The Gorilla Troops Optimizer-Based Ensemble Deep Learning Model for Real-Time Zero-Day Attack Detection and Classification

J. Vanitha
Department of Computer and Information Science, Faculty of Science, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu, India
vanithahenri@gmail.com (corresponding author)

P. Anandababu
Department of Computer and Information Science, Faculty of Science, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu, India
drpabcud@gmail.com

Received: 17 May 2025 | Revised: 16 June 2025 and 4 July 2025 | Accepted: 6 July 2025

 DOI: https://doi.org/10.48084/etasr.12210

ABSTRACT

Securing computer networks is becoming increasingly important and gaining significant attention. Security attacks, especially zero-day attacks, pose significant risks to enterprise and critical networks due to their unknown vulnerabilities and detection challenges. To ensure cybersecurity in networks, Intrusion Detection Systems (IDSs) observe network traffic for malicious actions and related attacks. Deep Learning (DL) and Machine Learning (ML)-based IDS are widely adopted for their adaptability and robust detection capabilities, particularly against zero-day attacks. This study presents the Gorilla Troops Optimizer-based Ensemble DL Model for Zero-Day Attack Detection (GTOEDLM-ZDAD) technique, aimed at classifying and detecting zero-day attacks using ensemble and advanced optimization algorithms. Initially, Linear Scaling Normalization (LSN) is used, and the Chimpanzee Optimization Algorithm (ChoA) is utilized for feature subset selection. An ensemble DL model uses Deep Q-Network (DQN), Bidirectional Gated Recurrent Unit (BiGRU), and Deep Belief Network (DBN) for classification. Finally, Gorilla Troops Optimizer (GTO)-based hyperparameter tuning is performed. A wide range of experimentation of the GTOEDLM-ZDAD technique on the ToN-IoT dataset achieved a superior accuracy of 98.33% over existing approaches and baseline models.

Keywords-Gorilla Troops Optimizer (GTO); ensemble models; zero-day attacks detection; feature selection; linear scaling normalization

I. INTRODUCTION

The digitalization of daily tasks has led to the widespread use of connected computers and applications [1]. Rapid IoT growth spans various sectors, comprising healthcare, agriculture, and industry, but along with its benefits, it results in multiple vulnerabilities [2]. Zero-day threats are a major problem for cybersecurity specialists [3], as such threats exist in nearly all hardware and software systems, and several events are patched after prominence [4]. Zero-day threats are implemented in standard software, such as office applications, operating systems, or browsers [5]. A zero-day cyber threat takes advantage of a vulnerability that has not been publicly disclosed [6]. Several research efforts have been made to identify known susceptibilities using ML and DL models [7]. IDS systems are used to detect zero-day threats, which are rapidly increasing, including NIDS, monitoring network events, and HIDS, analyzing host-related data [8]. Anomaly-based IDSs (AIDSs) flag deviations from normal behavior, while Deep Learning (DL) models use layered abstractions and backpropagation to uncover complex patterns [9].

This study presents a Gorilla Troops Optimizer-based Ensemble Deep Learning Model for Zero-Day Attack Detection (GTOEDLM-ZDAD) technique. The aim is to classify and detect zero-day attacks using ensemble and advanced optimization algorithms. Initially, Linear Scaling Normalization (LSN) is used to transform raw data into a normalized format. Then, the Chimpanzee Optimization Algorithm (ChoA) is utilized for feature selection. Afterward, an ensemble DL model uses a Deep Q-Network (DQN), a Bidirectional Gated Recurrent Unit (BiGRU), and a Deep Belief Network (DBN) for classification. Finally, Gorilla Troops Optimizer (GTO)-based hyperparameter selection is performed. The proposed model is tested on the ToN-IoT dataset.

- This approach utilizes LSN to transform raw input data into a consistent, normalized format, facilitating learning and improving the stability and convergence of the models. As a result, the overall performance of the model is significantly improved.
- The ChoA efficiently chooses the most relevant features, mitigating computational costs while maintaining high performance. An ensemble of DQN, BiGRU, and DBN utilizes their merits to accurately classify complex patterns, achieving high performance and robustness.
- The proposed approach integrates GTO to efficiently fine-tune model hyperparameters, ensuring optimal settings. This optimization improves learning ability and generalization, along with prediction accuracy, and reduces training time.
- The proposed hybrid framework combines bio-inspired optimization with an ensemble of DL models, utilizing advanced feature selection and tuning for superior performance. This integration improves accuracy and efficiency, overcoming existing limitations and providing a robust solution.

In [10], a holistic model was defined and developed. In [11], an Ensemble Learning (EL) approach was proposed, using a testbed for data collection and model training. The study in [12] aimed to recognize zero-day phishing threats more effectively. In [13], a Bi-LSTM was modified with game theory concepts, residual network (ResNet50), probabilistic graph-based backpropagation neural networks, CNN with LSTM, and an ANN AutoEncoder (AE). In [14], an unsupervised online attack recognition method was presented. In [15], an Evolvable IDS (EIDS) was proposed, using an open-set technique that depends on a distinctive AE. In [16], a structure collected from a 1D-CNN was utilized. In [17], a hybrid ML model integrated K-Means, Synthetic Minority Over-Sampling Technique (KMeans-SMOTE), and Principal Component Analysis (PCA) to improve detection. In [18], bio-inspired metaheuristic algorithms were used to reduce features to train ML models. In [19], Federated Learning (FL) was employed with an isolation forest optimized using Sailfish Optimization (SFO). In [20], a rule-based Deep Neural Network (DNN) was proposed to accurately detect novel network attacks in IoT settings.

Despite diverse approaches, most of them depend on static datasets, restricting adaptability to growing threat patterns. Various models lack robustness in real-time detection and scalability across varied IoT environments. The research gap is in dynamic zero-day threat adaptation, lightweight yet high-accuracy architectures, real-time response efficiency, and integrated learning models optimized for constrained IoT resources and scalability.

II. PROPOSED METHOD

This study presents the GTOEDLM-ZDAD model to classify and detect zero-day attacks. This method comprises data normalization, feature subset selection, ensemble attack classification, and parameter selection, as shown in Figure 1.

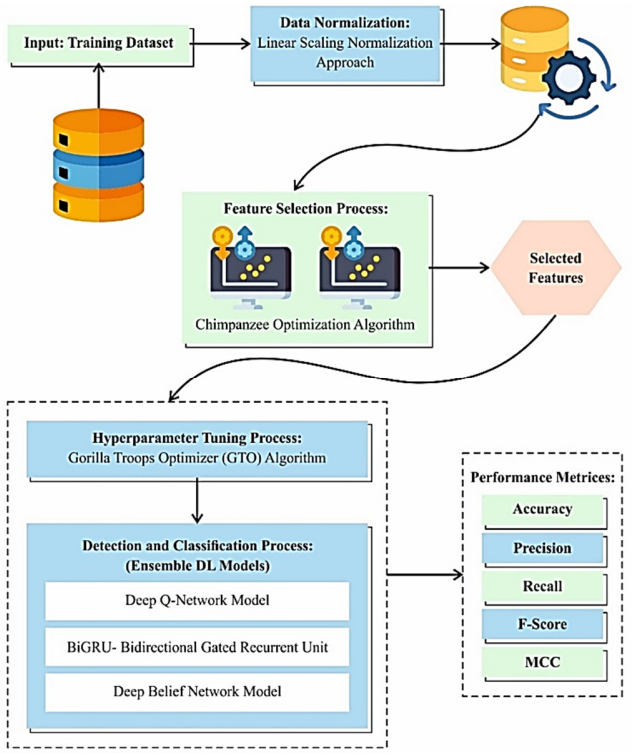

Fig. 1. Overall process of the proposed GTOEDLM-ZDAD technique.

A. Stage I: LSN

LSN is used to transform raw data into a normalized format. LSN is a data pre-processing model that alters features linearly to fall within a definite range, typically [0, 1]. This system ensures that every feature has an even measure, making them similarly significant in the recognition procedure. For zero-day attack detection, LSN is vital in processing varied system behaviors and network traffic metrics, allowing anomalies to stand out more efficiently. It protects the relationships among data points, ensuring that crucial patterns are not distorted. Furthermore, it improves the outcome of ML techniques by enhancing convergence and decreasing the impact of outliers.

B. Stage II: Feature Subset Selection

ChOA is used for the feature selection process [21], is based on the social behavior of chimpanzee populations. This method classifies chimpanzees as attackers, encirclers, chasers, and drivers, with the search process divided into two main stages: exploration and exploitation. In the exploration stage, every chimpanzee alters its location per task (encircle, attacker, driver, chaser). The upgrade of the present individual location is given by:

$$X_{chimp}(t+1) = X_{prey}(t) - a \cdot d \quad (1)$$

$$d = \left| c \cdot X_{prey}(t) - m \cdot X_{chimp}(t) \right| \quad (2)$$

where X_{chimp} and X_{prey} denote the present location vector of the chimpanzees and preys, respectively. Vector a controls the distance between preys and chimpanzees, c represents search

difficulty, d is the distance, and m is a chaos vector. The formulas for a and c are given below.

$$a = 2 \cdot f \cdot r_1 - f \quad (3)$$

$$c = 2 \cdot r_2 \quad (4)$$

where f denotes a factor of convergence, which non-linearly reduces from 2.5 to 0, and r_1 and r_2 are dual randomly generated numbers in [0,1].

$$f = 2.5\left(1 - \frac{t}{T}\right) \quad (5)$$

During exploitation, the chimpanzee model gradually refines the solution by updating the positions of the encircler, attacker, chaser, and driver roles. The location upgrade is given by:

$$X_{(t+1)} = \frac{X_1 + X_2 + X_3 + X_4}{4} \quad (6)$$

$X_{(t+1)}$ refers to an upgraded location vector of the present chimpanzee individual, and X_1, X_2, X_3, and X_4 denote the upgraded position vectors of the attacker, encircle, driver, and chaser, respectively, mathematically formulated as:

$$\begin{cases} X_1 = X_{Attacker} - a_1 \cdot d_{Attacker} \\ X_2 = X_{Barrier} - a_2 \cdot d_{Barrier} \\ X_3 = X_{Chaser} - a_3 \cdot d_{Chaser} \\ X_4 = X_{Driver} - a_4 \cdot d_{Driver} \end{cases} \quad (7)$$

$$\begin{cases} d_{Attacker} = |c_1 \cdot X_{Attacker} - m_1 \cdot X| \\ d_{Barrier} = |c_2 \cdot X_{Attacker} - m_2 \cdot X| \\ d_{Chaser} = |c_3 \cdot X_{Chaser} - m_3 \cdot X| \\ d_{Driver} = |c_4 \cdot X_{Driver} - m_4 \cdot X| \end{cases} \quad (8)$$

Here, d signifies the distance between the prey and the chimpanzee individual, c denotes an impact factor that blocks the chimpanzee throughout searching. The Fitness Function (FF) in ChoA balances minimizing selected features and maximizing classification accuracy:

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \quad (9)$$

Here, $\gamma_R(D)$ signifies an assumed classifier's classification error rate, $|R|$ indicates the cardinality of the nominated subset, $|C|$ refers to the overall features, and α and β are dual parameters corresponding to the importance of classifier quality and subset length, belonging to [1,0] and $\beta = 1 - \alpha$.

C. Stage III: Ensemble of Attack Classification

The classification model utilizes DL models, such as DQN, BiGRU, and DBN, in an ensemble.

1) DQN Technique

The Q-learning model serves as the foundation for the DQN framework [22]. Although Q-learning is a conventional RL method, DQN enhances it by integrating DL, specifically a CNN, to approximate the value function. This model uses experience replay and a separate target network to enable stable, efficient learning in high-dimensional state spaces.

Suppose α denoting the learning rate, γ signifying the factor of discount, r_t denoting the instant reward gained by the agent, and s_t and a_t characterizing the novel state and the action captured in that condition. During training, this method updates parameters θ through gradient descent to minimize the loss between target and current value network outputs. The loss function is usually stated as:

$$L(\theta) = \mathbb{E}[(y_t - Q(s_t, a_t; \theta))^2] \quad (10)$$

$L(\theta)$ denotes the loss function, being the MSE between the output of the present and the target value networks. y_t refers to the targeted value, provided by $y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-)$, r_t characterizes the instant reward from the atmosphere at time t, γ denotes the factor of discount, s_{t+1} signifies the following environmental condition at time $t+1$, and a' signifies the promising activities occupied in the following state s_{t+1}, $Q(s_{t+1}, a', \theta^-)$.

2) BiGRU Method

The GRU design is simpler than LSTM, as it retains only dual gating elements, quicker training, and smaller parameters simultaneously. The internal framework of GRU and its major equations are expressed as:

$$z_t = \sigma(W_Z \cdot [h_{t-1}, x_t] + b_Z \quad (11)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r \quad (12)$$

$$\tilde{h}_t = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \quad (13)$$

$$h_\tau = (1 - z_t) \odot h_{t-1} + z_r \odot \tilde{h}_t \quad (14)$$

where W_Z, W_r and W_h characterize the consistent weights, b_Z, b_r and b_h symbolize consistent biased terms, r_t, z_l, h_t and $\tilde{h}_t$ represent reset gates, update gates, activation states, and candidate activation states, correspondingly, σ signifies the sigmoid activation function, $[\cdot]$ denotes dual vectors linked, x_t refers to input to the GRU at instant t, $\tanh(\cdot)$ is the hyperbolic tangent activation function, and $\odot$ signifies matrix element multiplication. GRU processes data one-way, while Bi-GRU captures context from both directions for richer feature extraction. The state of the two Hidden Layers (HLs), backward and forward, is computed as:

$$\vec{h}_t = GRU(X_t, \vec{h}_{t-1}) \quad (15)$$

$$\overleftarrow{h}_t = GRU(\mathrm{X_t}, \overleftarrow{h}_{t-1}) \quad (16)$$

$$h_t = \omega_t \vec{h}_t + \nu_t \overleftarrow{h}_t + b_t \quad (17)$$

where $\vec{h}_t$ and $\overleftarrow{h}_t$ represent the outputs of the backwards and forward HLs at time t, ω_t represents weights consistent with the forward direction and ν_t signifies backward HLs, respectively, and b_t denotes the offset.

3) DBN Classifier

DBN uses the learning units derived from generative NNs, each containing a Restricted Boltzmann Machine (RBM) [23]. The RBM uses an input layer to feed data and a hidden layer to extract features with independent hidden units. The energy's joined expression is evaluated as:

$$EN(r, h, \varphi) = -\sum_{n=1}^{j} \frac{(r_n - d_n)^2}{2\vartheta^2} - \sum_{m=1}^{k} e_n \, h_m - \sum_{n=1m}^{j} \sum_{=1}^{k} \varpi_{nm} \frac{r_n}{\vartheta_m} h_m \quad (18)$$

The weightings between the hidden and visible components are specified by ϖ_{nm}, φ is mathematically stated as $\varphi = (e_m, d_n, \varpi_{nm})$, and the biased term of visible and hidden components is determined by d_n and e_m. The following equation represents the combined distribution.

$$Prb(r,h,\varphi) = \frac{d}{NR(\varphi)} \exp\left(-EN(r,h,\varphi)\right) \quad (18)$$

where φ specifies continuous standardization. The system allocates the probability-based value for all input vectors derived from $N\ (r,h,\varphi)$. The condition likelihood distribution is given by:

$$Prb(h_n|r,\varphi) = l\left(\sum_{n=1}^{j} \varpi_{nm}\, r_n + e_m\right) \quad (20)$$

$$Prb(r_n|h,\varphi) = F(\sum_{m=1}^{k} \varpi_{nm}\, h_n \vartheta_n^2 + b_m) \quad (21)$$

where ϑ denotes the standard deviation of the Gaussian variable, and F defines the Gaussian distribution. The contrasting difference is applied for the weights learning, upgraded according to:

$$\Delta\omega_{nm} = x(r_n h_{n\ data} - r_n h_{n\ \mathrm{Reconstruction}}) \quad (22)$$

where x denotes the learning rate, and r_n and h_n represent the input and hidden units. The RBM is projected according to the new restoration of the information. The learned parameters are presumed effective after the method recovers the novel information.

D. Stage IV: Parameter Selection

The GTO-based hyperparameter selection process is used to improve the detection results of ensemble methods [24]. GTO is a recent swarm model inspired by the social behavior of silverback gorillas, which form hierarchies and occasionally move independently in search of safety, mates, or food. This phenomenon is called exploration and is described as:

$$X(t+1) = (ub - lb) \times r_1 + lb \quad (23)$$

where ub and lb represent the lower and upper limits of the searching region, r_1, r_2, r_3, and r_4 depict an arbitrary number in $(0,1)$, $X(\mathrm{t+l})$ characterizes the gorilla's novel location, and $X_r(t)$ refers to the location of an arbitrarily chosen gorilla.

A stochastic number in $(0,1)$ is then selected. When the stochastic amount is lower than p, equation (23) is applied. If the stochastic value is 0.5 or higher, the gorillas are considered to be in the migration zone.

$$X(t+1) = (r_2 - C) \times X_r(t) + L \times H \quad (24)$$

The parameters, L and H, are calculated as:

$$C = F \times \left(1 - \frac{i}{\max_i}\right) \quad (25)$$

$$F = \cos(2 \times r_3) + 1 \quad (26)$$

$$L = C \times A \quad (27)$$

$$H = Z \times X(t) \quad (28)$$

$$Z = [-C, C] \quad (29)$$

where $X(t)$ denotes the present location of the gorilla, A refers to an arbitrary number in [-1, 1], $\max_i$ signifies the maximum iteration count, and i characterizes the present iteration.

$$X(t+1) = X(t) - L \times \left(L \times \left(X_t - X_r(t)\right) + r_4 \times \left(X(t) - X_r(t)\right)\right) \quad (30)$$

The following stage is the application of exploitation. To attain this, dual situations are measured. The male gorillas follow the leader if C is superior to or equivalent to W.

$$X(t+1) = X(t) - L \times M \times \left(X(t) - X_b(t)\right) + X(t) \quad (31)$$

where $X_b(t)$ signifies the optimal location of the gorilla. M is calculated as:

$$M = \left(\left|\frac{1}{N}\sum_{k=1}^{n} X_i\,(t)\right|^g\right)^{\frac{1}{g}} \quad (32)$$

$$g = 2^L \quad (33)$$

N characterizes the collective number of gorillas in groups. When C is lower than W, the male gorillas contest the lead for the power statement.

$$X(t+1) = X_b(t) - (X_b(t) \times Q - X(t) \times Q) \times B \quad (34)$$

The values of B and Q are computed as:

$$B = \beta \times E \quad (35)$$

$$Q = 2 \times r_5 - 1 \quad (36)$$

where r_5 denotes randomly generated numbers among $(0,\ 1)$, β signifies an expressed constant, and E is established as:

$$E = \begin{cases} K_1, if\ r_6 \geq 0.5 \\ K_2, if\ r_6 < 0.5 \end{cases} \quad (37)$$

Here K_1 denotes a stochastic number inside the search area, K_2 is a stochastic number in (0, 1), and r_6 is a stochastic number in (0-1). The GTO technique uses the following FF to improve classification, assigning higher values to better-performing candidate solutions.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{no.of\ misclassified\ samples}{Total\ no.of\ samples} \times 100 \quad (38)$$

III. PERFORMANCE VALIDATION

The experimental evaluation of the proposed GTOEDLM-ZDAD technique was performed using the ToN-IoT dataset [25]. This dataset covers several classes, including DDoS, DoS, Mirai, Benign, Spoofing, Recon, Web-based, and BruteForce. Only 38 features were selected from the 75 overall. Table I presents a comparative investigation of GTOEDLM-ZDAD over existing models, such as those in [26]. The Naïve Bayes (NB), CNN, DNN, KNN, SVM, and Decision Tree (DT) models achieved the lowest performance, the Random Forest (RF) model reached somewhat closer performance, and the proposed GTOEDLM-ZDAD method reported maximum performance with superior $prec_n$, $reca_l$, $accu_y$, and $F1_{score}$ of 93.32%, 93.30%, 98.33%, and 93.30%, respectively.

TABLE I. COMPARISON OF GTOEDLM-ZDAD WITH EXISTING METHODS ON THE TON-IOT DATASET

Method	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$
NB	61.31	60.84	68.48	75.54
CNN model	83.15	78.86	69.49	77.50
DNN algorithm	90.91	69.89	71.92	68.89
KNN classifier	92.95	73.97	70.91	71.20
SVM model	78.32	70.05	70.84	76.79
RF	94.02	71.26	77.86	74.66
DT	82.97	76.49	68.86	75.71
GTOEDLM-ZDAD	98.33	93.32	93.30	93.30

IV. CONCLUSION

This study presented the GTOEDLM-ZDAD model, aiming to detect and classify zero-day attacks using ensemble and advanced optimization models. Initially, LSN normalization is used to transform raw data into a normalized format, and then, the ChoA model is used for feature subset selection. An ensemble of DQN, BiGRU, and DBN models is used for classification. Finally, GTO-based hyperparameter tuning is performed. A wide range of experimentation of the GTOEDLM-ZDAD on the ToN-IoT dataset showed that it achieved a superior accuracy of 98.33% compared to existing baseline approaches. The limitations of the GTOEDLM-ZDAD technique comprise the utilization of a fixed feature set and evaluation on predefined datasets, which may restrict adaptability to growing cyber threats. Future work may focus on incorporating adaptive feature selection and testing in real-time dynamic environments for broader applicability.

REFERENCES

[1] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "Zero-day attack detection: a systematic literature review," *Artificial Intelligence Review*, vol. 56, no. 10, pp. 10733–10811, Oct. 2023, https://doi.org/10.1007/s10462-023-10437-z.

[2] M. Sarhan, S. Layeghy, M. Gallagher, and M. Portmann, "From zero-shot machine learning to zero-day attack detection," *International Journal of Information Security*, vol. 22, no. 4, pp. 947–959, Aug. 2023, https://doi.org/10.1007/s10207-023-00676-0.

[3] A. Blaise, M. Bouet, V. Conan, and S. Secci, "Detection of zero-day attacks: An unsupervised port-based approach," *Computer Networks*, vol. 180, Oct. 2020, Art. no. 107391, https://doi.org/10.1016/j.comnet.2020.107391.

[4] Y. Guo, "A review of Machine Learning-based zero-day attack detection: Challenges and future directions," *Computer Communications*, vol. 198, pp. 175–185, Jan. 2023, https://doi.org/10.1016/j.comcom.2022.11.001.

[5] T. Zoppi, A. Ceccarelli, and A. Bondavalli, "Unsupervised Algorithms to Detect Zero-Day Attacks: Strategy and Application," *IEEE Access*, vol. 9, pp. 90603–90615, 2021, https://doi.org/10.1109/ACCESS.2021.3090957.

[6] S. Ali, S. U. Rehman, A. Imran, G. Adeem, Z. Iqbal, and K. I. Kim, "Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection," *Electronics*, vol. 11, no. 23, Jan. 2022, Art. no. 3934, https://doi.org/10.3390/electronics11233934.

[7] H. Hindy, R. Atkinson, C. Tachtatzis, J. N. Colin, E. Bayne, and X. Bellekens, "Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection," *Electronics*, vol. 9, no. 10, Oct. 2020, Art. no. 1684, https://doi.org/10.3390/electronics9101684.

[8] A. T. Azar, S. U. Amin, M. A. Majeed, A. Al-Khayyat, and I. Kasim, "Cloud-Cyber Physical Systems: Enhanced Metaheuristics with Hierarchical Deep Learning-based Cyberattack Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17572–17583, Dec. 2024, https://doi.org/10.48084/etasr.8286.

[9] T. Nishitha and A. Khare, "Smart Contract-Enhanced Residual GRU with Merkle-Damgard Cryptography for IoT Attack Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19331–19336, Feb. 2025, https://doi.org/10.48084/etasr.8860.

[10] N. Peppes, T. Alexakis, E. Adamopoulou, and K. Demestichas, "The Effectiveness of Zero-Day Attacks Data Samples Generated via GANs on Deep Learning Classifiers," *Sensors*, vol. 23, no. 2, Jan. 2023, Art. no. 900, https://doi.org/10.3390/s23020900.

[11] S. Guo *et al.*, "A Zero-day Container Attack Detection based on Ensemble Machine Learning," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sinaia, Romania, Sept. 2023, pp. 1–8, https://doi.org/10.1109/ETFA54631.2023.10275683.

[12] Y. R. Purnamadewi and A. Zahra, "Enhancing detection of zero-day phishing email attacks in the Indonesian language using deep learning algorithms," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 1, pp. 505–512, Feb. 2025, https://doi.org/10.11591/eei.v14i1.8759.

[13] S. Akshaya and P. Ganapathi, "Augmenting Cyber Defense Counter To Zero-Day Attacks Through Predictive Analysis-A Fusion Methodology Assimilating Game Theory and RESNet Inspired Optimization Techniques," *International Journal of Communication Networks and Information Security*, vol. 16, no. 3, pp. 91–104, 2024.

[14] A. De Paola, S. Drago, P. Ferraro, and G. Lo Re, "Detecting Zero-Day Attacks under Concept Drift: An Online Unsupervised Threat Detection System," in *CEUR Workshop Proceedings*, 2024.

[15] D. Jin, S. Chen, H. He, X. Jiang, S. Cheng, and J. Yang, "Federated Incremental Learning based Evolvable Intrusion Detection System for Zero-Day Attacks," *IEEE Network*, vol. 37, no. 1, pp. 125–132, Jan. 2023, https://doi.org/10.1109/MNET.018.2200349.

[16] M. A. Talukder, M. Khalid, and N. Sultana, "A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction," *Scientific Reports*, vol. 15, no. 1, Feb. 2025, Art. no. 4617, https://doi.org/10.1038/s41598-025-87028-1.

[17] M. P. Singh, V. P. Singh, and M. Gupta, "Early Detection and Classification of Zero-Day Attacks in Network Traffic Using Convolutional Neural Network," in *The Future of Artificial Intelligence and Robotics*, 2024, pp. 812–822, https://doi.org/10.1007/978-3-031-60935-0_70.

[18] O. Almomani, "A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 409–429, Feb. 2021, https://doi.org/10.32604/cmc.2021.016113.

[19] A. Babu and A. Bagubali, "Federated Learning With Sailfish-Optimized Ensemble Models for Anomaly Detection in IoT Edge Computing Environment," *IEEE Access*, vol. 13, pp. 53171–53187, 2025, https://doi.org/10.1109/ACCESS.2025.3554301.

[20] S. Chakraborty, S. K. Pandey, S. Maity, and L. Dey, "Detection and Classification of Novel Attacks and Anomaly in IoT Network using Rule based Deep Learning Model," *SN Computer Science*, vol. 5, no. 8, Nov. 2024, Art. no. 1056, https://doi.org/10.1007/s42979-024-03429-5.

[21] X. He and C. Guo, "Research on Multi-Strategy Fusion of the Chimpanzee Optimization Algorithm and Its Application in Path Planning," *Applied Sciences*, vol. 15, no. 2, Jan. 2025, Art. no. 608, https://doi.org/10.3390/app15020608.

[22] Y. Liu, T. Yang, L. Tian, and J. Pei, "SGD-TripleQNet: An Integrated Deep Reinforcement Learning Model for Vehicle Lane-Change Decision," *Mathematics*, vol. 13, no. 2, Jan. 2025, Art. no. 235, https://doi.org/10.3390/math13020235.

[23] S. C. M. Sundararajan *et al.*, "IoT-based prediction model for aquaponic fish pond water quality using multiscale feature fusion with convolutional autoencoder and GRU networks," *Scientific Reports*, vol. 15, no. 1, Jan. 2025, Art. no. 1925, https://doi.org/10.1038/s41598-024-84943-7.

[24] K. Reddy, R. Sarma, and D. Guha, "Performance Analysis of Advanced Metaheuristics for Optimal Design of Multi-Objective Model Predictive Control of Doubly Fed Induction Generator," *Processes*, vol. 13, no. 1, Jan. 2025, Art. no. 221, https://doi.org/10.3390/pr13010221.

[25] "CIC-ToN-IoT." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/dhoogla/cictoniot.

[26] B. I. Hairab, H. K. Aslan, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly Detection of Zero-Day Attacks Based on CNN and Regularization Techniques," *Electronics*, vol. 12, no. 3, Jan. 2023, Art. no. 573, https://doi.org/10.3390/electronics12030573.