complex and new malware variants. The deep learning (DL) method can be a promising solution to detect all malware variants. This paper presents an Automated Android Malware Detection using Optimal Ensemble Learning Approach for Cybersecurity (AAMDOELAC) technique. The major aim of the AAMD-OELAC technique lies in the automated classification and identification of Android malware. To achieve this, the AAMD-OELAC technique performs data preprocessing at the preliminary stage. For the Android malware detection process, the AAMD-OELAC technique follows an ensemble learning process using three ML models, namely Least Square Support Vector Machine (LS- SVM), kernel extreme learning machine (KELM), and Regularized random vector functional link neural network (RRVFLN). Finally, the hunter-prey optimization (HPO) approach is exploited for the optimal parameter tuning of the three DL models, and it helps accomplish improved malware detection results. To denote the supremacy of the AAMD-OELAC method, a comprehensive experimental analysis is conducted. The simulation results portrayed the supremacy of the AAMD-OELAC technique over other existing approaches.

YOU ARE WHAT THE PERMISSIONS TOLD ME! ANDROID MALWARE DETECTION BASED ON HYBRID TACTICS

Recent years have witnessed a significant increase in the use of Android devices in many aspects of our life. However, users can download Android apps from third-party channels, which provides numerous opportunities for malware. Attackers utilize unsolicited permissions to gain access to the sensitive private intelligence of users. Since signature-based antivirus solutions no longer meet practical needs, efficient and adaptable solutions are desperately needed, especially in new variants. As a remedy, propose a hybrid Android malware detection approach that combines dynamic and static tactics. firstly, adopt static analysis inferring different permission usage patterns between malware and benign apps based on the machine-learning-based method. To classify the suspicious apps further, extract the object reference relationships from the memory heap to construct a dynamic feature base. then present an improved state-based algorithm based on DAMBA. Experimental results on a real-world dataset of 21,708 apps show that our approach outperforms the well-known detector with 97.5% F1-measure.

2 METAHEURISTICS WITH DEEP LEARNING MODEL FOR CYBERSECURITY AND ANDROID MALWARE DETECTION AND CLASSIFICATION

Since the development of information systems during the last decade, cybersecurity has become a critical concern

for many groups, organizations, and institutions. Malware applications are among the commonly used tools and tactics for perpetrating a cyberattack on Android devices, and it is becoming a challenging task to develop novel ways of identifying them. There are various malware detection models available to strengthen the Android operating system against such attacks. These malware detectors categorize the target applications based on the patterns that exist in the features present in the Android applications. As the analytics data continue to grow, they negatively affect the Android defense mechanisms. Since large numbers of unwanted features create a performance bottleneck for the detection mechanism, feature selection techniques are found to be beneficial. This work presents a Rock Hyrax Swarm Optimization with deep learning-based Android malware detection (RHSODL-AMD) model.
The technique presented includes finding the Application Programming Interfaces (API) calls and the most significant permissions, which results in effective discrimination between the good ware and malware applications. Therefore, an RHSO based feature subset selection (RHSO-FS) technique is derived to improve the classification results.

A METHOD FOR AUTOMATIC ANDROID MALWARE DETECTION BASED ON STATIC ANALYSIS AND DEEP LEARNING

The computers nowadays are being replaced by the smartphones for the most of the internet users around the world, and Android is getting the most of the smartphone systems' market. This rise of the usage of smartphones generally, and the Android system specifically, leads to a strong need to effectively secure Android, as the malware developers are targeting it with sophisticated and obfuscated malware applications. Consequently, a lot of studies were performed to propose a robust method to detect and classify android malicious software (malware). Some of them were effective, some were not; with accuracy below 90%, and some of them are being outdated; using datasets that became old containing applications for old versions of Android that are rarely used today. In this paper, a new method is proposed by using static analysis and gathering as most useful features of android applications as possible, along with two new proposed features, and then passing them to a functional API deep learning model we made. This method was implemented on a new and classified android application dataset, using 14079 malware and benign samples in total, with malware samples classified into four malware classes. Two major experiments with this dataset were implemented, one for malware detection with the dataset samples categorized into two classes as just malware and benign, the second one was made for malware detection and classification, using all the five classes of the dataset.

MACHINE LEARNING-BASED ADAPTIVE GENETIC ALGORITHM FOR ANDROID MALWARE DETECTION IN AUTO-DRIVING VEHICLES

The growing trend toward vehicles being connected to various unidentified devices, such as other vehicles or infrastructure, increases the possibility of external attacks on "vehicle cybersecurity (VC). Detection of intrusion is a very important part of network security for vehicles such as connected vehicles, that have open connectivity, and self- driving vehicles.

Consequently, security has become an important requirement in trying to protect these vehicles as attackers have become more sophisticated in using malware that can penetrate and harm vehicle control units as technology advances. Thus, ensuring the vehicles and the network are safe is very important for the growth of the automotive industry and for people to have more faith in it. In this study, a machine learning-based detection approach using hybrid analysis- based particle swarm optimization (PSO) and an adaptive genetic algorithm (AGA) is presented for Android malware detection in auto-driving vehicles. The "CCCS-CIC-AndMal-2020" dataset containing 13 different malware categories and 9504 hybrid features was used for the experiments. In the proposed approach, firstly, feature selection is performed by applying PSO to the features in the dataset. In the next step, the performance of XGBoost and random forest (RF) machine learning classifiers is optimized using the AGA.

**III.EXISTING SYSTEM**

The proposed project aims to overcome the limitations of existing systems by employing ensemble learning techniques for Android malware detection. Ensemble learning is a powerful method that combines the predictions of multiple machine learning models (such as decision trees, support vector machines, and neural networks) to produce more accurate and reliable results than any single model can achieve independently.

The proposed system involves the following key steps:

- **Data Collection:** Gather a comprehensive dataset containing both malware and benign Android applications.
- **Feature Extraction:** Analyze application behaviors and extract meaningful features (e.g., API calls, permissions, manifest details).
- **Model Training:** Train multiple base models on the dataset.
- **Ensemble Integration:** Combine the outputs of individual models using techniques like bagging, boosting, or stacking.
- **Performance Evaluation:** Evaluate the system using metrics such as detection accuracy, false positive rate, precision, recall, and computational efficiency.

This approach not only enhances detection performance but also ensures adaptability to evolving threats by leveraging diverse learning patterns from multiple algorithms.

Furthermore, since this system deals with potentially sensitive data, it is important to address **ethical concerns** related to data privacy, user consent, and secure handling of datasets used during training and evaluation.

Disadvantages of the Existing System

- **Lack of Machine Learning and Ensemble Learning Integration:** The existing system does not implement advanced machine learning techniques or ensemble models, which limits its ability to detect complex and evolving malware patterns.
- **No Analysis of Reverse-Engineered Application Characteristics:** The system does not analyze reverse-engineered apps, meaning it fails to consider deeply embedded malicious behavior that may be revealed through decompiled code or structural analysis of APK files.

## IV.PROPOSED SYSTEM

The proposed system introduces a novel approach to the static detection of Android malware by expanding the feature set and incorporating advanced ensemble learning techniques. Unlike traditional methods that rely on limited attributes, this system utilizes seven additional, strategically selected feature sets**.** These features are derived from diverse categories such as permissions, API calls, intent filters, activities, and other manifest-related attributes, offering a more comprehensive understanding of application behavior.

A robust dataset consisting of over 500,000 Android applications—both benign and malicious—was used for model training and evaluation. This dataset is notably larger than those used in many state-of-the-art studies, thereby increasing the model's generalizability and stability across different types of malware.

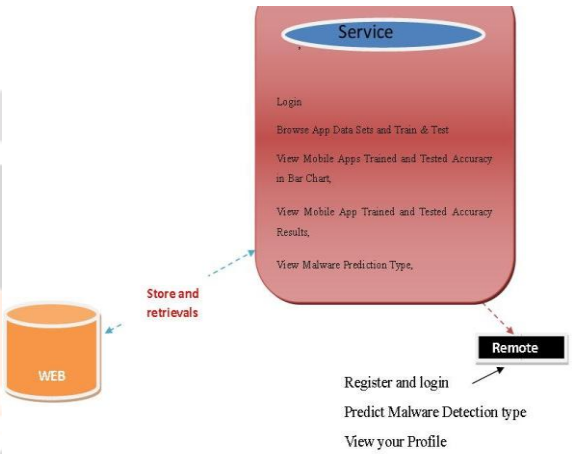
### Key Contributions and Methodology

- **Expanded Feature Engineering:** The system extracts a refined subset of features beyond traditional static attributes. These include recently identified behavioral indicators and structural patterns, which greatly enhance the predictive capabilities of the model.
- **Machine Learning Model Training:** A total of six distinct machine learning classifiers (e.g., Decision Trees, Random Forest, SVM, k-NN, Naive Bayes, and Logistic Regression) were trained using the enriched feature set. These models help identify malware patterns based on binary classification (malicious vs benign).
- **Boosting with AdaBoost Ensemble Technique:** To improve prediction accuracy and minimize false positives, the AdaBoost ensemble learning method was implemented, with Decision Tree as the base learner. AdaBoost enhances weak classifiers by combining them into a strong predictive model, ideal for complex malware classification tasks.
- **Time-Aware and Up-to-Date Dataset:** The training data includes the most recent malware samples**,** incorporating updated Android API levels, ensuring that the system remains relevant and effective against newly emerging threats.

**Target Users and Applications**

This system has wide-ranging applications across different user groups:

- **Cybersecurity Researchers:** For developing and validating malware detection algorithms, especially those focused on **URL-based malware detection** in Android ecosystems.
- **Developers and Security Analysts:** To build and maintain secure mobile applications and app store environments.
- **Educational Institutions:** As a practical implementation of **machine learning in cybersecurity**, useful for student projects, workshops, and academic research.




- **Corporate IT Departments and Organizations:** To monitor employee device usage and proactively detect malicious activity, particularly in Bring Your Own Device (BYOD) setups.

System Features and Benefits

- **Accurate Prediction:** The integration of advanced classifiers and ensemble techniques significantly increases the malware detection rate.
- **Visual Analytics:** The system offers visual insights such as detection charts, confusion matrices, and feature importance graphs, which aid in better understanding and debugging.
- **User-Friendly Web Interface:** Designed for usability, the web interface allows easy interaction with the detection engine for both technical and non-technical users.
- **Contribution to Digital Safety:** The Automated Android Malware Detection (AAMD) system contributes meaningfully to the field of intelligent malware defense, helping reduce cybersecurity threats in the Android ecosystem.

**Core Objective**

To automatically detect malicious URLs and Android applications using machine learning and ensemble techniques, thereby supporting safe browsing**,** malware prevention**,** and cybersecurity awareness for users and organizations alike.

**Fig : Sytem Architecture**

**V. RESULTS**


**Fig : Login Page**


**Fig : Register Page**

**Fig : Dataset File**


**Fig : Malware Prediction Page**

## VI. CONCLUSION

In this project, we developed an intelligent and framework for the detection of malicious Android applications, aimed at enhancing mobile cybersecurity. The system integrates various components of machine learning to analyze, classify, and predict whether a given Android- related URL or application behavior is benign or malicious.

The methodology begins by selecting and engineering meaningful features that represent Android application behavior. Using reverse engineering tools like **AndroGuard**, raw application data is decompiled and converted into a structured set of binary feature vectors. These vectors encapsulate core aspects such as permissions, API calls, intent filters, and network activity—providing the machine learning models with a rich understanding of the app's internal structure and behavior.

Once feature extraction is complete, the system employs **Python-based modules** to preprocess the data. The dataset is cleaned, shuffled, and split into training and testing sets, ensuring unbiased learning. The use of both benign and malicious samples ensures that the models learn to differentiate based on behavioral patterns rather than static signatures. This data is then used to train multiple classifiers including Naive Bayes, Support Vector Machine, Logistic Regression, Decision Trees, and K-Nearest Neighbors—all of which are combined into a robust Voting Classifier ensemble model for final predictions.

One of the most promising aspects of this project is its self-adaptive potential. By continuously learning from new user-submitted URLs and behaviors, the model can be retrained periodically to adapt to emerging threats. This self-evolving architecture is crucial in combating zero-day vulnerabilities and polymorphic malware, which change structure to evade traditional detection methods.

Beyond its technical merits, this project demonstrates how machine learning can effectively support digital safety initiatives. The system bridges a critical gap between static rule-based
antivirus software and intelligent, behavior-based malware detection. It not only detects threats but does so proactively—allowing for earlier intervention and mitigation.

Furthermore, the implementation of this project in a web-based Django environment showcases its practicality. With a clean interface for remote users and administrative dashboards for model training and results analysis, the system is both user-friendly and scalable for real- world application.

In conclusion, this project stands as a significant step toward strengthening mobile application security through automation and artificial intelligence. It lays the groundwork for future development in smart malware detection systems and contributes to the broader mission of cyber resilience in an increasingly mobile-dependent world.

**VIII. REFERENCES**

[1] A. Datta, S. Buchegger, L.-H.Vu, T. Strufe, and K. Rzadca, ‘‘Decentralized online social networks,’’ in *Handbook of Social Network Technologies and Applications*. Springer, 2010, pp. 349–378.

[2] L. Jiang and X. Zhang, ‘‘BCOSN: A blockchain-based decentralized online social network,’’ *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1454–1466, Dec. 2019.

[3] B. Guidi, A. Michienzi, and L. Ricci, ‘‘Steem blockchain: Mining the inner structure of the graph,’’ *IEEE Access*, vol. 8, pp. 210251–210266, 2020.

[4] W. Sherchan, S. Nepal, and C. Paris, ‘‘A survey of trust in social networks,’’ *ACM Comput. Surveys*, vol. 45, no. 4, pp. 1–33, Aug. 2013.

[5] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, ‘‘Estimation of energy consumption in machine learning,’’ *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, Dec. 2019.

[6] B. Guidi, K. G. Kapanova, K. Koidl, A. Michienzi, and L. Ricci, ‘‘The contextual ego network P2P overlay for the next generation social networks,’’ *Mobile Netw. Appl.*, vol. 25, no. 3, pp. 1062–1074, Jun. 2020.

[7] L. Mui, M. Mohtashemi, and A. Halberstadt, ‘‘A computational model of trust and reputation,’’ in *Proc. 35th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2002, pp. 2431–2439.

[8] S. Nepal, W. Sherchan, and C. Paris, ‘‘STrust: A trust model for social networks,’’ in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Nov. 2011, pp. 841– 846.

[9] D. Olmedilla, O. F. Rana, B. Matthews, and W. Nejdl, ‘‘Security and trust issues in semantic grids,’’ in *Semantic Grid: The Convergence of Technologies*, vol. 5271. Schloss Dagstuhl, Germany: Internationales Begegnungsund Forschungszentrum für Informatik (IBFI), 2006, pp. 1–11.

[10] G. Liu, Y.Wang, and M. Orgun, ‘‘Trust inference in complex trust-oriented social networks,’’ in *Proc. Int. Conf. Comput. Sci. Eng.*, Aug. 2009, pp. 996–1001.