ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

AUTOMATED ANDROID MALWARE DETECTION USING OPTIMAL ENSEMBLE LEARNING APPROACH

Inkollu Keerthi Sai

21N81A6216

Computer Science and Engineering (Cybersecurity)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

inkollukeerthisai@gmail.com

Akenagari Akshaya

21N81A6215

Computer Science and Engineering (Cybersecurity)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

akshayagoud16@gmail.com

Rachakonda Harshavardhan

21N81A6246

Computer Science and Engineering (Cybersecurity) Engineering(Cybersecurity)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

rachasri.harsha69@gmail.com

Nagolu Dheeraj Reddy

21N81A6231

Computer Science and

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

nagoludheeraj11@gmail.com

Mrs. P. Sandhya Reddy

Assistant Head Of The Department

Computer Science and Engineering (Cybersecurity)

Sphoorthy Engineering College,

Nadergul, Hyderabad,501510

sandhyareddy8.p@gmail.com

1. **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

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

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 (AAMD-OELAC) 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 threeMLmodels, namely Least Square SupportVector 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.

Keywords— Android Malware Detection, Ensemble Learning, Deep Learning (DL) ,Machine Learning (ML) ,Hunter-Prey Optimization (HPO), LS-SVM, KELM, RRVFLN.

2. INTRODUCTION

Cybersecurity is a growing concern, particularly in combating Android malware due to the widespread use of the Android operating system and the evolving sophistication of malicious applications. Traditional malware detection methods—**static**, **dynamic**, and **hybrid analyses**—each have limitations, especially against code obfuscation and runtime evasion.

To address these challenges, the paper introduces a novel method called **AAMD-OELAC (Automated Android Malware Detection using Optimal Ensemble**

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Learning Approach for Cybersecurity). This technique integrates:

- **Data preprocessing** to prepare Android Application Packages (APKs) for analysis.
- An **ensemble learning approach** combining three machine learning models:
  - **Least Square Support Vector Machine (LS-SVM)**
  - **Kernel Extreme Learning Machine (KELM)**
  - **Regularized Random Vector Functional Link Neural Network (RRVFLN)**
- **Hunter-Prey Optimization (HPO)** for optimal hyperparameter tuning of the models.

Key Contributions:

1. A novel, intelligent Android malware detection framework combining ensemble learning and hyperparameter optimization.
2. Enhanced detection accuracy through a blend of multiple classifiers.
3. Effective identification of malicious APK behaviors using ML/DL techniques.

This approach demonstrates superior performance in identifying malware, contributing significantly to Android cybersecurity research.

1. LITERATURE REVIEW

‘‘Adversarial superiority in Android malware detection: Lessons from reinforcement learning based evasion attacks and defenses,’’

Today, android smartphones are being used by billions of users and thus have become a lucrative target of malware designers. Therefore being one step ahead in this zero-sum game of malware detection between the anti-malware community and malware developers is more of a necessity than a desire. This work focuses on a proactive adversary-aware framework to develop adversarially superior android malware detection models. We first investigate the adversarial robustness of thirty-six distinct malware detection models constructed using two static features (permission and intent) and eighteen classification algorithms. We designed two Targeted Type-II Evasion Attacks (*TRPO-MalEAttack* and *PPO-MalEAttack*) based on reinforcement learning to exploit vulnerabilities in the

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

above malware detection models. The attacks aim to add minimum perturbations in each malware application and convert it into an adversarial application that can fool the malware detection models. The TRPO-MalEAttack achieves an average fooling rate of 95.75% (with 2.02 mean perturbations), reducing the average accuracy from 86.01% to 49.11% in thirty-six malware detection models. On the other hand, The PPO-MalEAttack achieves a higher average fooling rate of 96.87% (with 2.08 mean perturbations), reducing the average accuracy from 86.01% to 48.65% in the same thirty-six detection models. We also develop a list of the *TEN* most vulnerable android permissions and intents that an adversary can use to generate more adversarial applications. Later, we propose a defense strategy (*MalVPatch*) to counter the adversarial attacks on malware detection models. The MalVPatch defense achieves higher detection accuracy along with a drastic improvement in the adversarial robustness of malware detection models. Finally, we conclude that investigating the adversarial robustness of models is necessary before their real-world deployment and helps achieve adversarial superiority in android malware detection.

‘‘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, we propose a hybrid Android malware detection approach that combines dynamic and static tactics. We 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, we extract the object reference relationships from the memory heap to construct a dynamic feature base. We 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. Besides, our system is demonstrated to resist

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

permission abuse behaviors and obfuscation techniques.

‘‘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. In addition, the Adamax optimizer with attention recurrent autoencoder (ARAE) model is employed for Android malware detection. The experimental validation of the RHSODL-AMD technique on the Andro-AutoPsy dataset exhibits its promising performance, with a maximum accuracy of 99.05%.

‘‘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

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

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. As a result, our model overcomes the related works when using just two classes with F1-score of 99.5%. Also, high malware detection and classification performance was obtained by using the five classes, with F1-score of 97%.

3. METHODOLOGY

The AAMD-OELAC approach is designed to enhance the detection and classification of Android malware through an optimized ensemble learning framework. The methodology involves several key phases:

1. Data Preprocessing

- Raw Android application data (likely APK files) is preprocessed.
- Features are extracted that are relevant for distinguishing malware from benign applications.
- Preprocessing ensures data cleanliness, normalization, and suitability for machine learning algorithms.

2. Ensemble Learning Framework

- The core of the AAMD-OELAC system is its **ensemble learning model**, which combines the predictive capabilities of three distinct machine learning models:
  - **Least Squares Support Vector Machine (LS-SVM)**
  - **Kernel Extreme Learning Machine (KELM)**
  - **Regularized Random Vector Functional Link Neural Network (RRVFLN)**
- These models work collaboratively to improve classification accuracy by leveraging their individual strengths.

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

3. Hyperparameter Optimization

- The **Hunter-Prey Optimization (HPO)** algorithm is used to fine-tune the parameters of all three ensemble models.
- HPO is a metaheuristic inspired by the behavior of predator-prey interactions in nature, applied here to enhance learning performance.

4. Classification and Detection

- The optimized ensemble model is used to classify the Android applications as **malware or benign**.
- The combination of models and optimal tuning allows for more precise detection, even with complex or obfuscated malware.

5. Evaluation and Validation

- A comprehensive **experimental analysis** is conducted.
- Accuracy, prediction ratio, and comparison with existing methods are used as metrics to evaluate the effectiveness of the AAMD-OELAC system.
- Results are visualized using bar charts and statistical summaries.

This layered methodology ensures that the malware detection system is not only accurate but also robust against evolving malware threats. The use of an ensemble learning strategy, combined with metaheuristic optimization, represents a significant advancement over traditional single-model approaches.

Proposed System:

This paper presents an Automated Android Malware Detection using Optimal Ensemble Learning Approach for Cybersecurity (AAMD-OELAC) technique. The AAMDOELAC 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) algorithm is exploited for the optimal parameter tuning of the three DL models, and it helps accomplish improved malware detection results. To indicate the supremacy of the AAMD-OELAC approach, a comprehensive experimental analysis is carried out.

Advantages

• An intelligent AAMD-OELAC technique comprising data preprocessing, ensemble learning, and HPO-based hyperparameter tuning is presented for Android malware detection. To the best of our knowledge, the

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

AAMD-OELAC technique never existed in the literature.

• Perform ensemble learning-based classification process comprising LS-SVM, KELM, and RRVFLN models for Android malware detection.

• The combination of the HPO algorithm and ensemble learning process improves the detection accuracy of Android malware. By utilizing multiple classifiers and optimization strategies, the model can effectively identify malicious patterns and behaviours in Android applications.

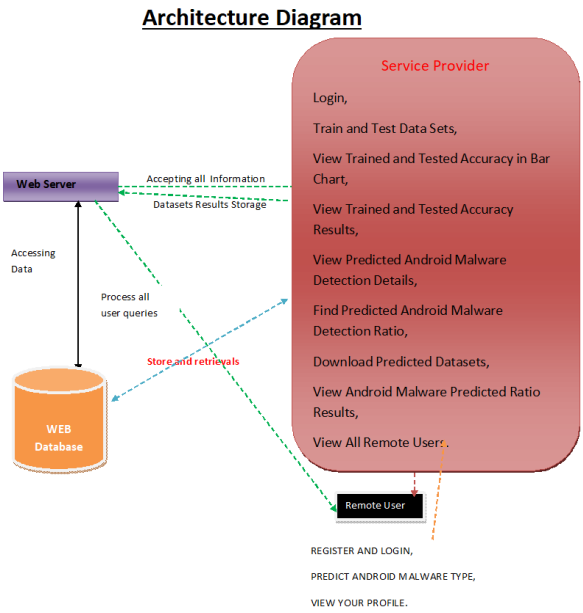

Fig.1: System architecture

1. IMPLEMENTATION

Modules

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Train & Test Data Sets, View Trained and Tested Datasets Accuracy in Bar Chart, View Trained and Tested Datasets Accuracy Results, View Prediction Of Cyber Attack Status, View Cyber Attack Prediction Status Ratio, Download Predicted Data Sets, View Cyber Attack Prediction Status Ratio Results, View All Remote Users..

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN,

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

PREDICT CYBER ATTACK STATUS, VIEW YOUR PROFILE.

ALGORITHMS

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

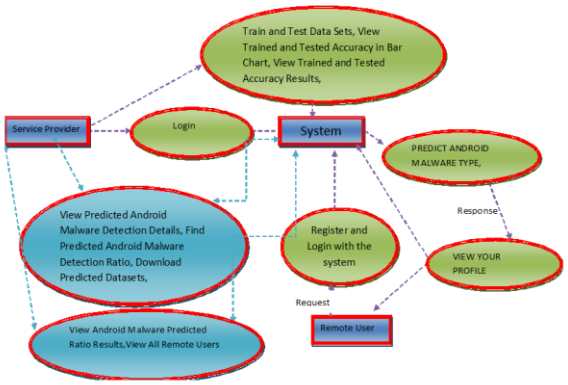

Fig 2 WorkFlow

5. EXPERIMENTAL RESULTS

Fig.1: User Interface.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Fig 2: Accuracy

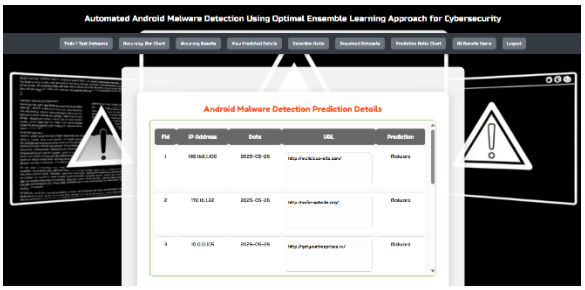

Fig 3: Prediction Details

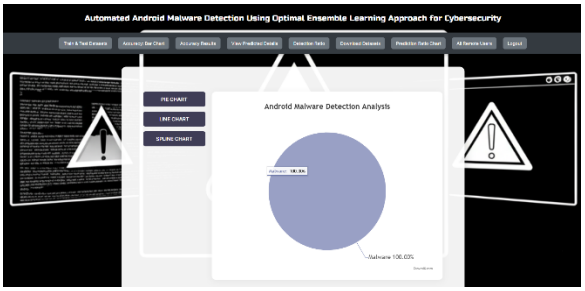

Fig.4: Detection Analysis

6. CONCLUSION

In this study, we have developed the design of the AAMD-OELAC technique for an accurate and automated Android malware detection process. The intention of the AAMD-OELAC approach focused on the automatic recognition and classification of Android malware. To achieve this, the AAMD-OELAC technique encompasses data preprocessing, ensemble classification, and HPO-based parameter tuning. For the Android malware detection process, the AAMD-OELAC technique follows an ensemble learning process using three ML models namely LS-SVM, KELM, and RRVFLN. Finally, the HPO algorithm is exploited for the optimal parameter tuning of the three DL models and it helps in accomplishing improved malware detection results. To portray the supremacy of the AAMD-OELAC method, a wide-ranging experimental analysis is conducted. The simulation results portrayed the supremacy of the AAMDOELAC technique over other existing approaches. Future work could focus on developing more advanced techniques to capture and analyze fine-grained behaviors, enabling better detection of sophisticated malware. In addition, future work could explore privacy-preserving approaches such as secure multi-party computation or federated learning, which enable collaborative malware detection without compromising user privacy.

REFERENCES

[1] H. Rathore, A. Nandanwar, S. K. Sahay, and M. Sewak, ‘‘Adversarial superiority in Android malware detection: Lessons from reinforcement learning based evasion attacks and defenses,’’ *Forensic Sci. Int., Digit. Invest.*, vol. 44, Mar. 2023, Art. no. 301511.

[2] H. Wang, W. Zhang, and H. He, ‘‘You are what the permissions told me! Android malware detection based on hybrid tactics,’’ *J. Inf. Secur. Appl.*, vol. 66, May 2022, Art. no. 103159.

[3] A. Albakri, F. Alhayan, N. Alturki, S. Ahamed, and S. Shamsudheen, ‘‘Metaheuristics with deep learning model for cybersecurity and Android malware detection and classification,’’ *Appl. Sci.*, vol. 13, no. 4, p. 2172, Feb. 2023.

[4] M. Ibrahim, B. Issa, and M. B. Jasser, ‘‘A method for automatic Android malware detection based on static analysis and deep learning,’’ *IEEE Access*, vol. 10, pp. 117334–117352, 2022.

[5] L. Hammood, İ. A. Doğru, and K. Kılıç, ‘‘Machine learning-based adaptive genetic algorithm for Android malware detection in auto-driving vehicles,’’ *Appl. Sci.*, vol. 13, no. 9, p. 5403, Apr. 2023.

[6] P. Bhat and K. Dutta, ‘‘A multi-tiered feature selection model for Android malware detection based on feature discrimination and information gain,’’ *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9464–9477, Nov. 2022.

[7] D.Wang, T. Chen, Z. Zhang, and N. Zhang, ‘‘A survey of Android malware detection based on deep learning,’’ in *Proc. Int. Conf. Mach. Learn. Cyber Secur.* Cham, Switzerland: Springer, 2023, pp. 228–242.

[8] Y. Zhao, L. Li, H. Wang, H. Cai, T. F. Bissyandé, J. Klein, and J. Grundy, ‘‘On the impact of sample duplication in machine-learning-based Android malware detection,’’ *ACM Trans. Softw. Eng. Methodol.*, vol. 30, no. 3, pp. 1–38, Jul. 2021.

[9] E. C. Bayazit, O. K. Sahingoz, and B. Dogan, ‘‘Deep learning based malware detection for Android systems: A comparative analysis,’’ *Tehnički vjesnik*, vol. 30, no. 3, pp. 787–796, 2023.

[10] H.-J. Zhu, W. Gu, L.-M. Wang, Z.-C. Xu, and V. S. Sheng, ‘‘Android malware detection based on multi-head squeeze-and-excitation residual network,’’ *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118705.

[11] K. Shaukat, S. Luo, and V. Varadharajan, ‘‘A novel deep learning-based approach for malware detection,’’ *Eng. Appl. Artif. Intell.*, vol. 122, Jun. 2023, Art. no. 106030.

[12] J. Geremias, E. K. Viegas, A. O. Santin, A. Britto, and P. Horchulhack, ‘‘Towards multi-view Android malware detection through image-based deep learning,’’ in *Proc. Int.Wireless Commun. Mobile Comput. (IWCMC)*, May 2022, pp. 572–577. 72516 VOLUME 11, 2023 IEEE Transaction on

Machine Learning,Volume:11,Issue Date:11.July.2023

[13] J. Kim, Y. Ban, E. Ko, H. Cho, and J. H. Yi, ‘‘MAPAS: A practical deep learning-based Android malware detection system,’’ *Int. J. Inf. Secur.*, vol. 21, no. 4, pp. 725–738, Aug. 2022.

[14] S. Fallah and A. J. Bidgoly, ‘‘Android malware detection using network traffic based on sequential deep learning models,’’ *Softw., Pract. Exper.*, vol. 52, no. 9, pp. 1987–2004, Sep. 2022.

[15] V. Sihag, M. Vardhan, P. Singh, G. Choudhary, and S. Son, ‘‘De-LADY: Deep learning-based Android malware detection using dynamic features,’’ *J. Internet Serv. Inf. Secur.*, vol. 11, no. 2, p. 34, 2021.

[16] W. Wang, M. Zhao, and J. Wang, ‘‘Effective Android malware detection with a hybrid model based on deep autoencoder and convolutional neural network,’’ *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3035–3043, Aug. 2019.

[17] P. Yadav, N. Menon, V. Ravi, S. Vishvanathan, and T. D. Pham, ‘‘Efficient-Net convolutional neural networks-based Android malware detection,’’ *Comput. Secur.*, vol. 115, Apr. 2022, Art. no. 102622.

[18] M. Masum and H. Shahriar, ‘‘Droid-NNet: Deep learning neural network for Android malware detection,’’ in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 5789–5793.

[19] F. Idrees, M. Rajarajan, M. Conti, T. M. Chen, and Y. Rahulamathavan, ‘‘PIndroid: A novel Android malware detection system using ensemble learning methods,’’ *Comput. Secur.*, vol. 68, pp. 36–46, Jul. 2017.

[20] A. Guerra-Manzanares, H. Bahsi, and M. Luckner, ‘‘Leveraging the first line of defense: A study on the evolution and usage of Android security permissions for enhanced Android malware detection,’’ *J. Comput. Virol. Hacking Techn.*, vol. 19, no. 1, pp. 65–96, Aug. 2022.

[21] A. Taha and O. Barukab, ‘‘Android malware classification using optimized ensemble learning based on genetic algorithms,’’ *Sustainability*, vol. 14, no. 21, p. 14406, Nov. 2022.

[22] K. Sabanci, M. F. Aslan, E. Ropelewska, and M. F. Unlersen, ‘‘A convolutional neural network-based comparative study for pepper seed classification: Analysis of selected deep features with support vector machine,’’ *J. Food Process Eng.*, vol. 45, no. 6, Jun. 2022, Art. no. e13955.

[23] A. Batouche and H. Jahankhani, ‘‘A comprehensive approach to Android malware detection using machine learning,’’ in *Information Security Technologies for*

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

Controlling Pandemics. USA: Springer, 2021, pp. 171–212.

[24] O. N. Elayan and A. M. Mustafa, ‘‘Android malware detection using deep learning,’’ *Proc. Comput. Sci.*, vol. 184, pp. 847–852, Jan. 2021.

[25] S. S. Sammen, M. Ehteram, Z. Sheikh Khozani, and L. M. Sidek,‘‘Binary coati optimization algorithm- multi- kernel least square support vector machine-extreme learning machine model (BCOAMKLSSVM- ELM): A new hybrid machine learning model for predicting reservoir water level,’’ *Water*, vol. 15, no. 8, p. 1593, Apr. 2023.