Human Interaction and Emerging Technologies (IHIET-AI 2025), Vol. 161, 2025, 277–283
https://doi.org/10.54941/ahfe1005919

Enhancing Android Security Through Artificial Intelligence: A Hyperparameter-Tuned Deep Learning Approach for Robust Software Vulnerability Detection

Mohammed Assiri

Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, P.O. BOX 16273, Al-Kharj, ZIP 3963, Saudi Arabia

ABSTRACT

Detecting software vulnerabilities is essential for cybersecurity, particularly in Android systems, which are widely used and vulnerable due to their open-source nature. Conventional signature-based malware detection methods are inadequate against sophisticated and evolving threats. This paper introduces a Hyperparameter-Tuned Deep Learning Approach for Robust Software Vulnerability Detection (HPTDLA-RSVD) aimed at enhancing Android security through an optimized deep learning model. The HPTDLA-RSVD methodology encompasses min-max data normalization, feature selection using the Ant Lion Optimizer (ALO), classification via a Deep Belief Network (DBN), and hyperparameter optimization with the Slime Mould Algorithm (SMA). Experimental evaluations on a benchmark dataset reveal that HPTDLA-RSVD surpasses existing techniques across multiple performance metrics, confirming its efficacy in identifying and mitigating software vulnerabilities on Android platforms.

Keywords: Artificial intelligence, Software vulnerability, Cybersecurity, Deep learning, Ant lion optimizer, Hyperparameter tuning

INTRODUCTION

Software vulnerabilities are weaknesses in software systems that malicious actors can exploit to compromise security, steal data, or inflict damage on systems and users (Yadav, 2022). These vulnerabilities may arise from coding errors, design flaws, or configuration issues and can sometimes facilitate attacks on other networks, amplifying the initial compromise (Ullah, 2023). Android, as a widely adopted open-source mobile operating system, offers significant customization and openness. While these features promote extensive application development, they also render Android a primary target for cyberattacks due to inherent security vulnerabilities in the operating system and application development processes (Almomani, 2022). Malicious applications can exploit system calls to access hardware resources and interface between the kernel and applications, often concealing

their activities through encoded techniques that are difficult to detect using standard static analysis methods (Haq, 2021).

The permissive nature of Android allows users to install unofficial third-party applications, extending its reach beyond official marketplaces (Chopra, 2023). The Android permission mechanism requires users to grant permissions manually, which malicious apps can exploit to gain unauthorized access and control without user consent (Kim, 2022). Such malware can cause data theft, file corruption, unwanted advertisements, and even device lockdowns demanding ransom payments (Raymond, 2023).

Traditional signature-based malware detection methods, reliant on databases of known malware characteristics, are inadequate against the proliferation of new malware variants (Yadav, 2022). Advanced techniques utilizing machine learning (ML) and deep learning (DL) have become essential for detecting unknown Android malware embedded in APK files. These approaches extract features such as permissions and API calls from known benign and malicious applications and employ ML algorithms like Random Forest (RF) and Decision Trees (DT) to develop models capable of identifying malicious applications with high accuracy (Al-Andoli, 2023).

This study proposes a Hyperparameter-Tuned Deep Learning Approach for Robust Software Vulnerability Detection (HPTDLA-RSVD) to enhance Android malware security using an optimized deep learning model. The HPTDLA-RSVD technique includes data normalization, feature selection via the Ant Lion Optimizer (ALO), classification using a Deep Belief Network (DBN), and hyperparameter tuning through the Slime Mould Algorithm (SMA). Experimental evaluation on a benchmark dataset demonstrates that the HPTDLA-RSVD method surpasses existing approaches in various performance metrics.

Literature Survey

Several studies have explored deep learning models for Android malware detection, leveraging both static and dynamic analysis techniques. Nasser et al. (2023) developed DL-AMDet, which detects Android malware using dynamic and static features. DL-AMDet comprises two detection architectures: one employing a CNN-BiLSTM model for static analysis and another using a Denoising Autoencoder (DAE) for dynamic analysis. Karbab and Debbabi (2021) introduced PetaDroid, an architecture for accurate Android malware detection and clustering similar malware variants via static analysis. PetaDroid adapts to changes over time and resists binary obfuscation techniques, utilizing methods from natural language processing (NLP) and machine learning. Amer and El-Sappagh (2022) proposed a behavioral Android malware smell predictor system, representing features in API call sequences using clustering techniques. The framework employs an LSTM network for classifying API and system call snapshots and an ensemble machine learning algorithm for classifying Android permissions. Shaukat et al. (2023) presented a deep learning-based approach analyzing portable executable (PE) files as images, integrating deep learning with machine learning to detect malware without extensive feature engineering.

Geremias et al. (2022) developed a multi-view Android malware detection approach using image-based deep learning. The method assesses applications based on multiple feature sets, transforms extracted features into images, and applies deep learning techniques. Albakri et al. (2023) proposed a Rock Hyrax Swarm Optimizer with a deep learning-based Android malware detection technique (RHSODL-AMD), involving API calls and essential permissions identification, feature selection, and an attention recurrent autoencoder (ARAE) model for malware detection. Ravi et al. (2022) introduced a multi-view attention-based deep learning method analyzing features like API calls and PE-Imports for malware detection. Evaluations demonstrated the approach's effectiveness.

Raphael and Mathiyalagan (Raphael, 17) implemented an intelligent hyperparameter-tuned deep learning-based malware detection model (IHPT-DLMD), involving feature selection using a binary coyote optimization algorithm, a bidirectional LSTM model, and hyperparameter tuning with the glowworm swarm optimization algorithm. These studies highlight the potential of deep learning models in enhancing Android malware detection accuracy, though challenges remain in optimizing these models for improved performance and computational efficiency.

The Proposed Method

We propose the Hyperparameter-Tuned Deep Learning Approach for Robust Software Vulnerability Detection (HPTDLA-RSVD) to enhance Android malware security through an optimized deep learning model. The HPTDLA-RSVD technique consists of Data Normalization using Min-Max Scaling, Feature Selection using the Ant Lion Optimizer (ALO), Classification using a Deep Belief Network (DBN), and Hyperparameter Tuning using the Slime Mould Algorithm (SMA)

Data Normalization

Data normalization scales feature values to a uniform range, reducing the impact of differing magnitudes among features. The min-max normalization method scales input data into the range [0, 1], enhancing feature input uniformity and aiding the learning algorithm's resilience to outliers and data variations (Li, 2020).

Feature Selection Using Ant Lion Optimizer (ALO)

Feature selection reduces dimensionality, improves model performance, and minimizes computational cost. The HPTDLA-RSVD technique employs the Ant Lion Optimizer (ALO) for feature selection (Risma, 2023), simulating antlions' hunting behavior, where ants represent potential feature subsets. The ALO process involves:

- Initialization: Randomly generate ant (feature subset) and antlion populations.
- Random Walk of Ants: Simulate ants' random walk to explore the search space.

- Ants Trapped by Antlions: Antlions influence ants' movement, guiding them to promising regions.
- Updating Antlions: Update antlions based on the fitness of trapped ants.
- Fitness Function: Balances maximizing classification accuracy and minimizing selected features:

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|}$$

where $\gamma_R(D)$ is the classification error rate, $|R|$ is the selected feature subset size, $|C|$ is the total feature count, and $\alpha + \beta = 1$

Classification Using Deep Belief Network (DBN)

A Deep Belief Network (DBN) is a generative model consisting of multiple layers of hidden units with connections between layers (Yin, 2023). DBNs effectively model complex, high-dimensional data, making them suitable for malware detection.

The DBN in HPTDLA-RSVD is built by stacking Restricted Boltzmann Machines (RBMs), trained in two phases:

- Unsupervised Pre-training: Each RBM is trained layer-wise to capture data distribution.
- Supervised Fine-tuning: The entire network is fine-tuned using backpropagation to minimize classification error.

The energy function of an RBM is:

$$E(v, h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} s_i s_j - \sum_{i=1}^{n} b_i v_i - \sum_{j=1}^{m} c_i h_i$$

where v and h are visible and hidden units, w_{ij} are weights, and b_i, c_j are biases.

Hyperparameter Tuning Using Slime Mould Algorithm (SMA)

Hyperparameter tuning optimizes deep learning models' performance. HPTDLA-RSVD utilizes the Slime Mould Algorithm (SMA) for hyperparameter optimization (Rifat, 2023), inspired by slime moulds' oscillatory foraging behavior.

In hyperparameter tuning, SMA adjusts parameters like learning rate and hidden units to minimize classification error:

- Initialization: Randomly generate hyperparameter sets within bounds.
- Fitness Evaluation: Evaluate each set using the classification error rate.
- Update Mechanism: Update solutions based on best-found solutions and a random walk simulating slime mould propagation.
- Termination: Repeat until convergence criteria are met, such as maximum iterations or a satisfactory error rate.

The fitness function for SMA is:

$$\begin{aligned} fitness\,(x_i) &= ClassifierErrorRate\,(x_i) \\ &= \frac{Number\ of\ Misclassified\ Instances}{\text{Total Number of Instances}} \times 100 \end{aligned}$$

Experimental Evaluation

We evaluated the HPTDLA-RSVD technique using the AndroAutoPsy dataset (AndroAutoPsy), comprising 7,500 instances as Benign: 5,000 instances, and Malware: 2,500 instances. Features include API calls, permissions, and system calls. Performance metrics used: Accuracy (Acc), Precision (Prec), Recall (Rec), F1-Score (F1), and Matthews Correlation Coefficient (MCC).

Results and Discussion

The HPTDLA-RSVD technique was assessed under different epochs to analyze performance over training iterations. At 1,000 epochs, it achieved the highest performance:

Table 1. Performance metrics of HPTDLA-RSVD technique at 1,000 epochs.

Accuracy	Precision	Recall	F1-Score	MCC
99.04%	98.95%	99.04%	99.00%	97.99%

The confusion matrix at 1,000 epochs showed effective detection of both benign and malicious samples. Precision-recall and ROC curves further illustrated strong classification performance.

Comparative Analysis

We compared HPTDLA-RSVD with other algorithms, including J48, Random Forest, Decision Table, Multilayer Perceptron, Logistic Model, AdaBoostM1, and AAMD-OELAC (Alamro, 2023).

Table 2. Performance evaluation of HPTDLA-RSVD in comparison to other algorithms.

Algorithm	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
J48	96.92	95.33	97.55	97.37
RF	97.91	96.70	97.35	96.74
DT	94.72	91.69	97.64	97.88
Multilayer Perceptron	98.21	97.21	98.02	98.28
Logistic Model	96.39	94.50	97.86	96.67
AdaBoostM1	88.51	81.83	91.94	94.38
AAMD-OELAC	98.97	98.27	98.44	98.54
HPTDLA-RSVD	99.04	98.95	99.04	99.00

HPTDLA-RSVD outperformed all methods in accuracy, precision, recall, and F1-score, demonstrating superior malware detection capability. In computational efficiency, HPTDLA-RSVD also showed advantages, requiring the least computational time.

CONCLUSION

This study presented the Hyperparameter-Tuned Deep Learning Approach for Robust Software Vulnerability Detection (HPTDLA-RSVD) to enhance Android malware detection. The HPTDLA-RSVD technique integrates:

- Min-Max Data Normalization: Ensures input features are on a similar scale.
- Feature Selection using Ant Lion Optimizer (ALO): Selects relevant features, reducing dimensionality.
- Classification using Deep Belief Network (DBN): Employs a deep learning model capturing complex patterns.
- Hyperparameter Tuning using Slime Mould Algorithm (SMA): Optimizes hyperparameters for better accuracy.

Experimental evaluations demonstrated that HPTDLA-RSVD outperforms existing methods in accuracy, precision, recall, F1-score, and computational efficiency, effectively detecting and classifying Android malware. Future work may involve extending HPTDLA-RSVD to other malware types and exploring real-time detection scenarios.

REFERENCES

Al-Andoli, M. N., Sim, K. S., Tan, S. C., Goh, P. Y. and Lim, C. P., 2023. An ensemble-based parallel deep learning classifier with PSO-BP optimization for malware detection. IEEE Access.

Alamro, H., Mtouaa, W., Aljameel, S., Salama, A. S., Hamza, M. A. and Othman, A. Y., 2023. Automated android malware detection using optimal ensemble learning approach for cybersecurity. IEEE Access.

Albakri, A., Alhayan, F., Alturki, N., Ahamed, S. and Shamsudheen, S., 2023. Metaheuristics with Deep Learning Model for Cybersecurity and Android Malware Detection and Classification. Applied Sciences, 13(4), p. 2172.

Almomani, I., Alkhayer, A. and El-Shafai, W., 2022. An automated vision-based deep learning model for efficient detection of Android malware attacks. IEEE Access, 10, pp. 2700–2720.

Amer, E. and El-Sappagh, S., 2022. Robust deep learning early alarm prediction model based on the behavioural smell for android malware. Computers & Security, 116, p. 102670.

AndroAutoPsy Dataset. OCS Lab. Retrieved from https://ocslab.hksecurity.net/andro-autopsy.

Chopra, R., Acharya, S., Rawat, U. and Bhatnagar, R., 2023. An energy efficient, robust, sustainable, and low computational cost method for mobile malware detection. Applied Computational Intelligence and Soft Computing, 2023.

Geremias, J., Viegas, E. K., Santin, A. O., Britto, A. and Horchulhack, P., 2022. Towards multi-view android malware detection through image-based deep learning. In 2022 International Wireless Communications and Mobile Computing (IWCMC) (pp. 572–577). IEEE.

Haq, I. U., Khan, T. A. and Akhunzada, A., 2021. A dynamic robust DL-based model for android malware detection. IEEE Access, 9, pp. 74510–74521.

Karbab, E. B. and Debbabi, M., 2021. Petadroid: adaptive android malware detection using deep learning. In Detection of Intrusions and Malware, and Vulnerability Assessment: 18th International Conference, DIMVA 2021, Virtual Event, July 14–16, 2021, Proceedings 18 (pp. 319–340). Springer International Publishing.

Kim, J., Ban, Y., Ko, E., Cho, H. and Yi, J. H., 2022. MAPAS: A practical deep learning-based android malware detection system. International Journal of Information Security, 21(4), pp. 725–738.

Li, H., Zhao, W., Zhang, Y. and Zio, E., 2020. Remaining useful life prediction using multi-scale deep convolutional neural network. Applied Soft Computing, 89, p. 106113.

Nasser, A. R., Hasan, A. M. and Humaidi, A. J., 2023. DL-AMDet: Deep Learning-Based Malware Detector for Android. Intelligent Systems with Applications, p. 200318.

Raphael, R. and Mathiyalagan, P., 2023. Intelligent hyperparameter-tuned deep learning-based android malware detection and classification model. Journal of Circuits, Systems and Computers, p. 2350191.

Raymond, V. J., Raj, R. and Retna, J., 2023. Investigation of Android Malware Using Deep Learning Approach. Intelligent Automation and Soft Computing, 35(2).

Ravi, V., Alazab, M., Selvaganapathy, S. and Chaganti, R., 2022. A multi-view attention-based deep learning framework for malware detection in smart healthcare systems. Computer Communications, 195, pp. 73–81.

Rifat, M. S. H., Niloy, M. A., Rizvi, M. F., Ahmed, A., Ahshan, R., Nengroo, S. H. and Lee, S., 2023. Application of binary slime mould algorithm for solving unit commitment problem. IEEE Access.

Risma, Y. M. and Utama, D. M., 2023. AVOA and ALO algorithm for energy-efficient no-idle permutation flow shop scheduling problem: A comparison study. Jurnal Optimasi Sistem Industri, 22(2), pp. 126–141.

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. Engineering Applications of Artificial Intelligence, 122, p. 106030.

Ullah, F., Ullah, S., Srivastava, G., Lin, J. C. W. and Zhao, Y., 2023. NMal-Droid: Network-based android malware detection system using transfer learning and CNN-BiGRU ensemble. Wireless Networks, pp. 1–22.

Yadav, P., Menon, N., Ravi, V., Vishvanathan, S. and Pham, T. D., 2022. A two-stage deep learning framework for image-based android malware detection and variant classification. Computational Intelligence, 38(5), pp. 1748–1771.

Yadav, P., Menon, N., Ravi, V., Vishvanathan, S. and Pham, T. D., 2022. EfficientNet convolutional neural networks-based android malware detection. Computers & Security, 115, p. 102622.

Yin, X., Huang, X., Pan, Y. and Liu, Q., 2023. Point and interval estimation of rock mass boreability for tunnel boring machine using an improved attribute-weighted deep belief network. Acta Geotechnica, 18(4), pp. 1769–1791.