Current Trends in Technology and Science

www.ctts.in, ISSN: 2279-0535, May 2025, Volume: 14, Issue:3

# Towards Robust Android Malware Detection: A Hybrid SVM-MLP Framework

Osama Bin Imtiyaz, Pradeep Kumar Pandey, Saurabh Mandloi
School of Computer Science and Technology, SAM Global University, Bhopal

**Abstract-** The rapid proliferation of Android devices has made them a primary target for cybercriminals, leading to a surge in sophisticated mobile malware threats. Traditional signature-based detection methods often fail to detect novel or obfuscated malware variants, such as zero-day attacks, necessitating more intelligent and adaptive solutions. In this context, the present study proposes a hybrid machine learning framework that integrates Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP) to detect Android malware effectively. The model leverages advanced data preprocessing techniques, including handling missing values and categorical encoding, followed by dimensionality reduction through Principal Component Analysis (PCA). This facilitates efficient learning while reducing model complexity and enhancing generalisation capabilities. The proposed system is evaluated on a publicly available Android malware dataset sourced from Kaggle. Performance is assessed using standard metrics such as accuracy, precision, recall, F1-score, specificity, and the Area Under the Curve (AUC). The hybrid SVM-MLP model achieves a remarkable detection accuracy of 98.99% and an AUC of 99.83%, significantly outperforming conventional standalone classifiers and existing benchmark models. Moreover, the system demonstrates resilience in classifying both benign and malicious applications with high precision, showcasing its practical utility in real-world mobile security applications. This research contributes to the growing body of work aimed at enhancing Android security through machine learning. The results suggest that hybrid models combining different classification strategies can provide more robust and scalable solutions for malware detection, especially in the evolving landscape of mobile and IoT-based ecosystems.

**Keywords:** Android Malware, Machine Learning, PCA, SVM, MLP, AUC.

## 1. Introduction

Android malware refers to malicious software specifically crafted to exploit devices running the Android operating system. Its primary objectives are to steal sensitive information, disrupt device functionality, or provide unauthorised access to attackers. For instance, an app titled "AdBlocker" was initially introduced as a utility to block advertisements but was later discovered to be malware itself, ironically bombarding users with more ads and benefitting cybercriminals. The proliferation of Android devices has transformed the way people communicate, access information, and interact with technology. Google's Android OS has played a significant role in this transformation, becoming the most widely used mobile operating system globally. Its adoption spans smartphones, tablets, smart TVs, and wearable devices. However, this popularity has also attracted cyber attackers who seek to exploit its vulnerabilities. The rise in Android malware underscores the urgent need for effective detection mechanisms to ensure user security and privacy. This research seeks to understand the evolving landscape of Android malware, its different forms, propagation methods, and potential impact. The analysis includes the strategies adopted by cybercriminals to develop and disseminate malware. A robust predictive framework is proposed, leveraging machine learning (ML), data analytics, and cybersecurity principles to distinguish between benign and malicious applications. Through in-depth analysis of legitimate and malicious apps, the study aims to extract and learn distinguishing characteristics to enhance malware detection. In the context of the Internet of Things (IoT), Android's significance has further expanded. Many IoT devices, such as smart cameras, wearables, and home automation systems, are powered by Android or Android-based systems. These devices are increasingly becoming targets of sophisticated malware attacks. The growing integration of Android in IoT ecosystems has escalated the challenge of protecting devices and networks from malware threats. To address this, the research also considers an AI and blockchain-based framework for detecting malware on Android-powered IoT devices. This approach is envisioned to combat existing malware and preempt future cyber-attacks through scalable, adaptive, and resource-efficient strategies. A key challenge with traditional malware detection methods is their reliance on signature-based systems, which often fail to detect new or polymorphic malware.

The static nature of these tools makes them ineffective against zero-day threats or rapidly evolving malware variants. As a solution, artificial intelligence, particularly ML and deep learning techniques, offers a promising alternative. These approaches can learn complex patterns from large datasets, identify subtle malicious behaviours, and adapt to emerging threats. This study introduces a hybrid model combining Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP) to improve Android malware detection. The model incorporates Principal Component Analysis (PCA) for dimensionality reduction, effective feature engineering, and robust classification capabilities. By doing so, it aims to provide a scalable, high-

accuracy detection system. The research concludes by positioning this hybrid model as a vital step toward building secure mobile environments and protecting Android users against increasingly sophisticated malware threats.

## 2. Related Work

The exponential growth of Android applications and the corresponding rise in mobile malware have driven significant research into automated malware detection mechanisms. Traditional antivirus tools primarily rely on signature-based detection, which fails to detect obfuscated or zero-day malware. To overcome these limitations, researchers have explored a variety of machine learning (ML) and deep learning (DL) techniques to enhance Android malware detection through static, dynamic, and hybrid analyses. Zhu et al. [1] introduced SEDMDroid, a stacking ensemble framework for static Android malware detection. Their model combines multiple Multi-Layer Perceptrons (MLPs) with a Support Vector Machine (SVM) fusion classifier. The model uses bootstrapped samples and PCA to diversify the training data. However, its low learning rate limited classification performance, resulting in reduced generalisation. Other studies have explored feature-based detection strategies. Kato et al. [3] proposed a method based on the Composition Ratio (CR) of permission pairs, leveraging patterns in permission usage to distinguish benign apps from malware. While effective, this technique requires regular retraining and is susceptible to poisoning attacks. Gong et al. [4] investigated overlay attacks in Android apps, proposing early detection methods during the app market review stage to block potential threats without disrupting legitimate overlay functionality. Several works have also investigated optimisation-based and deep-learning approaches. Almomani et al. [5] introduced a ransomware detection model using Binary Particle Swarm Optimization (BPSO) for hyperparameter tuning and feature selection. Similarly, Mercaldo and Santon [7] applied formal equivalence checking with supervised ML to reduce comparisons and detect malicious behaviours. Deep learning has gained traction for its ability to model complex patterns. Wu and Jung [8] developed AdMat, a CNN-based model that achieved over 97% accuracy even with limited data. Yuan et al. [10] proposed a lightweight on-device detection system using one-shot learning, achieving robustness against adversarial examples. Behaviour-based models such as EveDroid by Lei et al. [20] use event clusters and neural networks to capture high-level semantics from Android app behaviours. This enhances detection beyond the API level. Similarly, Zhang et al. [18] introduced a correlation-based method using abstract API calls, improving the detection of evolving malware that is suffering from high resource consumption. To address real-world deployment challenges, Ribeiro et al. [14] proposed HIDROID, a host-based Intrusion Detection and Prevention System (IDPS) optimised for resource-constrained Android devices. Their method efficiently samples runtime data while minimising CPU and battery usage. Recent advances also include hybrid and blockchain-integrated systems. Kumar et al. [19] combined ML clustering with blockchain to ensure secure malware tracking in IoT-based Android environments. Their framework facilitates traceability and tamper-proof detection pipelines. Despite these advancements, issues like false positives, model drift, and lack of adaptability persist. The proposed hybrid SVM-MLP framework in this paper builds upon these foundations by integrating dimensionality reduction, static feature extraction, and dual classifiers to boost accuracy, adaptability, and computational efficiency.

## 3. Problem Definition and Objective

The Android operating system has achieved unprecedented popularity due to its open-source nature, customizability, and broad adoption across smartphones, tablets, smart TVs, and IoT devices. However, this openness and widespread deployment have made Android a prime target for cyber threats. The decentralised nature of app distribution, the availability of third-party app stores, and inconsistent security enforcement across device manufacturers exacerbate the risks. Consequently, the Android ecosystem has become increasingly vulnerable to malicious applications (malware), ranging from adware and spyware to more severe threats such as trojans, ransomware, and remote access tools. Traditional malware detection systems predominantly rely on signature-based techniques. While effective against known malware, these methods fall short when dealing with polymorphic malware or zero-day threats that exhibit novel behaviours or bypass static signature rules. Moreover, many existing machine learning-based approaches suffer from low precision, high false positive rates, and poor adaptability to evolving malware variants. In addition, models that focus solely on static or dynamic analysis without intelligent feature engineering often fail to capture the nuanced behaviour of sophisticated malware. Given these challenges, there is an urgent need for a lightweight yet effective Android malware detection framework that can generalise well to unknown threats while maintaining high classification accuracy and low computational cost. This research aims to address these gaps by developing a hybrid detection model that combines the strengths of Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP). To enhance model efficiency and reduce dimensional complexity, Principal Component Analysis (PCA) is employed as a feature reduction technique. This ensures that only the most relevant attributes contribute to the classification process, improving accuracy and reducing overfitting. The primary objective of this study is to design a robust, scalable, and resource-efficient malware detection system capable of accurately distinguishing between benign and malicious Android applications. The model should minimise false positives to avoid disrupting legitimate app functionality and maximise detection rates to ensure user safety. Furthermore, the framework should be adaptable to future malware trends and applicable across diverse Android environments,

including mobile and IoT platforms. Through rigorous experimental validation, this work demonstrates the feasibility and effectiveness of the proposed hybrid approach in enhancing Android malware detection and contributing to a safer digital ecosystem.

## 4. Proposed Methodology

Android malware detection remains a highly dynamic and technically challenging domain due to the evolving behaviour of malicious applications and the complexity of differentiating them from legitimate software. This research introduces a novel and scalable hybrid model that integrates Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP) to achieve enhanced accuracy in malware classification. The methodology relies on a structured sequence of operations, beginning with data acquisition and concluding with model evaluation using multiple performance metrics. The overall workflow and system architecture are designed to ensure high precision, robustness, and efficiency, even in resource-constrained environments like mobile and IoT devices.

### 4.1 Workflow Overview

The workflow of the proposed model is divided into five major components, each essential for ensuring a high-performing and scalable malware detection framework:

**Data Source:** The dataset used in this study is sourced from **Kaggle**, a reputable platform for open-access datasets and machine learning competitions. The Android malware dataset includes a comprehensive set of application metadata, permissions, API call logs, and behavioural indicators for both benign and malicious applications. The rich feature set enables in-depth learning and improved model generalisation.

**Preprocessing:** Before model training, preprocessing is applied to ensure data quality and consistency. Missing values are identified and addressed using imputation strategies such as mean or median replacement, depending on the feature distribution. Categorical attributes such as labels (malicious or benign) are encoded using **label encoding** to convert them into numerical values that are compatible with ML algorithms. Data normalisation and scaling are also performed to ensure uniformity and improve learning stability.

**Feature Selection**: To enhance model efficiency and avoid overfitting, **Principal Component Analysis (PCA)** is used for dimensionality reduction. PCA transforms the original features into a set of linearly uncorrelated components, preserving most of the variance while reducing computational complexity. This step not only speeds up training but also improves classification accuracy by eliminating noise and redundant attributes.

**Classification:** In the proposed methodology, two machine learning classifiers, Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP), are deployed in tandem to enhance detection accuracy. SVM is utilised to construct optimal hyperplanes that effectively separate malware from benign applications within a high-dimensional feature space. Its strength lies in solving binary classification problems where clear decision boundaries can be identified, making it particularly suitable for distinguishing between two well-defined classes. On the other hand, MLP, a type of feedforward artificial neural network, is employed to model complex, non-linear patterns that may not be separable using linear techniques alone. Comprising an input layer, one or more hidden layers, and an output layer, the MLP learns from data by adjusting its weights through backpropagation to minimise classification errors. By integrating SVM's ability to handle linearly separable data with MLP's capacity to learn intricate, non-linear relationships, the hybrid approach achieves a more comprehensive understanding of the dataset. This complementary deployment of SVM and MLP results in a more robust and accurate malware detection model capable of generalising well to previously unseen and obfuscated threats.

**Evaluation Metrics:** to assess the performance of the proposed model, several standard evaluation metrics are used, including accuracy, which is a proportion of correctly classified instances. Precision is a proportion of true positives among predicted positives. Recall (sensitivity) is the ability of the model to detect all positive instances. F1-score is a harmonic mean of precision and recall, useful for imbalanced datasets. Auc (area under the curve) is a measure of the model's ability to distinguish between classes across different threshold values. These metrics ensure a comprehensive understanding of the classifier's behaviour across both majority and minority classes.

### 4.2 Architecture Diagram

The architecture of the proposed model is designed as a modular pipeline that integrates multiple stages, each contributing to the system's overall accuracy and efficiency. The architecture, as illustrated in **Figure 1**, consists of the following interconnected modules:

**Input Layer (Raw Data Ingestion)** The pipeline begins with the ingestion of raw data from the Kaggle Android malware dataset. This includes application attributes, permissions, intents, and behavioural logs.

**Data Cleaning and Transformation Module:** This module performs preprocessing tasks such as missing value imputation, label encoding, normalisation, and outlier detection. The cleaned dataset is then passed forward for dimensionality reduction.

**Dimensionality Reduction Module (PCA):** PCA is applied to project the high-dimensional input features into a lower-dimensional subspace while preserving essential

variance. This step filters noise and improves downstream classification efficiency.

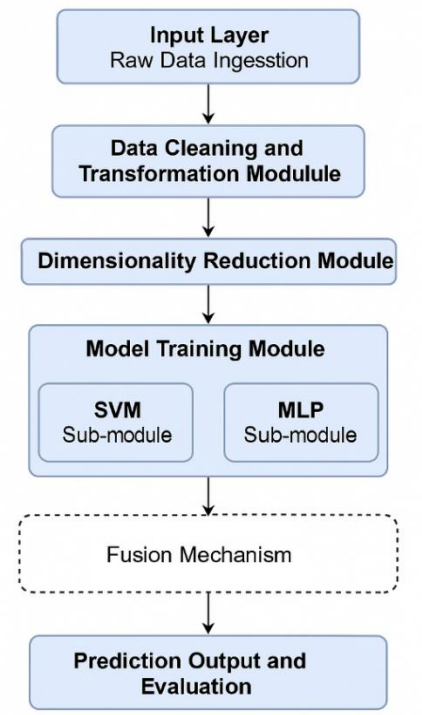


**Figure 1:** Hybrid architecture for Android malware detection pipeline.

**Model Training Module: SVM Sub-module:** Constructs optimal hyperplane using support vectors. It excels at capturing well-separated classes with fewer samples. **MLP Sub-module:** Processes the PCA-transformed input through multiple fully connected layers. Activation functions such as ReLU are used to introduce non-linearity, and backpropagation is used for weight optimisation.

**Fusion Mechanism (Optional Ensemble):** Although SVM and MLP can be used independently, a soft voting or averaging mechanism may be employed to combine predictions from both classifiers for improved robustness.
**Prediction Output and Evaluation:** The final predictions are generated, and performance is assessed using the metrics mentioned above. The confusion matrix is also generated to visualise true/false positives and negatives.

## 5. Experimental Setup

The experimental evaluation of the proposed hybrid SVM-MLP framework was conducted using Python programming language, with a focus on libraries such as NumPy, Pandas, Scikit-learn, and TensorFlow for data manipulation, model building, and evaluation. The implementation and code execution were carried out in the Spyder Integrated Development Environment (IDE), which provides a user-friendly interface suitable for scientific computing and machine learning workflows. For model training and evaluation, the dataset was partitioned into two subsets: 70% of the data was allocated for training the model, while the remaining 30% was used for testing. This split ensures that the model is trained in a sufficient amount of data and validated on a diverse sample to assess its generalisation capabilities. Before applying dimensionality reduction techniques, data normalisation was performed to scale the features into a consistent range. This step is particularly important to improve the performance of both Principal Component Analysis (PCA) and machine learning classifiers by ensuring that each feature contributes equally during training. Model training was conducted separately for SVM and MLP classifiers to evaluate their effectiveness. Subsequently, a hybrid configuration was implemented by combining the predictions of both classifiers. This dual approach allowed for a comparative and integrated analysis of performance under independent and collaborative settings. Grid search and cross-validation techniques were employed to optimise the model parameters and enhance prediction accuracy. Grid search systematically tested different combinations of hyperparameters, while cross-validation ensured robustness by evaluating the model's consistency across multiple data folds. This experimental setup enabled a thorough and reliable performance assessment of the proposed malware detection framework.

## 6. Results and Discussion

The performance of the proposed hybrid malware detection framework was rigorously evaluated using multiple classification metrics to compare the effectiveness of the individual classifiers—Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). These metrics include accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC), which collectively provide a comprehensive assessment of the model's performance on the test dataset. The experimental results indicate that both classifiers are capable of achieving high detection performance, but the MLP slightly outperforms the SVM across all evaluated metrics. The MLP achieved an impressive accuracy of 98.99%, precision of 99.00%, recall of 98.00%, F1-score of 98.00%, and an AUC of 99.83%. In comparison, SVM obtained an accuracy of 97.54%, precision of 97.99%, recall of 96.01%, F1-score of 97.01%, and an AUC of 99.44%. These results are summarised in Table 1.

Table 1. Performance Comparison of SVM and MLP Models

| Metric | SVM (%) | MLP (%) |
|---|---|---|
| Accuracy | 97.54 | 98.99 |
| Precision | 97.99 | 99.00 |
| Recall | 96.01 | 98.00 |
| F1-Score | 97.01 | 98.00 |
| AUC | 99.44 | 99.83 |

The superior performance of the MLP can be attributed to its ability to learn complex non-linear relationships in the dataset, especially after feature transformation using Principal Component Analysis (PCA). SVM, while effective in linearly separable spaces, showed slightly lower recall, indicating a few more false negatives compared to the MLP. These results validate the effectiveness of deep learning-based models in static Android malware detection and highlight the advantage of combining machine learning techniques for robust performance. The hybrid framework offers promising prospects for deployment in real-world Android security systems and mobile application vetting pipelines (Zhu et al., 2021; Wu & Jung, 2021).

**Conclusion and Future Work**

This research presents a hybrid machine learning framework that integrates Support Vector Machines (SVM) and Multi-Layer Perceptrons (MLP) for robust Android malware detection. By combining the linear classification strength of SVM with the non-linear learning capabilities of MLP, the model effectively captures diverse patterns within the data. The implementation of Principal Component Analysis (PCA) for dimensionality reduction ensures that the most relevant features contribute to the classification process, thereby enhancing both accuracy and computational efficiency. Experimental results on a publicly available Android malware dataset demonstrate the effectiveness of the proposed model, with the MLP achieving a maximum accuracy of 98.99% and an AUC of 99.83%. These results significantly outperform many existing detection techniques, particularly in terms of precision, recall, and generalisation capability. The hybrid approach not only reduces false positives but also improves detection rates for previously unseen threats, making it suitable for real-world deployment. This study confirms that hybrid classifiers supported by intelligent feature reduction can offer scalable, adaptive, and high-precision malware detection solutions. Future work will focus on expanding this framework to incorporate real-time dynamic analysis and deploying it in resource-constrained environments such as IoT devices, ensuring broader protection against emerging mobile threats.

**References**

[1]. H. Zhu, Y. Li, R. Li, J. Li, Z. You and H. Song, “SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection” in IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 984-994, 1 April-June 2021.

[2]. A. Alzubaidi, “Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review” in IEEE Access, vol. 9, pp. 146318-146349, 2021.

[3]. H. Kato, T. Sasaki and I. Sasase, “Android Malware Detection Based on Composition Ratio of Permission Pairs” in IEEE Access, vol. 9, pp. 130006-130019, 2021.

[4]. C. Li, X. Chen, D. Wang, S. Wen, M. Ejaz Ahmed, S. Camte “Backdoor Attack on Machine Learning Based Android Malware Detectors” in IEEE Transactions on Dependable and Secure Computing.

[5]. L. Gong, Z. Li, H. Wang, H. Lin, X. Ma, and Y. Liu, “Overlay-based Android Malware Detection at Market Scales: Systematically Adapting to the New Technological Landscape” in IEEE Transactions on Mobile Computing.

[6]. I. Almomani, R. Qaddoura, M. Habib, S. Alsoghyer, A. Khayer “ “Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data” in IEEE Access, vol. 9, pp. 57674-57691, 2021.

[7]. F. Mercaldo and A. Santone, “Formal Equivalence Checking for Mobile Malware Detection and Family Classification” in IEEE Transactions on Software Engineering.

[8]. L. N. Vu and S. Jung, “AdMat: A CNN-on-Matrix Approach to Android Malware Detection and Classification” in IEEE Access, vol. 9, pp. 39680-39694, 2021.

[9]. L. Gong, H. Lin, Z. Li, F. Qian, Y. Li, X. Ma, Y. Liu “Systematically Landing Machine Learning onto Market-Scale Mobile Malware Detection,” in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1615-1628, 1 July 2021.

[10]. W. Yuan, Y. Jiang, H. Li and M. Cai, “A Lightweight On-Device Detection Method for Android Malware” in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5600-5611, Sept. 2021.

[11]. K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, “A Review of Android Malware Detection Approaches Based on Machine Learning” in IEEE Access, vol. 8, pp. 124579124607, 2020.

[12]. D. Li and Q. Li, “Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware Detection” in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3886-3900, 2020.

[13]. Q. Han, V. S. Subrahmanian and Y. Xiong, “Android Malware Detection via (Somewhat) Robust Irreversible Feature Transformations” in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3511-3525, 2020.

[14]. J. Ribeiro, F. B. Saghezchi, G. Mantas, J. Rodriguez and R. A. Abd-Alhameed, “HIDROID: Prototyping a Behavioural Host-Based Intrusion Detection and Prevention System for Android” in IEEE Access, vol. 8, pp. 23154-23168, 2020.

[15]. X. Wang, C. Li and D. Song, “CrowdNet: Identifying Large-Scale Malicious Attacks Over Android Kernel Structures” in IEEE Access, vol. 8, pp. 15823-15837, 2020.

[16]. Y. Zhang, Y. Sui, S. Pan, Z. Zheng, B. Ning, I. Tsang, W. Zhou "Familial Clustering for Weakly-Labelled Android Malware Using Hybrid Representation Learning" in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3401-3414, 2020.

[17]. S. Aonzo, A. Merlo, M. Migliardi, L. Oneto and F. Palmieri, "Low-Resource Footprint, Data-Driven Malware Detection on Android" in IEEE Transactions on Sustainable Computing, vol. 5, no. 2, pp. 213-222, 1 April-June 2020.

[18]. H. Zhang, S. Luo, Y. Zhang and L. Pan, "An Efficient Android Malware Detection System Based on Method-Level Behavioural Semantic Analysis" in IEEE Access, vol. 7, pp. 69246-69256, 2019.

[19]. R. Kumar, X. Zhang, W. Wang, R. U. Khan, J. Kumar and A. Sharif, "A Multimodal Malware Detection Technique for Android IoT Devices Using Various Features" in IEEE Access, vol. 7, pp. 64411-64430, 2019.

[20]. T. Lei, Z. Qin, Z. Wang, Q. Li and D. Ye, "EveDroid: Event-Aware Android Malware Detection Against Model Degrading for IoT Devices" in IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6668-6680, Aug. 2019.

[21]. Z. Ma, H. Ge, Y. Liu, M. Zhao, and J. Ma, "A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms" in IEEE Access, vol. 7, pp. 21235-21245, 2019.

[22]. A. Azmoodeh, A. Dehghantanha and K. R. Choo, "Robust Malware Detection for Internet of (Battlefield) Things Devices Using Deep Eigenspace Learning" in IEEE Transactions on Sustainable Computing, vol. 4, no. 1, pp. 88-95, 1 Jan.-March 2019.

[23]. A. Demontis, M. Melis, B. Biggio, D. Maiorca, D. Arp, K. Rieck " Yes, Machine Learning Can Be More Secure! A Case Study on Android Malware Detection" in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 4, pp. 711-724, 1 July-Aug. 2019.

[24]. P. Feng, J. Ma, C. Sun, X. Xu and Y. Ma, "A Novel Dynamic Android Malware Detection System with Ensemble Learning" in IEEE Access, vol. 6, pp. 30996-31011, 2018.

[25]. Z. Yuan, Y. Lu and Y. Xue, "Droid detector: android malware characterisation and detection using deep learning" in Tsinghua Science and Technology, vol. 21, no. 1, pp. 114-123, Feb. 2016.

[26]. L. Cen, C. S. Gates, L. Si and N. Li, "A Probabilistic Discriminative Model for Android Malware Detection with Decompiled Source Code" in IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 4, pp. 400-412, 1 July-Aug. 2015.

[27]. G. Canfora, F. Mercaldo and C. A. Visaggio, "Mobile malware detection using opcode frequency histograms" 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), Colmar, France, 2015, pp. 27-38.

[28]. A. Karim, R. Salleh and S. A. A. Shah, "DeDroid: A Mobile Botnet Detection Approach Based on Static Analysis" 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 2015, pp. 13271332.

[29]. V. Wahanggara and Y. Prayudi, "Malware Detection through Call System on Android Smartphone Using Vector Machine Method," 2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec), Jakarta, Indonesia, 2015, pp. 62-67.

[30]. K. A. P. Da Costa, L. A. Da Silva, G. B. Martins, G. H. Rosa, C. R. Pereira, and J. P. Papa, "Malware Detection in Android-Based Mobile Environments Using Optimum-Path Forest" 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2015, pp. 754-759.

[31]. Zhongyuan Qin, Yuqing Xu, Yuxing Di, Qunfang Zhang and Jie Huang, "Android malware detection based on permission and behaviour analysis" International Conference on Cyberspace Technology (CCT 2014), Beijing, 2014, pp. 1-4.

[32]. Meng Shanshan, Yang Xiaohui, Song Yubo, Zhu Kelong and Chen Fei, Annedroids's sensitive data leakage detection based on API monitoring," International Conference on Cyberspace Technology (CCT 2014), Beijing, 2014, pp. 1-4.

[33]. H. Zeng, Y. Ren, Q. -X. Wang, N. -Q. He and X. -Y. Ding, "Detecting malware and evaluating the risk of the app using Android permission-API system," 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing(ICCWAMTIP), Chengdu, China, 2014, pp. 440-443.

[34]. S. Gunalakshmii and P. Ezhumalai, "Mobile keylogger detection using machine learning technique," Proceedings of IEEE International Conference on Computer Communication and Systems ICCCS14, Chennai, India, 2014, pp. 051-056.

[35]. H. Shahriar and V. Clincy, "Detection of repackaged Android Malware," The 9th International Conference for Internet Technology and Secured Transactions (ICITST2014), London, UK, 2014, pp. 349-354.