An Artificial Intelligence Driven Optimal Deep Belief Network Model for Malware Classification on IoT-Cloud Environment

Khalid Ammar
Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, United Arab Emirates
k.ammar@ajman.ac.ae

Mohamad Khairi Ishak
Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, United Arab Emirates
m.ishak@ajman.ac.ae (corresponding author)

Received: 7 August 2025 | Revised: 9 September 2025 and 17 September 2025 | Accepted: 18 September 2025

 DOI: https://doi.org/10.48084/etasr.13929

ABSTRACT

Computer-based systems, including mobile devices, desktops, Internet of Things (IoT), and Cyber-Physical Systems (CPS), are designed to protect data effectively. However, malware targets these systems, threatening data accessibility, integrity, and confidentiality through cyberattacks. This study proposes an Artificial Intelligence-driven Optimal Deep Belief Network for Malware Detection and Classification (AIODBN-MDC) approach, aiming to detect and classify malware in IoT-based cloud infrastructure. Initially, z-score normalization is performed to scale the data in a standard form. Then, a Bottleneck-Driven DBN (BDDBN) model is utilized to detect and classify the malware. Finally, the Enhanced Grasshopper Optimization Algorithm (EGOA) model is employed to fine-tune the hyperparameters of the BDDBN classifier. Experimental investigation of the proposed AIODBN-MDC technique on an Android malware dataset demonstrated an accuracy of 99.34%, outperforming existing methods.

Keywords-malware detection; machine learning; Internet of Things (IoT); security; parameter adjustment; cloud computing

I. INTRODUCTION

In recent times, the application of interconnected smart devices, generally known as IoT, has witnessed tremendous development [1]. IoT devices can be accessed from any location, such as offices, vehicles, and homes, making day-to-day tasks simpler. These kinds of smart devices are used in vehicular networks, medical services, industries, smart cities, smart grids, and smart homes [2]. These smart devices feature unique characteristics, including lighter protocols, minimal energy consumption, and compact sizes, which enhance their adaptability [3, 4]. However, there is no confirmed security system to ensure the digital security of these devices. IoT devices are prone to security threats and various attacks, as they lack built-in conventional support systems and security mechanisms, becoming a vulnerable platform for intruders, who can launch various types of network attacks [6]. Cloud resources are shared, and tasks are delivered to maximize the use, location-independent accessibility, and virtualization of physical resources. Security risks in IoT-cloud platforms arise during data processing and operation, requiring secure connectivity, reliable access, and effective data management to address threats like malware [7]. Malware is a considerable threat to advanced computing devices for its illicit purposes [8]. Efficient malware detection requires effective tools and techniques. A common approach is behaviour-based detection, which analyses system call patterns. However, behavior-based malware detection approaches are inferior for detecting hidden and continually changing malware due to their complex and restrictive nature.

In [9], a soft-voting ensemble machine learning technique used Pearson's correlation coefficient to select features and SMOTE to balance the dataset. In [10], a DL-based Bidirectional-GRU-CNN (BiGRU-CNN) technique was developed for malware detection. In [11], a novel threat intelligence model was based on DL methods, including a deep pattern extractor to detect and classify threats. In [12], a Dual-Channel-CNN used the Spider Monkey Optimizer (DCCNN-SMO). In [13], a Federated Malware Detection (FED-MAL) method was presented. In [14], an innovative network forensics model was introduced, using PSO-based parameter optimization for a DNN to detect abnormal IoT smart home activities. In [15], Swarm Intelligence (SI) was used with a

DNN. In [16], an artificial, fully automatic Intrusion Detection System (IDS) was proposed to improve security against cyberattacks. In [17], a hybrid approach utilized LSTM and CNN models for multiclass attack classification. In [18], the Optimal Bottleneck driven Deep Belief Network-enabled Cybersecurity Malware Classification (OBDDBN-CMC) technique utilized z-score normalization and the Grasshopper Optimization Algorithm (GOA) to tune hyperparameters. This model outperformed others such as Incremental Naïve Bayes (INB), Support Vector Machine (SVM), Two-layer Deep Learning Adaptive Multi-scale Deep Neural Transformer (DL-AMDNT), K-Nearest Neighbors (KNN), and Enhanced Adaptive Multi-Scale Deep Neural Forest (EAMD-NF). Despite advances in DL and swarm intelligence models, many approaches lack effective integration of feature optimization and tuning, restricting adaptability. The research gap is the absence of unified frameworks integrating deep feature learning with robust optimization to improve detection accuracy.

This study proposes an Artificial Intelligence-driven Optimal Deep Belief Network for Malware Detection and Classification (AIODBN-MDC) approach, with the following contributions:

- Z-score normalization is applied to standardize the input features, ensuring a consistent data distribution and enhancing training stability and convergence. This process also improves the model's ability to learn efficiently from varying input scales.
- The BDDBN technique is utilized for extracting intrinsic and informative deep features. This enables more accurate malware detection and classification and improves feature learning and model generalization.
- The EGOA method is implemented to fine-tune hyperparameters. This also ensures optimal performance by navigating intrinsic search spaces. It also improves detection accuracy and mitigates the risk of overfitting.
- The AIODBN-MDC approach uniquely incorporates deep feature extraction with intelligent parameter tuning, significantly improving the accuracy and efficiency of malware detection. Its novelty lies in utilizing both DL and metaheuristic optimization in a unified model.

II. THE PROPOSED MODEL

The proposed AIODBN-MDC approach comprises data normalization, BDDBN-based malware detection, and EGOA-based tuning, as shown in Figure 1.

A. Data Normalization

Data normalization transforms data so that they have zero mean and standard deviation equal to one. This method is chosen for its efficiency in standardizing features by centering them around zero with unit variance. Unlike min-max scaling, normalization is less sensitive to outliers, resulting in more consistent learning and performance across diverse datasets. This is commonly used to compare and analyze variables that have diverse scales or units of measurement. The z-score of a data point x is computed by subtracting the dataset's mean and dividing by its standard deviation.

$$z = (x - \mu) / \sigma \quad (1)$$

where x denotes the data point, μ implies the mean of the dataset, and σ is the standard deviation of the dataset. This normalization technique enables easy analysis and comparison of variables with dissimilar scales.

B. Malware Detection Using BDDBN

The BDDBN model is employed for automated malware detection [18]. This model is chosen for its robust capability in learning hierarchical and abstract feature representations through deep layers. This method also mitigates dimensionality, improving detection efficiency and mitigating computational load. This model offers enhanced accuracy and generalization in complex malware patterns compared to conventional DBNs or shallow models. DBN is a conceptual paradigm for learning models with deep structures (different layers of nonlinear arithmetical units). Compared to prior modelling methods for shallow structures (such as individual layers of nonlinear arithmetical units), this model has stronger modelling ability while dealing with real-time datasets (i.e., images, video, and natural speech). DBN is a multilayer ANN that utilizes a combination of supervised and unsupervised training to determine the network parameters, thereby resolving the challenges of models that can easily get stuck in local optima. A DBN is constructed by retraining and fine-tuning an ANN. DBN is a series of RBM cascades. An RBM includes the hidden layer and visible layer (h_j and v_j) and the joint distribution for the sequence of parameters is expressed as:

$$E(v_i, h_j; \theta) = -\sum_{ij} w_{ij}\, v_j h_j - \sum_i b_i\, v_i - \sum_j a_j\, h_j \quad (2)$$

where $\theta = \{w, a, b\}$ and w_{ij} denote the weight connections of the VL and HL. b_j and a_j represent the bias vectors, respectively.

$$P_\theta(v, h) = \frac{1}{Z(\theta)} \exp\left(-E(v, h; \theta)\right)$$

$$= \frac{1}{Z(\theta)} \prod_{ij} e^{w_{ij} v_i h_j} \prod_i e^{b_i v_j} \prod_j e_j^{ah} \quad (3)$$

$$P(h|v) = \prod_j P\left(h_j \middle| v\right) \quad (4)$$

It is easier to obtain the probability that the j^{th} node of the hidden unit is 1 or 0 than the v visible layer:

$$P(h_j = 1|v) = \frac{1}{1+\exp\left(-\sum_i w_{ij} v_i - a_j\right)}$$

$$P(v|h) = \prod_i P\left(v_j \middle| h\right) \quad (5)$$

$$P(v_j = 1|h) = \frac{1}{1+\exp\left(-\sum_j w_{ij} h_j - b_i\right)}$$

Maximizing the log-likelihood function:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(v) - \frac{\lambda}{N} \|w\|_F^2 \quad (6)$$

$$\frac{\partial L(\theta)}{\partial w_{ij}} = E_{P_{data}}\left[v_i h_j\right] - E_{P_\theta}\left[v_i h_j\right] - \frac{2\lambda}{N} w_{ij} \quad (7)$$

Comparable to a conventional BPNN, a supervised learning model is used to construct the DBN.

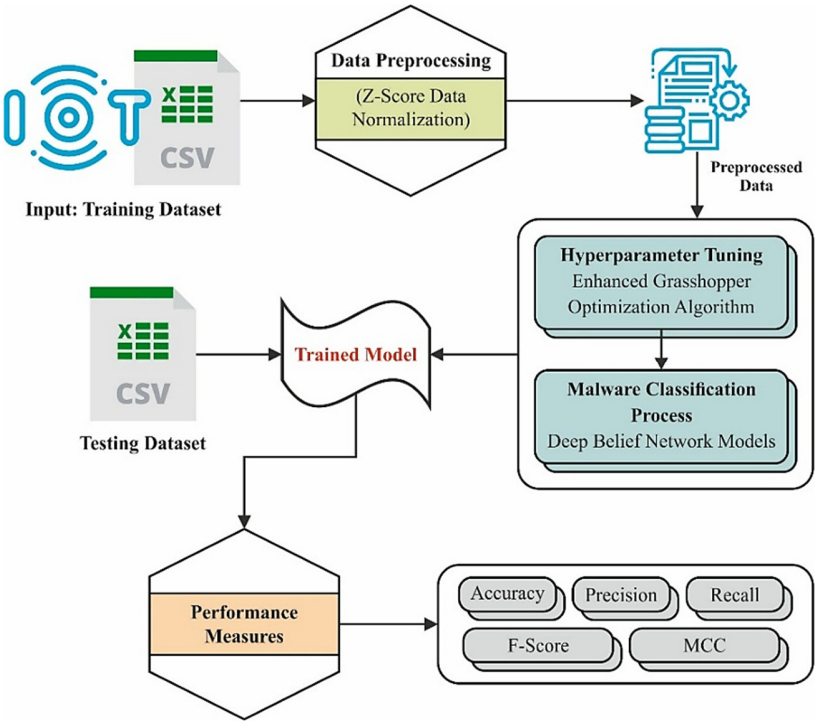

Fig. 1. Working flow of the AIODBN-MDC model.

C. EGOA-Based Parameter Tuning

The EGOA is employed to fine-tune the hyperparameters of the BDDBN model [19]. This method is chosen for its improved exploration and exploitation capabilities, allowing it to navigate intrinsic search spaces efficiently. Unlike standard optimization methods, EGOA avoids premature convergence. This results in more accurate and robust tuning of model parameters, improving overall detection performance.

The GOA simulates the social performance and hunting approach of grasshoppers. The swarming performance of grasshoppers is a mathematical process utilized for calculating the position X_i of all the solutions. The following formula determines the performance of the grasshopper swarm.

$$X_i^d = c_1\left(\sum_{j=1,i\neq j}^{N} c_2 \frac{UB_d - LB_d}{2} s\left(\left|x_j^d - x_i^d\right|\right)\frac{|x_j - x_i|}{d_{ij}}\right) + \widehat{T_d} \quad (8)$$

$$s = fe^{\frac{-r}{l}} - e^{-r} \quad (9)$$

where s implies the power of two social forces, repulsion and attraction, among grasshopper swarms, r and $d_{ij}\left(= \left|x_j - x_i\right|\right)$ represent the Euclidean distance, $\widehat{d_{ij}} = \frac{|x_j - x_i|}{d_{ij}}$ denotes the unit vector from the i^{th} to the j^{th} grasshopper swarm, and f denotes the power of attraction, where l denotes the length of attraction. UB_d and LB_d are the upper and lower bounds from the d^{th} dimension of the searching space. The term $\widehat{T_d}$ represents the target of the d-dimensional space, and N denotes the number of grasshoppers.

The grasshopper's position is upgraded based on its current position and the global optimum. For balancing exploration and exploitation, the reduction of safe places is defined by a shrinking feature c as:

$$c = c_{\max} - t\frac{c_{\max} - c_{\min}}{t_{\max}} \quad (10)$$

where $c_{\max}$ and $c_{\min}$ represent the maximal and minimal values of c, proposed as 1 and 0.00001, respectively. Here, t denotes the current iteration, and $t_{\max}$ denotes the maximum iteration value.

EGOA is improved by using a Random Weight (RW) approach to address limitations in the original GOA, such as slow convergence and getting trapped in local optima. By replacing the linearly decreasing control parameter c in (10) with a nonlinear adaptive coefficient, the RW method improves both exploration and exploitation, improving GOA's performance in solving complex optimization problems. In the RW approach, the nonlinear adaptive coefficient c upgrade is given by:

$$c = \begin{cases} c_{\max} - (c_{\max} - c_{\min}) \times k \times \left(1 + \left(cos\left(\frac{pi \times l}{L}\right)^2\right)\right), \\ l \leq 0.5 \times L, \\ c_{\max} - (c_{\max} - c_{\min}) \times k \times \left(1 - \left(cos\left(\frac{pi \times l}{L}\right)^2\right)\right), \\ 0.5 \times L < l \leq L \end{cases} \quad (11)$$

where k denotes the constant value in [0, 1], and l and L imply the value of the present and maximal iteration counts.

The parameter c in (11) not only enhances the search choice of GOA but also improves population diversity. Optimal fitness is a vital feature of the EGOA approach. An encoding result is employed to develop a good candidate. In this study, the precision rate is used as the fitness function.

$$Fitness = \max(P) \quad (12)$$

$$P = \frac{TP}{TP+FP} \quad (13)$$

where FP and TP denote the false and true positive results.

III. PERFORMANCE VALIDATION

The dataset [20] comprises 9419 samples and two class labels, as shown in Table I. This section presents the results of the AIODBN-MDC model on this dataset.

TABLE I. DATASET DESCRIPTION

Class	Samples
Benign	5065
Malware	4354
Total	**9419**

Figure 2 shows the confusion matrices of the AIODBN-MDC approach on malware recognition, indicating that the AIODBN-MDC method effectively detects and classifies malware.

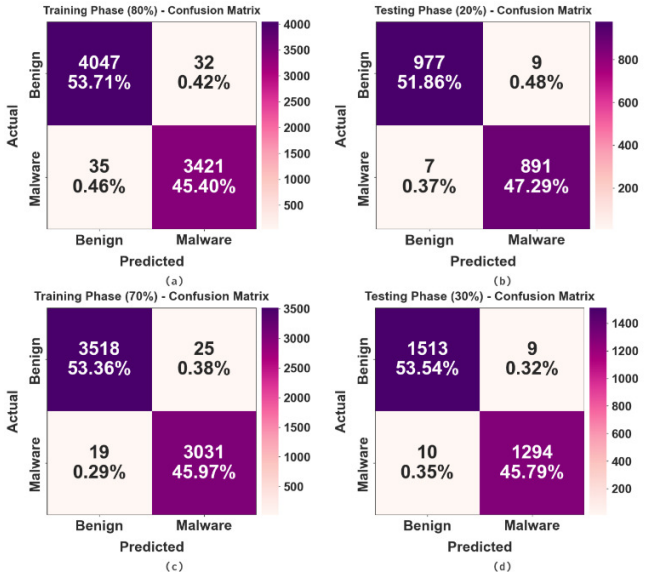

Fig. 2. Confusion matrices: (a, b) 80% TRSE and 20% TSSE, (c,d) 70% TRSE and 30% TSSE.

Table II presents the malware detection results of the AIODBN-MDC model under an 80:20 TRSE/TSSE, demonstrating that it appropriately categorized two classes. On 80% TRSE, the AIODBN-MDC technique obtained an average $accu_y$ of 99.10%, $prec_n$ of 99.11%, $reca_l$ of 99.10%, F_{score} of 99.10%, and MCC of 98.21%. On 20% TSSE, the AIODBN-MDC technique achieved an average $accu_y$ of 99.15%, $prec_n$ of 99.14%, $reca_l$ of 99.15%, F_{score} of 99.15%, and MCC of 98.30%.

TABLE II. MALWARE DETECTION WITH AIODBN-MDC ON 80% TRSE AND 20% TSSE

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRSE (80%)					
Benign	99.22	99.14	99.22	99.18	98.21
Malware	98.99	99.07	98.99	99.03	98.21
Average	**99.10**	**99.11**	**99.10**	**99.10**	**98.21**
TSSE (20%)					
Benign	99.09	99.29	99.09	99.19	98.30
Malware	99.22	99.00	99.22	99.11	98.30
Average	**99.15**	**99.14**	**99.15**	**99.15**	**98.30**

Table III illustrates the malware detection performance of the AIODBN-MDC method under 70:30 of TRSE/TSSE. On 70% the TRSE, the AIODBN-MDC model achieved an average $accu_y$ of 99.34%, $prec_n$ of 99.32%, $reca_l$ of 99.34%, F_{score} of 99.33%, and MCC of 98.66%. On 30% of TSSE, the AIODBN-MDC model achieved an average $accu_y$ of 99.32%, $prec_n$ of 99.33%, $reca_l$ of 99.32%, F_{score} of 99.32%, and MCC of 98.65%. Figure 3 shows the training and validation accuracy per epoch of the AIODBN-MDC approach at 70:30 TRSE/TSSE. Both training and validation accuracy improve with growing epochs, showing effective learning without overfitting.

TABLE III. MALWARE DETECTION WITH AIODBN-MDC AT 70% TRSE AND 30% TSSE

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
TRSE (70%)					
Benign	99.29	99.46	99.29	99.38	98.66
Malware	99.38	99.18	99.38	99.28	98.66
Average	**99.34**	**99.32**	**99.34**	**99.33**	**98.66**
TSSE (30%)					
Benign	99.41	99.34	99.41	99.38	98.65
Malware	99.23	99.31	99.23	99.27	98.65
Average	**99.32**	**99.33**	**99.32**	**99.32**	**98.65**

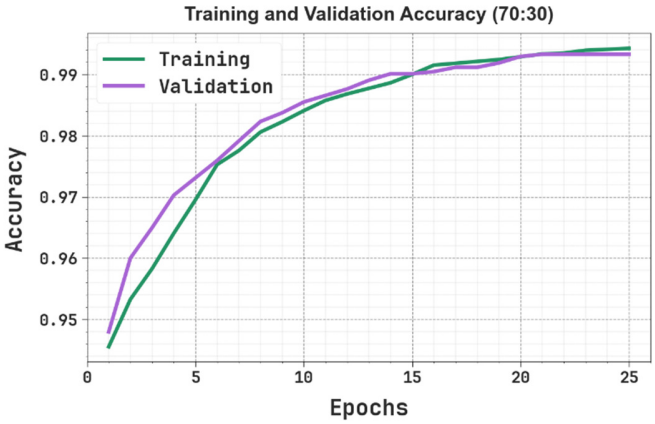

Fig. 3. $Accu_y$ curve AIODBN-MDC on 70% TRSE.

Figure 4 shows the training and validation loss curves of the AIODBN-MDC method at 70:30 of the TRSE/TSSE. Training loss measures the error between the predicted and actual output in training data, while validation loss measures

the performance of the AIODBN-MDC technique in validation data. Both metrics decrease with epochs, indicating improved accuracy and efficiency.

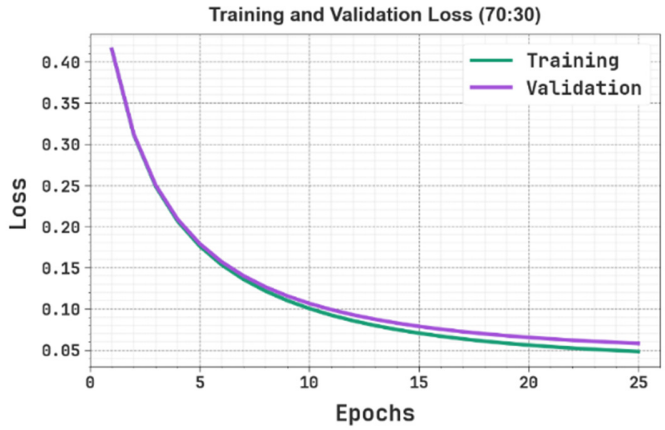

Fig. 4. Loss curve of AIODBN-MDC on 70%TRSE.

Table IV demonstrates a comparison of the AIODBN-MDC technique with existing models. The AIODBN-MDC technique achieved a higher $accu_y$ of 99.32%, while the OBDDBN-CMC, INB, SVM, two-layer DL-AMDNT, KNN, and EAMD-NF models achieved 99.04%, 97.81%, 95.18%, 93.98%, 92%, and 87.46%, correspondingly. Based on F_{score}, the AIODBN-MDC approach achieved 99.32% while the OBDDBN-CMC, INB, SVM, two-layer DL-AMDNT, KNN, and EAMD-NF models achieved 99.04%, 97.64%, 97.96%, 94.66%, 90.83%, and 88.48% respectively. These results show that the proposed AIODBN-MDC outperformed existing methods.

TABLE IV. COMPARISON OF AIODBN-MDC WITH EXISTING TECHNIQUES

Method	$Accu_y$	F_{score}
AIODBN-MDC (Proposed)	99.32	99.32
OBDDBN-CMC [18]	99.04	99.04
INB [18]	97.81	97.64
SVM [18]	95.18	97.96
Two-Layer DL-AMDNT [18]	93.98	94.66
KNN Model [18]	92.00	90.83
EAMD-NF [18]	87.46	88.48

IV. CONCLUSION

This study presents a novel AIODBN-MDC approach to detect and classify malware through three phases: data normalization, BDDBN-based malware detection, and EGOA-based parameter tuning. Initially, the z-score was used for normalization. Then, the BDDBN approach was utilized for detecting malware. Finally, EGO was applied for effectual fine-tuning. The evaluation results demonstrate the greater efficiency of the AIODBN-MDC method compared to other approaches. However, the limitations of the proposed method comprise limited evaluation across diverse real-world datasets, which may affect generalizability, and a lack of integration with real-time detection systems, impacting practical deployment. Future work may explore broader dataset validation, real-time implementation, and improved adaptability to growing threats.

ACKNOWLEDGMENT

The authors extend their appreciation to Ajman University, UAE, for supporting this research work.

REFERENCES

[1] M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, "An inception V3 approach for malware classification using machine learning and transfer learning," *International Journal of Intelligent Networks*, vol. 4, pp. 11–18, Jan. 2023, https://doi.org/10.1016/j.ijin.2022.11.005.

[2] A. Anand, S. Rani, D. Anand, H. M. Aljahdali, and D. Kerr, "An Efficient CNN-Based Deep Learning Model to Detect Malware Attacks (CNN-DMA) in 5G-IoT Healthcare Applications," *Sensors*, vol. 21, no. 19, Jan. 2021, Art. no. 6346, https://doi.org/10.3390/s21196346.

[3] S. M. Alshahrani *et al.*, "IoT-Cloud Assisted Botnet Detection Using Rat Swarm Optimizer with Deep Learning," *Computers, Materials and Continua*, vol. 74, no. 2, pp. 3085–3100, Oct. 2022, https://doi.org/10.32604/cmc.2023.032972.

[4] M. Shobana and S. Poonkuzhali, "A Novel Approach for Detecting IoT Botnet Using Balanced Network Traffic Attributes," in *Service-Oriented Computing – ICSOC 2020 Workshops*, Dubai, United Arab Emirates, 2021, pp. 534–548, https://doi.org/10.1007/978-3-030-76352-7_48.

[5] G. A. A. Mary, B. Sathyasri, K. Murali, L. A. J. Prabhu, and N. Bharatha Devi, "Electrocardiogram signal classification in an IoT environment using an adaptive deep neural networks," *Neural Computing and Applications*, vol. 35, no. 21, pp. 15333–15342, Jul. 2023, https://doi.org/10.1007/s00521-023-08534-9.

[6] S. M. T. Nizamudeen, "Intelligent intrusion detection framework for multi-clouds – IoT environment using swarm-based deep learning classifier," *Journal of Cloud Computing*, vol. 12, no. 1, Sep. 2023, Art. no. 134, https://doi.org/10.1186/s13677-023-00509-4.

[7] A. Al-Marghilani, "Comprehensive Analysis of IoT Malware Evasion Techniques," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7495–7500, Aug. 2021, https://doi.org/10.48084/etasr.4296.

[8] X. Deng, B. Chen, X. Chen, X. Pei, S. Wan, and S. K. Goudos, "A Trusted Edge Computing System Based on Intelligent Risk Detection for Smart IoT," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1445–1454, Oct. 2024, https://doi.org/10.1109/TII.2023.3245681.

[9] M. Jumaah, A. A. Yassin, Z. A. Abduljabbar, M. Jawad, V. O. Nyangaresi, and A. H. Ali, "Amalgamating Ensemble Machine Learning Soft Voting Classifier, SMOTE, and Pearson's Correlation Coefficient for Enhanced Malware Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 22746–22752, Jun. 2025, https://doi.org/10.48084/etasr.10420.

[10] R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture internet of things malware detection and classification," *Computers & Security*, vol. 120, Sep. 2022, Art. no. 102779, https://doi.org/10.1016/j.cose.2022.102779.

[11] M. Al-Hawawreh, N. Moustafa, S. Garg, and M. S. Hossain, "Deep Learning-Enabled Threat Intelligence Scheme in the Internet of Things Networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 2968–2981, Jul. 2021, https://doi.org/10.1109/TNSE.2020.3032415.

[12] P. Vijayalakshmi and D. Karthika, "Hybrid dual-channel convolution neural network (DCCNN) with spider monkey optimization (SMO) for cyber security threats detection in internet of things," *Measurement: Sensors*, vol. 27, Jun. 2023, Art. no. 100783, https://doi.org/10.1016/j.measen.2023.100783.

[13] M. Abdel-Basset, H. Hawash, K. M. Sallam, I. Elgendi, K. Munasinghe, and A. Jamalipour, "Efficient and Lightweight Convolutional Networks for IoT Malware Detection: A Federated Learning Approach," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 7164–7173, Apr. 2023, https://doi.org/10.1109/JIOT.2022.3229005.

[14] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, Sep. 2020, https://doi.org/10.1016/j.future.2020.03.042.

[15] M. Abd Elaziz, M. A. A. Al-qaness, A. Dahou, R. A. Ibrahim, and A. A. A. El-Latif, "Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm," *Advances in Engineering Software*, vol. 176, Feb. 2023, Art. no. 103402, https://doi.org/10.1016/j.advengsoft.2022.103402.

[16] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, May 2020, Art. no. 102031, https://doi.org/10.1016/j.simpat.2019.102031.

[17] K. Kalaivani and M. Chinnadurai, "A Hybrid Deep Learning Intrusion Detection Model for Fog Computing Environment," *Intelligent Automation & Soft Computing*, vol. 29, no. 3, pp. 1–15, 2021, https://doi.org/10.32604/iasc.2021.017515.

[18] M. Maray *et al.*, "Optimal Bottleneck-Driven Deep Belief Network Enabled Malware Classification on IoT-Cloud Environment," *Computers, Materials and Continua*, vol. 74, no. 2, pp. 3101–3115, Oct. 2022, https://doi.org/10.32604/cmc.2023.032969.

[19] L. Wu, J. Wu, and T. Wang, "The improved grasshopper optimization algorithm with Cauchy mutation strategy and random weight operator for solving optimization problems," *Evolutionary Intelligence*, vol. 17, no. 3, pp. 1751–1781, Jun. 2024, https://doi.org/10.1007/s12065-023-00861-z.

[20] "Android Malware Dataset for Machine Learning." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/shashwatwork/android-malware-dataset-for-machine-learning.