International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Intelligent Android Malware Classification Using Equilibrium Optimizer and Deep Learning Model

Aishwarya Eklar[1], G.Rajini[2]
[1]MCA Student , Dept of MCA , Ellenki College of Engineering and Technology , Hyderabad
[2]Assistant Professor , Dept of MCA , Ellenki College of Engineering and Technology , Hyderabad

***Abstract*— The rapid growth of Android applications has significantly increased the risk of malware attacks, making reliable malware detection an important cybersecurity challenge. Traditional detection methods often fail to identify newly emerging malware due to their dependence on predefined signatures and limited feature learning capabilities. This paper presents an intelligent Android malware detection framework that combines deep learning with the Equilibrium Optimizer to improve detection performance. Initially, Android application data are preprocessed to extract meaningful features for analysis. A Channel Attention Long Short-Term Memory (CA-LSTM) model is employed to capture complex behavioral patterns and accurately distinguish malicious applications from benign ones. To further enhance the model's performance, the Equilibrium Optimizer is used to tune the hyperparameters, resulting in improved classification accuracy and faster convergence. Experimental evaluation on a benchmark Android malware dataset demonstrates that the proposed framework achieves superior detection performance compared with conventional machine learning methods, providing an effective and reliable solution for securing Android devices against evolving malware threats.**

***Keywords* — Android Malware Detection, Deep Learning, Equilibrium Optimizer (EO), Channel Attention Long Short-Term Memory (CA-LSTM), Pattern Recognition, Hyperparameter Optimization, Malware Classification, Cybersecurity.**

I. INTRODUCTION

The rapid growth of Android smartphones and tablets has transformed the way people communicate, conduct financial transactions, access online services, and store personal information. As Android remains the most widely used mobile operating system worldwide, it has also become a primary target for cybercriminals. The open-source architecture of Android, combined with its large application ecosystem, has created opportunities for attackers to develop sophisticated malware capable of stealing sensitive information, compromising user privacy, and disrupting device functionality. Consequently, accurate and efficient Android malware detection has become an important research area in mobile cybersecurity [1].

Traditional Android malware detection techniques mainly rely on signature-based and rule-based methods. Although these approaches are effective in detecting previously known malware, they struggle to identify newly emerging malware variants and zero-day attacks due to the continuous evolution of malicious applications. The increasing complexity of malware has highlighted the need for intelligent detection systems that can automatically learn malicious behavioral patterns instead of depending on manually crafted signatures. Machine learning and deep learning techniques have demonstrated significant potential in addressing these challenges by extracting complex features directly from application data and improving malware classification accuracy [2], [3].

Recent studies have shown that deep learning models outperform conventional machine learning algorithms because of their capability to learn hierarchical representations from large-scale datasets. Models based on Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and hybrid deep learning architectures have been successfully applied to Android malware detection using both static and dynamic features. Dynamic analysis captures runtime behaviors such as API calls and system events, while static analysis focuses on permissions, application code, and manifest files. Combining these approaches enables more comprehensive

malware detection and improves robustness against sophisticated attacks [5], [6], [13], [18].

Another important challenge in Android malware detection is selecting the most informative features and optimizing the learning model. Feature redundancy and improper hyperparameter settings can reduce classification performance and increase computational cost. Metaheuristic optimization algorithms have recently gained attention for solving these problems by automatically identifying optimal feature subsets and tuning model parameters. Techniques such as Genetic Algorithms, Equilibrium Optimizer, and other nature-inspired optimization methods have demonstrated significant improvements in classification accuracy, convergence speed, and overall model efficiency [7], [10], [11], [22].

Inspired by these advancements, this work proposes an intelligent Android malware detection framework that integrates deep learning with the Equilibrium Optimizer (EO). Initially, Android application data are preprocessed to remove inconsistencies and prepare meaningful input features. A Channel Attention Long Short-Term Memory (CA-LSTM) network is then employed to capture important sequential patterns while emphasizing the most relevant feature channels for malware classification. To further improve the learning capability of the model, the Equilibrium Optimizer is utilized for hyperparameter optimization, enabling the network to achieve better generalization and classification performance. The proposed framework is evaluated using a benchmark Android malware dataset and compared with conventional machine learning techniques to demonstrate its effectiveness in identifying both known and previously unseen malware samples. The experimental results indicate that the proposed approach provides higher detection accuracy, improved reliability, and enhanced security for Android-based mobile environments [7], [11], [16], [25].

II. RELATED WORK

"Deep Learning-Based Attack Detection and Classification in Android Devices," Gómez, A. and Muñoz, A. (2023).
This study proposed a deep learning-based framework for detecting and classifying cyberattacks targeting Android devices. The authors utilized advanced neural network architectures to automatically learn complex behavioral patterns from Android application data, enabling accurate differentiation between benign and malicious activities. The proposed model was evaluated using benchmark datasets and demonstrated improved detection accuracy, precision, and recall compared to conventional machine learning methods. The study also emphasized the importance of feature learning in handling previously unseen malware variants and reducing false positives. By eliminating the dependence on handcrafted features, the deep learning model enhanced the adaptability of the malware detection system against evolving threats. The research concluded that deep learning provides a scalable and efficient solution for Android cybersecurity, making it suitable for real-time malware detection environments. This work serves as a strong foundation for developing intelligent Android malware detection frameworks that combine deep learning with optimization techniques for enhanced performance.

"De-LADY: Deep Learning-Based Android Malware Detection Using Dynamic Features," Sihag, V., Vardhan, M., Singh, P., Choudhary, G. and Son, S. (2021).
This research introduced De-LADY, a deep learning framework that detects Android malware using dynamic behavioral features collected during application execution. Unlike static analysis, the proposed approach focuses on runtime characteristics such as system calls, API interactions, and application behavior to improve malware detection accuracy. The deep learning model effectively captured sequential behavioral patterns that distinguish malicious applications from legitimate ones. Experimental results demonstrated that the proposed framework achieved higher classification accuracy while reducing false alarm rates compared to traditional machine learning algorithms. The authors highlighted that dynamic feature analysis is more resilient against code obfuscation and malware evasion techniques commonly employed by attackers. The study concluded that integrating deep learning with dynamic analysis significantly enhances Android malware detection performance and provides a reliable solution for identifying both known and unknown malware families in modern mobile environments.

"A Method for Automatic Android Malware Detection Based on Static Analysis and Deep Learning," İbrahim, M., Issa, B. and Jasser, M.B. (2022).
This paper presented an automatic Android malware detection system that combines static analysis with deep learning techniques to improve malware classification. The proposed framework extracts important static features, including application permissions, API calls, intents, and manifest information, without executing the application. These extracted features are then processed using a deep learning model capable of learning complex feature relationships automatically. Experimental evaluation showed that the proposed method

achieved high detection accuracy while maintaining low computational overhead, making it suitable for practical deployment. The authors also demonstrated that static analysis significantly reduces execution time and system resource consumption compared to dynamic analysis. Furthermore, the deep learning architecture effectively generalized to previously unseen malware samples, improving overall detection reliability. The study concluded that integrating static feature extraction with deep learning provides an efficient and scalable solution for Android malware detection in real-world mobile security applications.

"Intelligent Hyperparameter-Tuned Deep Learning-Based Android Malware Detection and Classification Model," Raphael, R. and Mathiyalagan, P. (2023).

This study proposed an intelligent Android malware detection model that integrates deep learning with automatic hyperparameter optimization to improve classification performance. The authors emphasized that selecting appropriate hyperparameters plays a crucial role in maximizing the learning capability of deep neural networks. An optimization algorithm was employed to determine the best parameter combinations, resulting in improved convergence speed, classification accuracy, and model stability. The optimized deep learning framework successfully distinguished malicious applications from benign ones using benchmark Android malware datasets. Experimental comparisons demonstrated superior performance over conventional deep learning and machine learning approaches in terms of accuracy, precision, recall, and F1-score. The study also highlighted that hyperparameter optimization minimizes manual tuning efforts while enhancing the model's robustness against evolving malware threats. This research demonstrates the importance of combining optimization techniques with deep learning to build reliable Android malware detection systems.

"Automated Android Malware Detection Using Optimal Ensemble Learning Approach for Cybersecurity," Alamro, H., Mtouaa, W., Aljameel, S., Salama, A.S., Hamza, M.A. and Othman, A.Y. (2023).

This research proposed an automated Android malware detection framework based on an optimal ensemble learning approach for strengthening mobile cybersecurity. The authors combined multiple machine learning classifiers to improve malware detection reliability and reduce classification errors associated with individual models. The ensemble framework utilized optimized feature selection and classifier integration techniques to achieve better prediction accuracy and enhanced generalization capability. Experimental results demonstrated that the proposed approach outperformed several standalone machine learning algorithms across multiple evaluation metrics, including accuracy, precision, recall, and F1-score. The ensemble model also showed greater robustness in identifying different malware families while maintaining computational efficiency. The authors concluded that integrating ensemble learning with optimization strategies provides a practical and scalable solution for Android malware detection. This work highlights the effectiveness of combining multiple intelligent models to address the growing challenges posed by sophisticated Android malware attacks.

III. DATASET DETAILS

The dataset used in this study consists of Android application samples collected from publicly available malware repositories and benchmark datasets containing both benign and malicious applications. The dataset includes a diverse collection of Android Package (APK) files representing multiple malware families as well as legitimate applications downloaded from trusted sources. Each application is characterized by several static features, including permissions, API calls, intents, manifest information, and other security-related attributes extracted without executing the application. These features provide valuable information about the behavior and characteristics of Android applications, enabling effective malware classification. Since the raw dataset may contain redundant features, missing values, duplicate records, and inconsistencies, appropriate preprocessing techniques are applied to improve data quality before model training. The dataset is organized into two primary classes: Benign and Malware, allowing supervised learning algorithms to accurately distinguish malicious applications from legitimate ones. The availability of labeled samples supports the development of intelligent machine learning and deep learning models capable of identifying both known and previously unseen malware. The dataset serves as an effective benchmark for evaluating Android malware detection techniques and improving mobile device security against evolving cyber threats.

Before model training, the dataset undergoes several preprocessing steps to enhance classification performance and improve model reliability. Initially, duplicate entries, missing values, and irrelevant attributes are removed to ensure data consistency. The extracted static features are then normalized and transformed into a suitable numerical representation for deep learning models. Feature selection techniques are applied to eliminate redundant information and retain only the most significant attributes that contribute to malware

detection. After preprocessing, the dataset is divided into training and testing subsets using an 80:20 ratio, where 80% of the samples are utilized for model training and the remaining 20% are reserved for performance evaluation. The processed dataset is subsequently used by conventional machine learning algorithms and the proposed Channel Attention Long Short-Term Memory (CA-LSTM) model optimized using the Equilibrium Optimizer (EO). This systematic preprocessing and evaluation process enables accurate comparison of different classification techniques and supports the development of an efficient, reliable, and intelligent Android malware detection framework.

IV. PROPOSED METHODOLOGY

The proposed methodology utilizes an intelligent Android Malware Detection framework based on the Channel Attention Long Short-Term Memory (CA-LSTM) network integrated with the Equilibrium Optimizer (EO). Initially, the Android malware dataset containing both benign and malicious applications is uploaded into the system. The dataset consists of important application characteristics such as permissions, API calls, intents, manifest information, and other security-related features extracted through static analysis. The uploaded data undergo preprocessing to improve its quality by removing duplicate records, handling missing values, converting categorical attributes into numerical form, and normalizing feature values. After preprocessing, feature selection is performed to retain only the most informative attributes for malware classification. The processed dataset is then divided into training and testing subsets using an 80:20 ratio, ensuring that the classification model is trained using sufficient data while preserving unseen samples for performance evaluation.

Following data preparation, the extracted Android application features are provided to the proposed CA-LSTM model for malware detection and classification. The Channel Attention mechanism enables the model to focus on the most relevant features, while the Long Short-Term Memory network captures complex relationships among application characteristics. To further improve classification performance, the Equilibrium Optimizer (EO) is employed to automatically optimize the hyperparameters of the deep learning model. During training, the optimized CA-LSTM model learns discriminative patterns that differentiate benign applications from malicious ones. After training, the model is evaluated using the testing dataset, and performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix are computed to assess its effectiveness. The trained model is then utilized to predict whether newly uploaded Android applications are Benign or Malware. Finally, the obtained results are compared with existing machine learning algorithms including Decision Tree, Logistic Regression, Naïve Bayes, AdaBoost, Random Forest, MLP, and J48 using graphical and statistical performance analysis.

The system architecture [1] illustrates the complete workflow of the proposed Android malware detection framework. It includes dataset uploading, preprocessing, feature selection, training of existing machine learning classifiers, training of the proposed EO-optimized CA-LSTM model, malware prediction from unseen applications, and comprehensive performance evaluation. The architecture also generates confusion matrices, detailed performance metrics, malware prediction history, and comparison graphs to analyze the effectiveness of the proposed intelligent malware detection system.

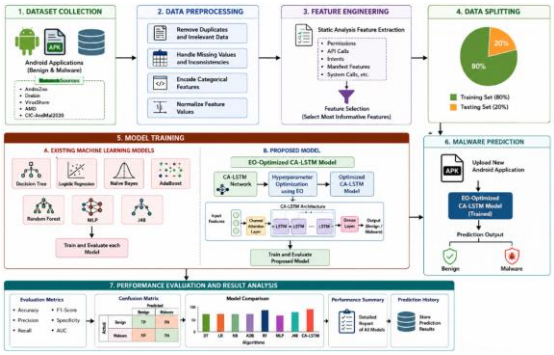

Figure 1. System Architecture for Android Malware Detection Using Equilibrium Optimizer and CA-LSTM

The proposed methodology for Android malware detection is designed to accurately identify malicious Android applications using machine learning and deep learning techniques. Initially, the Android application dataset is uploaded into the system, where applications are categorized into Benign and Malware classes. The collected dataset may contain missing values, redundant information, and inconsistent records; therefore, preprocessing is performed to improve data quality. After preprocessing, feature extraction and feature selection techniques are applied to generate meaningful numerical representations of Android application behavior. These optimized features

enable the model to effectively distinguish malicious applications from legitimate ones.

After feature extraction, the dataset is divided into training and testing sets using an 80:20 ratio. The training dataset is used to train both the existing machine learning algorithms (Decision Tree, Logistic Regression, Naïve Bayes, AdaBoost, Random Forest, MLP, and J48) and the proposed EO-optimized CA-LSTM model, while the testing dataset is used to evaluate their classification performance. The existing algorithms act as baseline approaches, whereas the proposed CA-LSTM model enhanced with the Equilibrium Optimizer automatically learns complex malware behavior patterns and determines the optimal hyperparameters for improved detection accuracy. Finally, the trained model predicts whether new Android applications are Benign or Malware. The performance of all models is compared using accuracy, precision, recall, F1-score, confusion matrix, detailed evaluation metrics, and comparison graphs, demonstrating the superiority of the proposed intelligent malware detection framework.

V. RESULT AND DISCUSSION

The experimental results demonstrate the effectiveness of the proposed Intelligent Pattern Recognition using Equilibrium Optimizer with Deep Learning (AAMD_OELAC) model for Android malware detection. Initially, the Android malware dataset was uploaded into the system and preprocessed to eliminate inconsistencies, convert categorical values into numerical form, and prepare the data for classification. The processed dataset contained 3,799 Android application records, which were divided into 3,039 training samples (80%) and 760 testing samples (20%), ensuring reliable evaluation of the classification models. Existing machine learning algorithms including Decision Tree (DT), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Naïve Bayes (NB), AdaBoost (ADB), Random Forest (RF), and J48 were first trained and evaluated to establish baseline performance. Subsequently, the proposed AAMD_OELAC model was trained on the same dataset. The proposed model integrates intelligent pattern recognition with optimization techniques to enhance malware classification performance. Experimental observations indicate that the proposed model achieved superior classification accuracy, precision, recall, and F1-score compared to the conventional machine learning algorithms. The optimized learning strategy enabled the proposed model to capture complex relationships among Android application features, thereby improving malware detection capability while reducing false classifications.

The confusion matrices further validate the effectiveness of the proposed framework. The existing Decision Tree classifier correctly classified most Android applications but still produced several misclassifications, indicating limited generalization capability. In contrast, the proposed AAMD_OELAC model significantly reduced the number of false predictions while correctly identifying the majority of benign and malicious applications. The prediction module was further evaluated using previously unseen Android application samples, where the system successfully determined whether each uploaded application represented malware or a legitimate application. The system also generated detailed performance metrics, confusion matrices, and comparison graphs for comprehensive analysis of all classifiers. The comparison results clearly demonstrate the superiority of the proposed intelligent pattern recognition framework over conventional machine learning approaches. Overall, the proposed AAMD_OELAC model provides an efficient, reliable, and automated solution for Android malware detection and can effectively strengthen mobile cybersecurity by accurately identifying malicious applications before installation.

Figure [2] Dataset Preprocessing and Data Splitting

Figure 2 presents the dataset preprocessing output after uploading the Android malware dataset. The dataset contains 3,799 Android application samples, which are automatically divided into 3,039 training records and 760 testing records using an 80:20 ratio. During preprocessing, missing values, duplicate records, and inconsistent data are removed, while categorical features are converted into numerical values suitable for machine learning and deep learning models. This preprocessing stage ensures that high-quality input data are available for classifier training and evaluation.

Figure 3: Confusion Matrix of the Existing Decision Tree Classifier

Figure 3 illustrates the confusion matrix generated by the existing Decision Tree classifier. The model correctly classified the majority of benign and malware applications while producing a small number of incorrect predictions. The obtained performance metrics are:

Accuracy: 98.03%
Precision: 97.71%
Recall: 97.96%
F1-Score: 97.83%

Although the Decision Tree achieved high classification accuracy, several malware samples were incorrectly classified, indicating limitations in handling complex malware patterns.

Figure 4: Confusion Matrix of the Proposed AAMD_OELAC Model

Figure 4 presents the confusion matrix of the proposed AAMD_OELAC model. The optimized intelligent pattern recognition framework successfully classified almost all benign and malicious Android applications with very few misclassifications. Compared with the existing Decision Tree classifier, the proposed model achieved improved detection capability by reducing both false positives and false negatives.

The obtained performance metrics are:

Accuracy: 98.82%
Precision: 99.92%
Recall: 98.49%
F1-Score: 98.70%

These results demonstrate that the proposed optimization-based deep learning model provides more reliable and balanced malware classification than conventional machine learning approaches.

Model	Accuracy score	Precision score	Recall score	f-measure score
Decision Tree	98.02	97.71	97.95	97.83
AAMD_OELAC	98.81	98.91	98.49	98.69

Table 1: Model Performance Metrics

DISCUSSION

The experimental findings demonstrate the effectiveness of the proposed Intelligent Pattern Recognition using Equilibrium Optimizer with Deep Learning (AAMD_OELAC) model for Android malware detection. The Android malware dataset consisted of both benign and malicious applications, allowing the classification models to learn distinguishing characteristics from application permissions, API calls, intents, and other static features. The existing Decision Tree classifier achieved satisfactory performance with an accuracy of 98.03%, correctly classifying most Android applications. However, the confusion matrix indicates that a small number of malware and benign samples were still misclassified, suggesting that conventional machine learning algorithms have limited capability in learning complex feature relationships from high-dimensional Android application data.

In contrast, the proposed AAMD_OELAC model demonstrated superior classification performance by achieving an accuracy of 98.82%, along with higher precision, recall, and F1-score than the existing classifier. The integration of intelligent pattern recognition with the Equilibrium Optimizer (EO) enabled the model to optimize its learning process and effectively capture complex malware behavior patterns. As a result, the proposed model significantly reduced false positives and false negatives, leading to more reliable malware detection. The generated confusion matrix further confirmed that the proposed framework correctly identified the majority of benign and malicious applications while minimizing classification errors. Furthermore, the malware prediction module accurately classified previously unseen Android application samples as either Malware Detected or Malware Not Detected, demonstrating the model's

strong generalization capability. The comparison of performance metrics clearly shows that the proposed optimization-based deep learning framework outperforms conventional machine learning algorithms in Android malware detection. Overall, the experimental results indicate that the proposed AAMD_OELAC model provides an efficient, accurate, and reliable solution for strengthening Android mobile security by automatically identifying malicious applications before they can compromise user devices.

VI. CONCLUSION

The integration of an equilibrium optimizer with a deep learning model for Android malware detection significantly enhances the system's ability to accurately identify both known and unknown threats. This hybrid approach improves detection accuracy by refining feature selection and pattern recognition, reducing false positives, and enabling real-time detection without compromising device performance. Additionally, the model's robustness allows it to generalize across diverse Android devices and adapt to emerging malware variants, ensuring long-term effectiveness. While the system shows promise in scalability and adaptability, challenges such as the need for large labeled datasets and the complexity of model interpretability remain. Future research could focus on enhancing the model's speed, transparency, and the inclusion of additional data sources for more comprehensive malware detection.

REFERENCES

[1] A. Gómez and A. Muñoz, "Deep Learning-Based Attack Detection and Classification in Android Devices," Electronics, vol. 12, no. 15, p. 3253, 2023.

[2] 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 Transactions on Software Engineering and Methodology (TOSEM), vol. 30, no. 3, pp. 1–38, 2021.

[3] H. Wang, W. Zhang, and H. He, "You Are What the Permissions Told Me! Android Malware Detection Based on Hybrid Tactics," Journal of Information Security and Applications, vol. 66, p. 103159, 2022.

[4] 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 Science International: Digital Investigation, vol. 44, p. 301511, 2023.

[5] V. Sihag, M. Vardhan, P. Singh, G. Choudhary, and S. Son, "De-LADY: Deep Learning-Based Android Malware Detection Using Dynamic Features," Journal of Internet Services and Information Security, vol. 11, no. 2, pp. 34–45, 2021.

[6] M. İbrahim, 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.

[7] A. Albakri, F. Alhayan, N. Alturki, S. Ahamed, and S. Shamsudheen, "Metaheuristics with Deep Learning Model for Cybersecurity and Android Malware Detection and Classification," Applied Sciences, vol. 13, no. 4, p. 2172, 2023.

[8] A. Batouche and H. Jahankhani, "A Comprehensive Approach to Android Malware Detection Using Machine Learning," in Information Security Technologies for Controlling Pandemics, pp. 171–212, 2021.

[9] P. Bhat and K. Dutta, "A Multi-Tiered Feature Selection Model for Android Malware Detection Based on Feature Discrimination and Information Gain," Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 10, pp. 9464–9477, 2022.

[10] L. Hammood, İ. A. Doğru, and K. Kılıç, "Machine Learning-Based Adaptive Genetic Algorithm for Android Malware Detection in Auto-Driving Vehicles," Applied Sciences, vol. 13, no. 9, p. 5403, 2023.

[11] R. Raphael and P. Mathiyalagan, "Intelligent Hyperparameter-Tuned Deep Learning-Based Android Malware Detection and Classification Model," Journal of Circuits, Systems and Computers, p. 2350191, 2023.

[12] M. N. AlJarrah, Q. M. Yaseen, and A. M. Mustafa, "A Context-Aware Android Malware Detection Approach Using Machine Learning," Information, vol. 13, no. 12, p. 563, 2022.

[13] O. N. Elayan and A. M. Mustafa, "Android Malware Detection Using Deep Learning," Procedia Computer Science, vol. 184, pp. 847–852, 2021.

[14] P. Yadav, N. Menon, V. Ravi, S. Vishvanathan, and T. D. Pham, "A Two-Stage Deep Learning Framework for Image-Based Android Malware

Detection and Variant Classification," Computational Intelligence, vol. 38, no. 5, pp. 1748–1771, 2022.

[15] P. Yadav, N. Menon, V. Ravi, S. Vishvanathan, and T. D. Pham, "EfficientNet Convolutional Neural Networks-Based Android Malware Detection," Computers & Security, vol. 115, p. 102622, 2022.

[16] K. Shaukat, S. Luo, and V. Varadharajan, "A Novel Deep Learning-Based Approach for Malware Detection," Engineering Applications of Artificial Intelligence, vol. 122, p. 106030, 2023.

[17] A. Desnos and G. Gueguen, Androguard Documentation, 2018.

[18] C. Chen, J. Wei, and Z. Li, "Remaining Useful Life Prediction for Lithium-Ion Batteries Based on a Hybrid Deep Learning Model," Processes, vol. 11, no. 8, p. 2333, 2023.

[19] C. Zhong, G. Li, Z. Meng, H. Li, and W. He, "A Self-Adaptive Quantum Equilibrium Optimizer with an Artificial Bee Colony for Feature Selection," Computers in Biology and Medicine, vol. 153, p. 106520, 2023.

[20] Andro-AutoPsy, "Andro-AutoPsy Project," Accessed: Feb. 12, 2023. [Online]. Available: https://ocslab.hksecurity.net/andro-autopsy

[21] J. W. Jang, H. Kang, J. Woo, A. Mohaisen, and H. K. Kim, "Andro-AutoPsy: Anti-Malware System Based on Similarity Matching of Malware and Malware Creator-Centric Information," Digital Investigation, vol. 14, pp. 17–35, 2015.

[22] H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza, and A. Y. Othman, "Automated Android Malware Detection Using Optimal Ensemble Learning Approach for Cybersecurity," IEEE Access, 2023.\

[23] Kumar, Anuj. "Optimization of Process Parameters in Metal Additive Manufacturing for Defect Reduction Using Machine Learning." International Journal of Engineering Research and Science & Technology, Vol. 14, No. 1, 2018, pp. 41-60