International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I4P104

Volume 74 Issue 4, 37-51, April 2026

Original Article

A Hybrid of Deep Learning-Based Zero-Day Attack Detection and Classification Method using a Two-Tier Metaheuristic Optimization Algorithm

J. Vanitha[1*], P. Anandababu[2]

[1,2]*Department of Computer and Information Science, Faculty of Science, Annamalai University.*

[1*]*Corresponding Author : vanithahenri@gmail.com*

Received: 11 June 2025 Revised: 07 February 2026 Accepted: 12 February 2026 Published: 29 April 2026

Abstract *- A zero-day attack is an important cyberattack for the cybersecurity community and the public. It utilizes exposures that have not been revealed openly or new attacking strategies to evade being perceived by present recognition tools. Practitioners, researchers, and businesses have struggled to invent devices to detect cybersecurity attacks for the past few years. Notably, those efforts initiated rule-based, signature- or supervised-based Machine Learning (ML) techniques that are verified efficient for perceiving previously faced and considered interruptions. This manuscript proposes a Hybrid of Deep Learning-Based Zero-Day Attack Detection Utilizing a Two-Tier Metaheuristic Optimization Algorithm (HDLZAD-TTMOA) model. The HDLZAD-TTMOA model intends to project an advanced zero-day attack classification mechanism using optimized techniques. In the initial state, the min-max standardization is used for transforming input data into a compatible structure. Meanwhile, the whale optimization algorithm is utilized for attribute subset selection. Moreover, a hybridization of the Convolution Neural Network, Temporal Convolutional Network, and Long Short-Term Memory (CNN-TCN-LSTM) method was implemented for recognition. Finally, the Enhanced Crayfish Optimization Algorithm (ECOA)-based tuning has been utilized for boosting the recognition results of the CNN-TCN-LSTM algorithm. The efficiency of the HDLZAD-TTMOA system can be inspected against the ToN-IoT and CIC-IDS-2017 database. The comparative analysis of the HDLZAD-TTMOA methodology illustrated that greater detection efficiency is related to recent methodologies.*

Keywords *- Enhanced Crayfish Optimization Algorithm, Feature Selection, Min-Max Normalization, Zero-Day Attack Detection and Classification.*

1. Introduction

An ever-rising percentage of cyberattacks is specifically targeted at a particular organization to steal data, to implement industrial sabotage, espionage, or denial of service. The most dangerous cyber threats include complex and zero-day threats [1]. Even though they are intended to be better understood by the community, they remain complicated to identify and track in the vast haystack of alerts and system logs. Artificial Intelligence (AI) devices are vital to detect unknown or complex threats [2]. Unknown threats are called zero-day threats. They leverage the formerly unknown flow of the system. Complicated threats are termed multi-level threats [3]. A zero-day threat could be defined as an interesting traffic pattern. Attackers acquire zero-day threats of unknown nature and utilize them with other complicated threats to safeguard themselves from being identified by the intrusion detection models [4]. Zero-day threats can come in several forms, like viruses, worms, network attacks, Trojans, and other malware. An Intrusion Detection System (IDS) is the major layer of protection against cyberattacks [5]. IDS has been around for an extended period, from the classical ones that utilize threat signatures, to the downstream methods advanced utilizing ML approaches [6]. The process of removing threat signatures is time-consuming and complicated. Furthermore, these techniques only work for previously analyzed and detected threats, but are susceptible to novel threats that have never been identified. The recent versions of IDS are signature-less [7]. These kinds of IDS utilize ML-based approaches, specifically the Deep Learning (DL) ones, for threat recognition [8].

Depending on prior history, rule-based and signature-based models have yielded effective outcomes to detect conventional cyber-threats that specify different patterns, most probably termed fingerprints or signatures. Cyber-threat exists, are persistent and advanced, and are named Advanced Persistent Threat (APT). Such threats can modify memory, communicate between modules, interact with control and command, open files, activate threads, and implement packet routing [9]. Consequently, to cope with such progressive

attacks, ML models aid in advanced, precise, and somewhat accurate predictions of circumstances and upcoming events, and how to perform intelligently in those conditions [10]. Generally, ML endeavours to learn how to yield enhanced and more valuable conditions in the future. Figure 1 depicts the infrastructure of the zero-day cyberattack identification process.

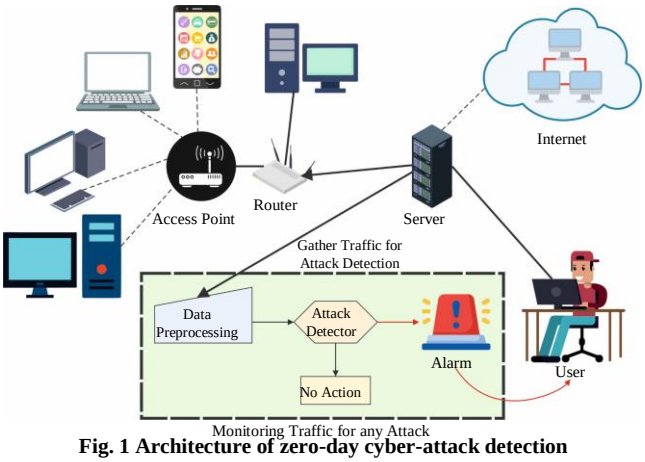

Fig. 1 Architecture of zero-day cyber-attack detection

This manuscript proposes a Hybrid of Deep Learning-Based Zero-Day Attack Detection Utilizing a Two-Tier Metaheuristic Optimization Algorithms (HDLZAD-TTMOA) model. The key contributions are:

- Data transformation using min-max normalization.
- Utilized Whale Optimization Algorithm (WOA) for FS to detect the most appropriate attack features.
- Furthermore, a hybridization of the Convolution Neural Network, Temporal Convolutional Network, and Long Short-Term Memory (CNN-TCN-LSTM) was deployed to detect data dispersion and classify attacks.
- Moreover, Enhanced Crayfish Optimization Algorithm (ECOA)-based tuning is utilized for optimal model configuration.
- The efficiency of the HDLZAD-TTMOA approach is examined against ToN-IoT and CIC-IDS-2017 datasets.

2. Related Works

Sarhan et al. [11] introduced a novel zero-shot learning technique to assess the ML-based NIDS performance to recognize zero-day threat consequences. In the inference phase, these techniques develop the relationships between zero-day and known threats to identify them as harmful. Soltani et al. [12] introduced a DL-based IDS structure flexible to novel threats, containing diverse stages. The initial stage utilizes DL-based open set detection models to recognize unknown instances that are novel threats, and create a report from dissimilar known threats concurrently. Then, the new instances are gathered by incorporating the clustering and deep models. These clusters are the primary key to making the labelling process more effective and decreasing the effort and time of the specialized knowledge team. In [13], an innovative active learning structure dependent on Deep Q-Network (DQN) is introduced. This structure is made from a NIDS classifier, an example annotator, and a selection approach. DQN techniques act as a new controlling element to choose the zero-day instances. The BiLSTM model is also incorporated into the DQN technique.

In [14], the efficiency of dual methods is examined for identifying electricity theft employing a dataset of real electricity utilization readings. Particularly, this study signifies the primary attempt to use a comparative analysis of the implementation of supervised DL and anomaly detection techniques against zero-day electricity theft cyber-threats. The authors in [15] focused on verifying the recognition method's dependability when confronted with a threat it wasn't trained on before. Thus, the CNN classifier performance is assessed to identify the harmful threat traffic, particularly the threats never stated before in the system, that is, Zero-Day threats. Shukla [16] introduced a structural pattern over an evolutionary hybrid model that integrates Simulated Annealing (SA) and TLBO, named TLBOSA for IDS. The SVM is utilized to choose the related features that could assist in precisely categorizing the threats.

In [17], a DL-based method is proposed for zero-day threat recognition. This method employs system flow telemetry enlarged with asset-level graph attributes sent to a two-AE framework for novelty and anomaly recognition. Wakili and Bakkali [18] developed ZeroDefense, an adaptive IDS by integrating multiple classifiers. Almofarreh et al. [19] presented an effective Zero Day Attack Detection using Equilibrium Optimizer (EO)-based FS and optimized Bidirectional Gated Recurrent Unit (BiGRU) classifier. Also, Subtraction Average-Based Optimizer (SAO)-based parameter tuning is applied. Dakheel, Dakheel, and Flayyih [20] introduced a Bayesian Optimized Random Forest (BORF) methodology by utilizing RF and Bayesian Optimization.

Mohamed et al. [21] developed a probabilistic composite framework for zero-day exploit recognition by combining Adaptive Wave Principal Component Analysis Autoencoder (AWPA) with an optimal DL model. Ahanger et al. [22] presented a survey of recent ML and DL methods. Abid, Alhebaishi, and Althaqafi [23] developed a Robust Zero Day Attack Detection with Optimal DL (RZDAD ODL) method by incorporating Honey Badger Algorithm (HBA)-based feature optimization with a Conditional Variational Autoencoder (CVAE) for attack identification, and Rider Optimization Algorithm (ROA) for optimal parameter tuning. Rahim and Chishti [24] examined the role of DL-based IDS by utilizing DL methods. Mirza et al. [25] introduced a Zero Day BERTa (ZDBERTa) based on a Zero Shot Learning (ZSL) model by utilizing Byte Pair Encoding, a Transformer Encoder, and

GANs. Though the current methods are efficient, few models, such as ZeroDefense and ZDAD EODL, exhibit restrictions due to limited edge utilization and computational resources. Many models show less efficiency due to poor generalization and probabilistic frameworks. Also, diverse transformer-based techniques rely on synthetic data and mitigate labeling efforts. There also exists a research gap in attaining scalability, latency, and generalization across various network scenarios.

3. Methodology

The HDLZAD-TTMOA model designs a new zero-day attack detection using DL and advanced optimization models. It comprises data normalization, FS, classification, and tuning. Figure 2 shows the HDLZAD-TTMOA infrastructure.

3.1. Input Data Scaling

Here, input data is scaled using min-max normalization to convert raw input to a standardized way that ensures comparability and consistency across features. This is a data pre-processing model that usually measures feature values to a definite range [0, 1]. It ensures that each feature contributes equally to the method, eliminating bias caused by varying scales [26]. For a zero-day attack detection model, min-max normalization aids in normalizing system behaviour metrics or network traffic data, making anomalies more distinct. Upholding the relationships among data points upholds patterns vital for effective classification. Moreover, it enhances the convergence and accuracy of ML techniques, certifying trustworthy recognition of earlier hidden threats.

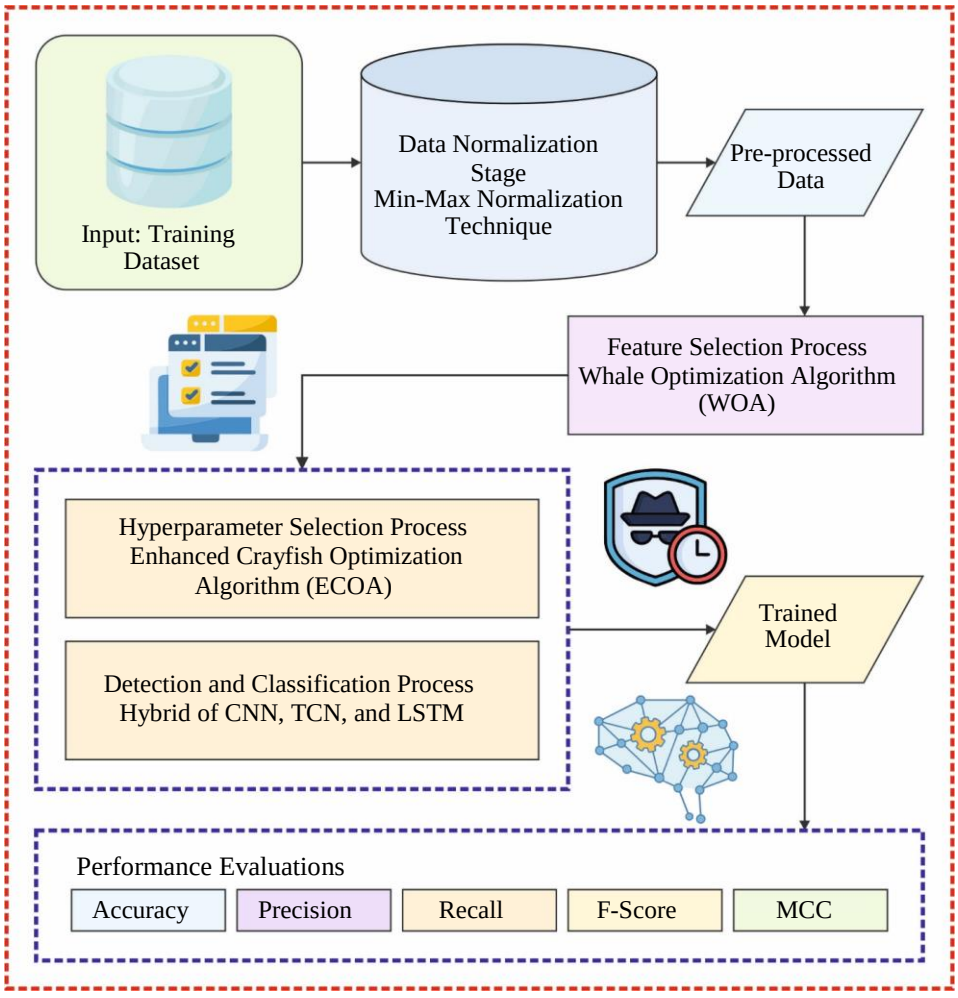

Fig. 2 Structure of HDLZAD-TTMOA model

3.2. Dimensionality Reduction: WOA

In addition, the FS process is executed by WOA to detect highly relevant and significant features in the input data. The WOA is a population-based optimizer method influenced by the humpback whales' bubble net searching approach [27]. The tactic permits whales to powerfully surround prey and attack in a spiral movement while searching the region. It incorporates dual stages: exploration (searching for prey) and exploitation (bubble-net attacking and encircling the prey) to balance global and local searches. Prey encircling assumes that the optimum candidate solutions discovered thus far represent the targeted victim. Another solution for the

candidate (whales) is to fine-tune their locations regarding their optimal location, which is mathematically expressed as demonstrated below:

$$y(z+1) = y^*(z) - A \cdot |C \cdot y^*(z) - y(z)| \quad (1)$$

Whereas $y(z)$ refers to the existing whale position and $y^*(z)$ refers to the location of the targets recognized thus far, A and C are vectors of coefficients described as:

$$A = 2\alpha \cdot \gamma - \alpha \quad (2)$$

$$C = 2 \cdot \gamma \quad (3)$$

By $\gamma \in [0,1]$ to be randomly generated vectors, and a reduction in the linear function. In addition, the bubble-net attack strategy relates to dual behaviours: spiral updating of positions and shrinking encircling. Firstly, the searching agent meets nearby optimal solutions by reducing the A values, decreasing the searching area, and concentrating on exploitation.

3.3. Spiral Updating Location

Whales are swimming near the victim in a spiral form. These behaviours are demonstrated as:

$$x(z+1) = |x^*(z) - x(z)| \cdot e^{bl} \cdot \cos(2\pi l) + x^*(z) \quad (4)$$

Meanwhile, b denotes a continual description of the logarithmical spiral form. l refers to randomly generated numbers in [1, 1], and $|x^*(z) - x(z)|$ signifies the distance between the prey and the whale. The possibility of choosing spiral updating or shrinking encircling is equivalent to a 50% possibility for all behaviours.

3.4. Exploration Level

To stop early convergence and guarantee a different searching region, WOA presents an arbitrary searching method. Whales move arbitrarily towards another candidate that is accurately stated as:

$$x(z+1) = x_{rand} - A \cdot |C \cdot x_{rand} - X(z)| \quad (5)$$

On the other hand, x_{rand} denotes the location of a whale selected at random. A and C represent similar as described previously. These behaviours are activated if $|A| > 1$, guaranteeing enough exploration. Figure 3 portrays the flowchart of WOA. The FF illustrates classifier efficacy and the chosen feature count. It aims to maximize accuracy while minimizing the feature set size. Accordingly, the FF is shown in Eq. (6).

$$Fitness = \alpha * ErrorRate + (1-\beta) * \frac{\#SF}{\#All\_F} \quad (6)$$

Here, $ErrorRate$ denotes the classification error rate. $ErrorRate$ has been computed as the ratio of incorrectly classified to the classifier counts, ranging from zero to one. $\#SF$ and $\#All\_F$ refer to the overall features. β is used to control the prominence of classifier excellence and subset length.

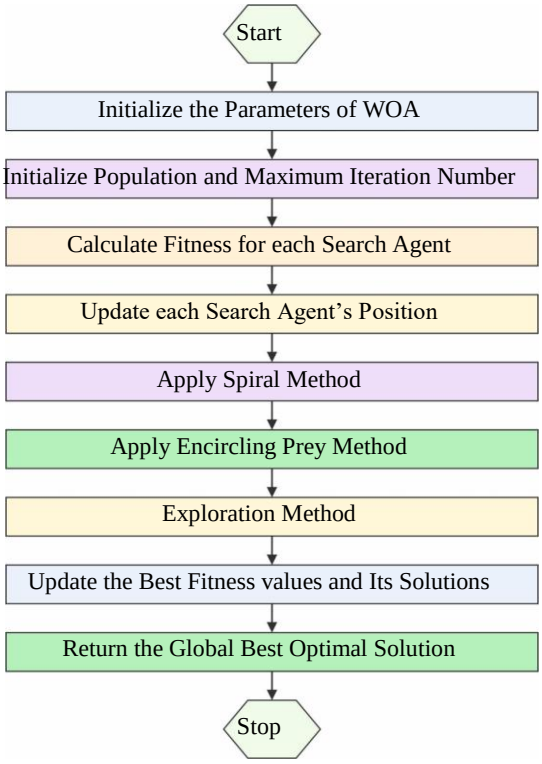

Fig. 3 Flowchart of WOA

3.5. Detection using CNN-TCN-LSTM Model

Here, the HDLZAD-TTMOA technique derives a hybrid of the CNN-TCN-LSTM technique for detection and classification. CNN uses convolutional calculations to remove features from data, making it appropriate for settings where spatial characteristics are essential [28]. The architecture of this network typically comprises pooling, convolutional, and a Fully Connected (FC) layer. The convolutional layer performs feature extraction, the pooling layer carries out feature sub-sampling, and the FC layer connects the removed features and gains classification grades. TCN has hierarchically taken either temporal or spatial information. It comprises 1D convolution, causal convolution sequence method, residual connections, and expansion convolutions. The performance of the activation function TCN for Eqs. (7) and (8) are stated. A complete architecture of TCN with d layers is demonstrated.

$$R_t^{(i,l)} = F\left(W^{(1)} R_{t-s}^{(i,l-1)} + W^{(2)} S_t^{(i,l-1)} + b\right) \quad (7)$$

$$\hat{R}_t^{(i,l)} = \left(R_t^{(i,l-1)} + V\hat{R}_t^{(i,l)} + e\right) \quad (8)$$

Whereas, W specifies the trainable weighted matrix, b characterizes the trainable bias, and l i characterizes the layer and unit number, correspondingly. $S^{(i,l)}$ Specify training function. $\hat{R}_t^{(i,l)}$ Directs expansion convolution at t. $R_t^{(i,l)}$ denotes the expansion convolution outcome after adding the residual value t.

The objective of designing shorter-term memory networks is to address the problem of longer-term dependences, which is solved by utilizing a constant memory signified as C_z.

During the initial phase, information required to be removed from the state of the cell is implemented by the sigmoid layer, named the forget gate. These gates output a value of 0 or 1 to the cell state, C_{z-1} according to the values of h_{z-1} and x_z for all components. When the value is 1, each of the cell state values C_{z-1} is distributed to C_t, and when the value is 0, it removes the data from the cell state C_{t-1}, and no values are assigned to C_z, as specified below.

$$f_z = \sigma\left(W_f \times [h_{z-1r}x_z] + b_f\right) \quad (9)$$

During this next stage, decisions are required to define which novel information must be restored in the state of the cell. This is achieved by the sigmoid layer named the input gate. Then, the hyperbolic tangent layer presents values represented as C_z, which are in addition to the state of the cell. At last, these dual stages are integrated to upgrade the cell state value, as depicted in Eqs. (10) and (11):

$$i_z = \sigma(W_i \times [h_{z-1}, x_z] + b_i) \quad (10)$$

$$\tilde{C}_z = tanh(W_c \times [h_{z-1}, x_z] + b_c) \quad (11)$$

Here, to upgrade the cell state from C_{z-1} to C_z, it is essential to multiply the preceding cell state value by f_z. Formerly, $i_z \times \tilde{C}_z$ is added, leading to the novel values for the state of the cell.

$$C_z = f_z \times C_{z-1} + i_z \times \tilde{C}_z \quad (12)$$

A sigmoid layer selects which portions of the cell layer must be directed to the outcome. Formerly, the layer of cell value proceeded through a hyperbolic tangent layer, and its value is multiplied by the output of the preceding sigmoid layer to define the related quantities sent to the output.

$$(W_i \times [h_{z-1r}x_z] + b_i) \quad (13)$$

$$\tilde{C}_z = tanh(W_c \times [h_{z-12}x_z] + b_c) \quad (14)$$

The method contains two parallel branches, consisting of TCN and LSTM techniques. A 1D convolution layer is applied for primary feature extraction on all branches. The outputs of these branches are combined and then go through a sigmoid activation function to communicate feature values between (0,1). LSTMs are well-known for their capability to arrest longer-term dependencies owing to their gating mechanisms that assist in preserving related previous information over longer sequences. This generates LSTMs that are mostly active for tasks, while patterns might recur over a long period. In contrast, TCNs that apply dilated causal convolutions are experts at taking local temporal patterns within a fixed receptive area, which can additionally increase to shield a broader range of time stages owing to the dilation. This architecture permits TCNs to handle sequences more powerfully and in parallel, possibly offering quicker convergence and a stronger temporal representation. By integrating LSTM and TCN, methods can take advantage of either local or global temporal dependencies that may decrease overfitting to particular temporal designs and increase the overall accuracy and robustness in regression tasks.

3.6. Parameter Optimizer: ECOA

Eventually, the ECOA-based hyperparameter selection procedure will be implemented to augment the recognition outcomes of the CNN-TCN-LSTM approach. The COA was intended according to the Crayfish (CF) environmental behaviour, with its essential method stimulated by the ecological temperature regulation [29]. During this model, the $tempe$ is fixed as a randomly generated parameter to mimic the vagueness of the model, as defined below.

$$tempe = random * 15 + 20 \quad (15)$$

$$p = C_1 \times \left(\frac{1}{\sqrt{2}\times\pi\times\sigma} \times \exp\left(-\frac{(temp-\mu)^2}{2\sigma^2}\right)\right) \quad (16)$$

Whereas $rand$ denotes an arbitrary number in [0, 1], μ signifies the appropriate ecological temperature, C_1 means constant (0.2), and σ indicates another constant.

If $tempe$ is very high, for example, $> 30\ ^\circ C$, CF seeks shelter in caves to escape the heat.

$$X_{shade} = \frac{X_G + X_L}{2} \quad (17)$$

X_G characterizes the best location gained over the iteration counts, and X_L characterizes the optimum location gained after upgrading the preceding generation of the population. During this shelter stage, the struggle for caves is arbitrary.

If $rand < 0.5$, it specifies that the source of the cave has not been engaged by other CFs, and the CF would arrive at the cave for shelter from the heat.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + C_2 \times random \times \left(X_{shade} - X_{i,j}^{t}\right) \quad (18)$$

Here t characterizes the present round, $t+1$ characterizes the following production round, $X_{i,j}^{t}$ denotes the location of the CF in the tth round, and C_2 denotes the reducing curve, computed as demonstrated:

$$C_2 = 2 - (t/T) \quad (19)$$

Here, T characterizes the maximal round counts.

Once the cave's resources are restricted, numerous CF fight for the chance to arrive at the cave. These behaviours relate to the struggle stage of the model, while the interaction amongst CF individuals is supposed to attain the optimum distribution of the resource. After the ecological $tempe > 30\ ^{\circ}C$ and $random \geq 0.5$, it specifies that another CF has additionally selected a similar cave.

$$X_{i,j}^{t+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{shade} \quad (20)$$

$$z = round\big(random \times (N-1)\big) + 1 \quad (21)$$

Now, N characterizes the CF's population dimensions, and z signifies an arbitrary individual inside the population of CF. If $tempe \leq 30\ ^{\circ}C$, the CF arrives at the searching stage. The food size and location can be described as demonstrated:

$$X_{food} = X_G \quad (22)$$

$$Q = C_3 \times rand \times \left(\frac{fitness_i}{fitness_{food}}\right) \quad (23)$$

Whereas C_3 denotes the food aspect, demonstrating the maximum food. $fitness_i$ embodies the ith CF's fitness values, and $fitness_{food}$ symbolizes the fitness values at the food position.

If $Q > (C3\ +\ 1)/2$, specifies that food is high, and the CF would rip food separately, utilizing the succeeding equation:

$$X_{food} = \exp\left(-\frac{1}{Q}\right) \times X_{food} \quad (24)$$

Afterwards, the food turns out to be small, and the CF would utilize its pincers to tear it apart. Then, the 2nd and 3rd claws would seize and consume the food. During this model, this alternate procedure is to pretend to utilize an amalgamation of cosine and sine functions, as exposed below:

$$X_{i,j}^{t+1} = X_{i,j}^{t} + X_{food} \times p \times (\cos(2 \times \pi \times random) - \sin(2 \times \pi \times random)) \quad (25)$$

If $Q \leq (C3\ +\ 1)/2$, it designates that the food is not very big, and the CF will shift straight near the food and eat it.

$$X_{i,j}^{t+1} = \left(X_{i,j}^{t} - X_{food}\right) \times p + p \times random \times X_{i,j}^{t} \quad (26)$$

While the COA performed well in several optimization issues, it even has restrictions after addressing higher-dimensional, composite, or multi-modal difficulties. In the searching procedure, the population might converge prematurely to the local best owing to an absence of diversity. Furthermore, the updated rules in the foraging and sheltering stages are comparatively simpler and might not thoroughly explore the searching region. In high-dimensional difficulties, the convergence rate is sub-optimum. To deal with these problems, this study presented developments to the COA, targeting to improve its accelerated convergence and global searching capacity and prevent getting stuck in local bests.

Mirror reflection learning improves population diversity. For all individuals X_i from the population, their mirror reflection location X_{MRL} has computed here:

$$X_{MRL} = (0.5 + 0.5\lambda)\ (X_{min} + X_{max}) + \lambda X_i \quad (27)$$

In the novel COA, the updated guidelines in the sheltering stage are comparatively simpler, which might not be enough to successfully lead the population towards the global best practice. With the Aquila Optimizer (AO), an effectively guided updated method is presented to increase the model's search accuracy and convergence speed. This combination gives a more effective and concentrated searching procedure, assisting the model for high convergence while improving the accuracy of the solution.

$$X_{i,j}^{t+1} = X_{food}\left(1 - \frac{t}{T}\right) + \left(\frac{\sum_{i=1}^{N} X_i}{N} - X_{food}\right).rand \quad (28)$$

Here, t signifies the present round, and T refers to the maximal iterations.

Arithmetic crossover improves the genetic population diversity by substituting information among several sizes, thus increasing the search area. This improves the global searching ability and prevents the model from getting stuck in local minima. For all individuals X_i within the population, dual sizes $index_1$ and $index_2$ are arbitrarily chosen, and linear combinations are carried out to make a novel individual location X_{MSvi}. The equation to generate X_{MSvi} is as demonstrated:

$$X_{MSvi} = rand.X_i, index_1 + (1 - rand).X_{i,index_2} \quad (29)$$

Levy Flight (LF) is a random walking approach considered by longer-distance jumps that allows operative examination of the searching region and aids the model in escaping from local bests. By presenting LF, the model's global searching capability is considerably improved.

$$X_{i,j}^{t+1} = \left(X_{i,j}^{t} - X_{food}\right) \times p + p \otimes Levy(\lambda) \times X_{i,j}^{t} \quad (30)$$

Fitness selection (FS) is crucial for enhancing ECOA efficiency. Hyperparameter tuning comprises a solution-encoding approach to compute candidate efficiency, with accuracy serving as the primary metric for the FF.

$$Fitness = \max(P) \quad (31)$$

$$P = \frac{TP}{TP+FP} \quad (32)$$

From the above equation, TP and FP indicate the true and positive rates.

4. Experimental Assessments

This study examines the HDLZAD-TTMOA model's performance assessment using the ToN-IoT dataset [30] and the CIC-IDS-2017 dataset [31]. The former one includes 75 attributes (chosen 75) and 8 classes. The latter one holds 78 attributes (chosen 49) with 7 classes.

Figure 4 represents the classification solutions of the HDLZAD-TTMOA technique on the ToN-IoT dataset. Figures 4(a)-(b) illustrate the confusion matrices under 70%TRASET and 30%TESSET. Figures 4(c)-(d) signify the PR and ROC, which resulted in a superior outcome across all class labels.

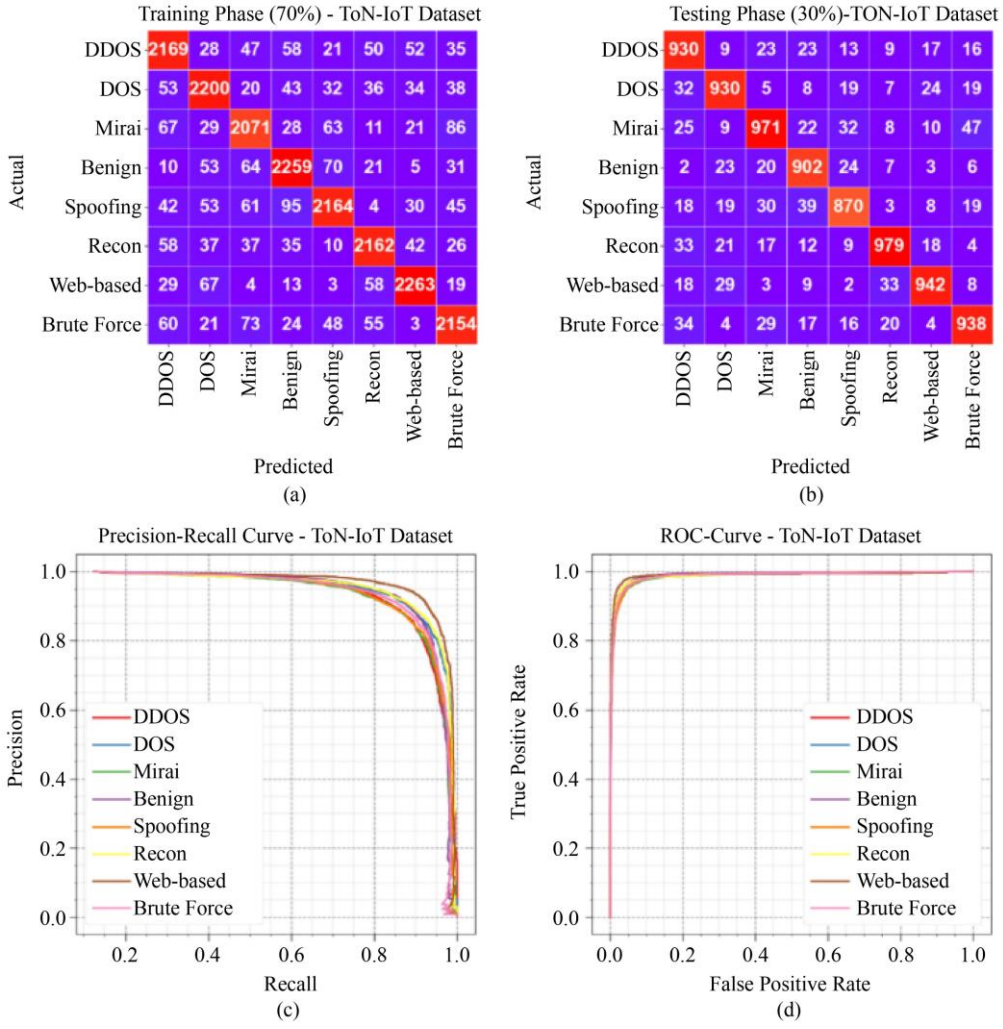

Fig. 4 ToN-IoT database (a and b) confusion matrices and (c and d) Curves of PR and ROC

In Table 1 and Figure 5, global classification solutions of the HDLZAD-TTMOA approach are described below for the ToN-IoT dataset using 70%TRASET and 30%TESSET. The presented values concluded that the HDLZAD-TTMOA model capably identifies the attack instances. With 70% TRASET, the HDLZAD-TTMOA approach attains an average $accur_y$ of 97.25%. Afterwards, using 30% TESSET, the HDLZAD-TTMOA system gets an average $accur_y$ of 97.21%.

Table 1. Results of HDLZAD-TTMOA method under ToN-IoT dataset

Class	$Accur_y$	$Preci_n$	$Recal_l$	$F1_{Score}$	MCC
TRASET (70%)					
DDoS	96.89	87.18	88.17	87.67	85.89
DoS	97.22	88.42	89.58	89.00	87.41
Mirai	96.88	87.13	87.16	87.14	85.37
Benign	97.19	88.41	89.89	89.15	87.54
Spoofing	97.06	89.76	86.77	88.24	86.57
Recon	97.55	90.20	89.82	90.01	88.61
Web-based	98.06	92.37	92.14	92.25	91.15
BruteForce	97.12	88.50	88.35	88.42	86.78
Average	**97.25**	**88.99**	**88.99**	**88.99**	**87.42**
TESSET (30%)					
DDoS	96.76	85.16	89.42	87.24	85.42
DoS	97.29	89.08	89.08	89.08	87.53
Mirai	96.67	88.43	86.39	87.40	85.49
Benign	97.44	87.40	91.39	89.35	87.93
Spoofing	97.01	88.32	86.48	87.39	85.70
Recon	97.61	91.84	89.57	90.69	89.33
Web-based	97.79	91.81	90.23	91.01	89.76
BruteForce	97.11	88.74	88.32	88.53	86.88
Average	**97.21**	**88.85**	**88.86**	**88.84**	**87.25**

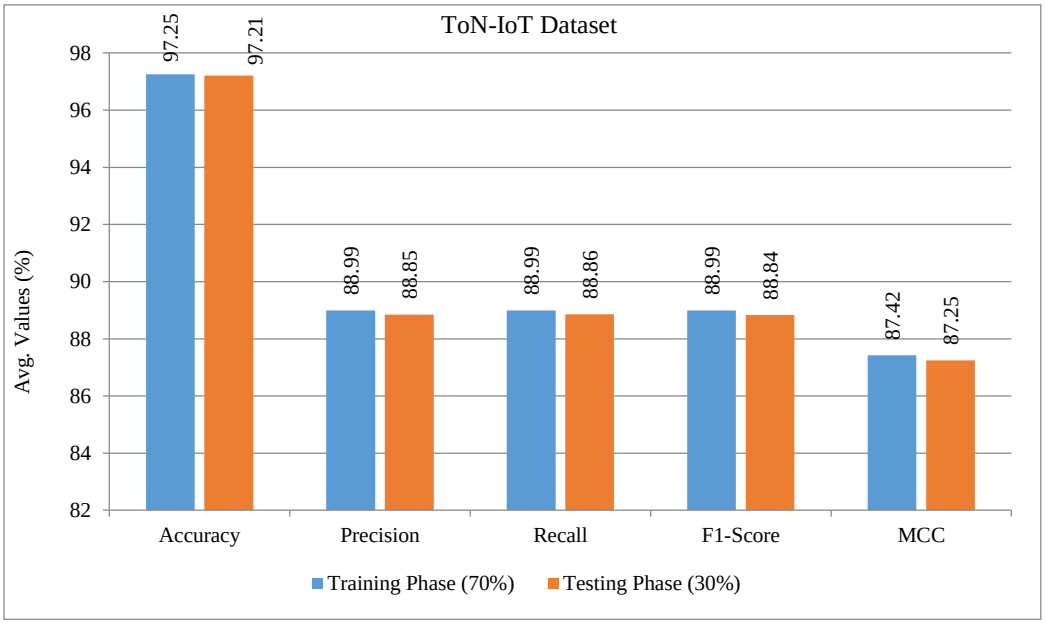

Fig. 5 Average outcomes of HDLZAD-TTMOA technique with ToN-IoT database

On the ToN-IoT database, Figure 6 presents TRAN $accur_y$ (TRANAY) and Validation $accur_y$ (VALAY) values increase steadily, illustrating robust, enhanced performance across multiple iterations. Their close alignment across epochs indicates minimal overfitting and reliable predictions on unseen data. Figure 7 depicts the TRAN loss (TRANLO) and VALA loss (VALALO) value decreases, highlighting the capability to balance generalization and data fitting on the ToN-IoT database. The gradual loss reduction ensures optimal performance and progressively refines predictions.

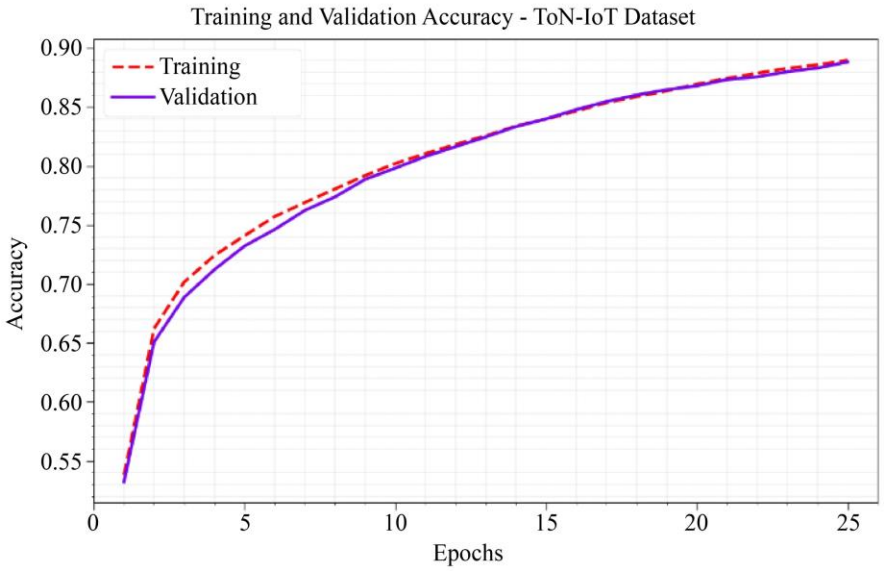

Fig. 6 $Accur_y$ curve of HDLZAD-TTMOA with ToN-IoT database

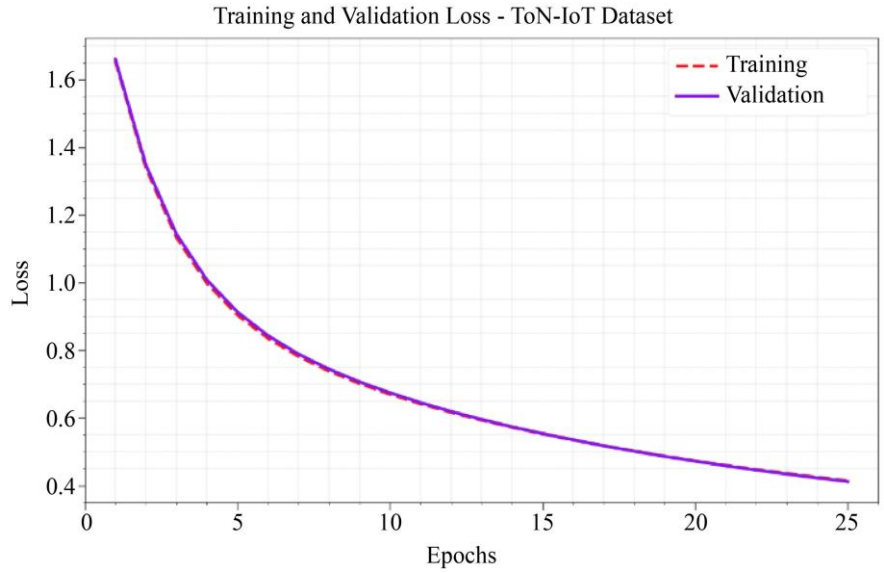

Fig. 7 Loss curve of HDLZAD-TTMOA approach under ToN-IoT dataset

Figure 8 indicates the outcomes of HDLZAD-TTMOA with the CIC-IDS-2017 database. Figuress. 8(a)–(b) depicts the confusion matrix across all seven classes using 70% TRASET and 30% TESSET. Figures 8(c)-(d) display PR and ROC, signifying robust presentation across classes.

In Table 2 and Figure 9, global classification solutions of the HDLZAD-TTMOA model are revealed below the CIC-IDS-2017 database using 70%TRASET and 30%TESSET.

Using 70% TRASET, the HDLZAD-TTMOA reaches an average $accur_y$ of 97.73%. Moreover, using 30% TESSET, the HDLZAD-TTMOA achieves an average $accur_y$ of 97.56%. Figure 10 depicts the TRANAY and VALAY solutions of the HDLZAD-TTMOA method below the CIC-IDS-2017 dataset computed over 25 epochs. The outcomes show a growing tendency in TRANAY and VALAY, highlighting the robust and steady performance.

Their close alignment across epochs indicates mitigated overfitting and reliable predictions on unseen data.

Figure 11 illustrates the TR vANLO and VALALO graphs of the HDLZAD-TTMOA technique below the CIC-IDS-2017 dataset computed over 25 epochs. The TRANLO and VALALO values lessen, reflecting the capability of the HDLZAD-TTMOA method.

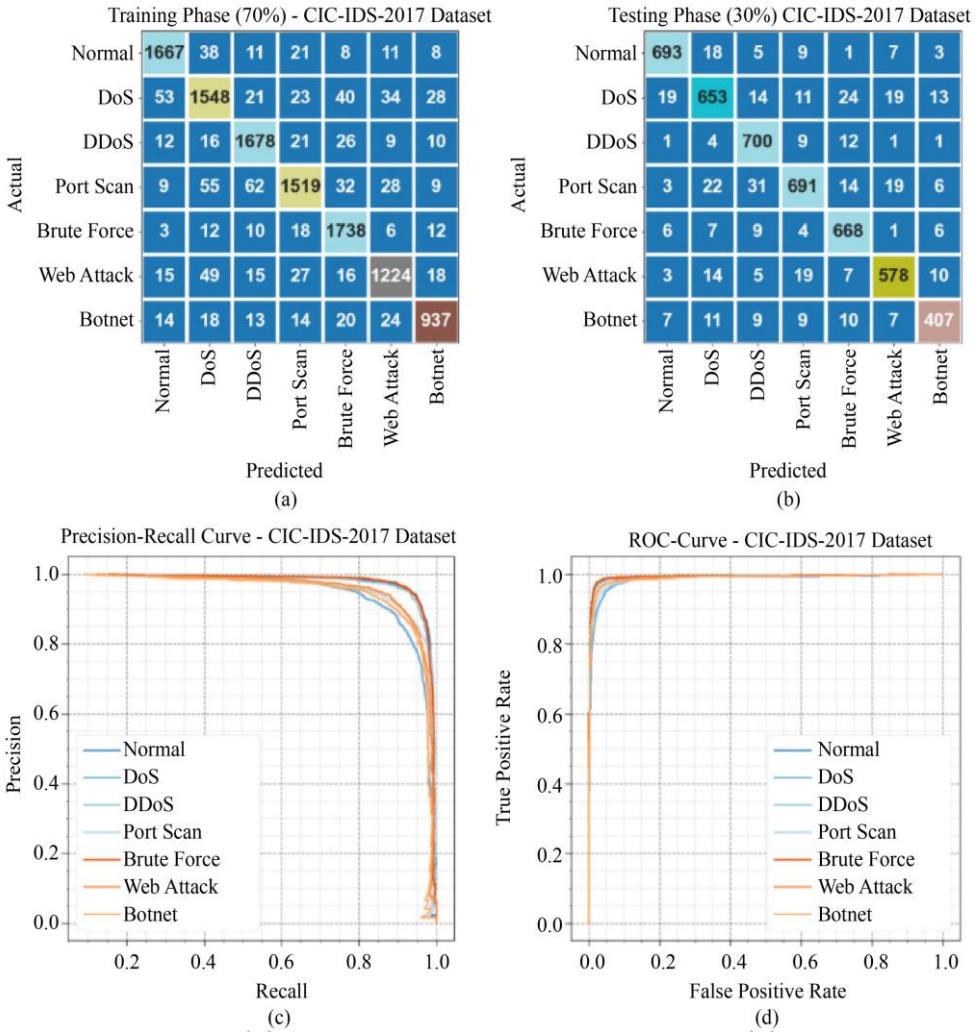

Fig. 8 CIC-IDS-2017 database (a and b) confusion matrices and (c and d) Curves of PR and ROC

Table 2. Classifier results of the HDLZAD-TTMOA approach under the CIC-IDS-2017 database

Classes	$Accur_y$	$Preci_n$	$Recal_l$	$F1_{Score}$	MCC
TRASET (70%)					
Normal	98.19	94.02	94.50	94.26	93.18
DoS	96.54	89.17	88.61	88.89	86.84
DDoS	97.98	92.71	94.70	93.69	92.50
Port Scan	97.15	92.45	88.62	90.50	88.85
Brute Force	98.19	92.45	96.61	94.48	93.43
Web Attack	97.75	91.62	89.74	90.67	89.39
Botnet	98.32	91.68	90.10	90.88	89.96
Average	**97.73**	**92.01**	**91.84**	**91.91**	**90.59**
TESSET (30%)					
Normal	98.29	94.67	94.16	94.41	93.41
DoS	96.33	89.57	86.72	88.12	85.97
DDoS	97.90	90.56	96.15	93.27	92.08
Port Scan	96.75	91.89	87.91	89.86	87.95
Brute Force	97.90	90.76	95.29	92.97	91.77
Web Attack	97.67	91.46	90.88	91.17	89.82
Botnet	98.08	91.26	88.48	89.85	88.80
Average	**97.56**	**91.45**	**91.37**	**91.38**	**89.97**

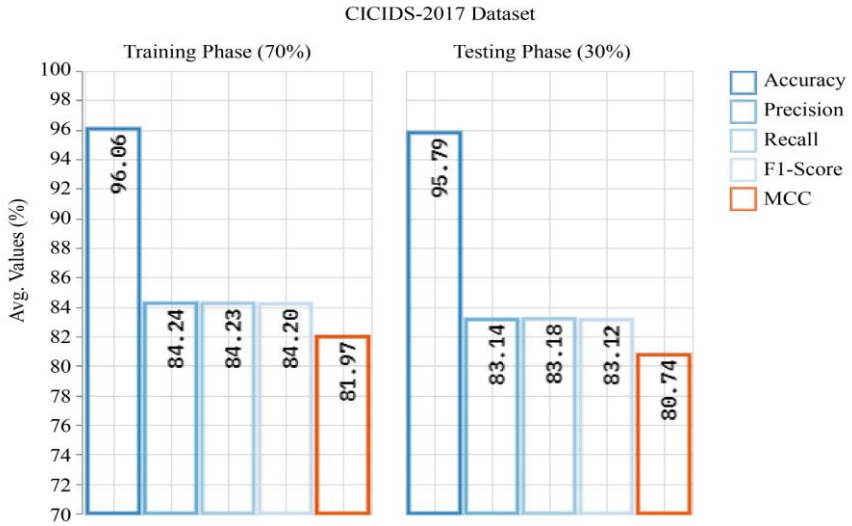

Fig. 9 Average outcome of HDLZAD-TTMOA approach under CIC-IDS-2017 dataset

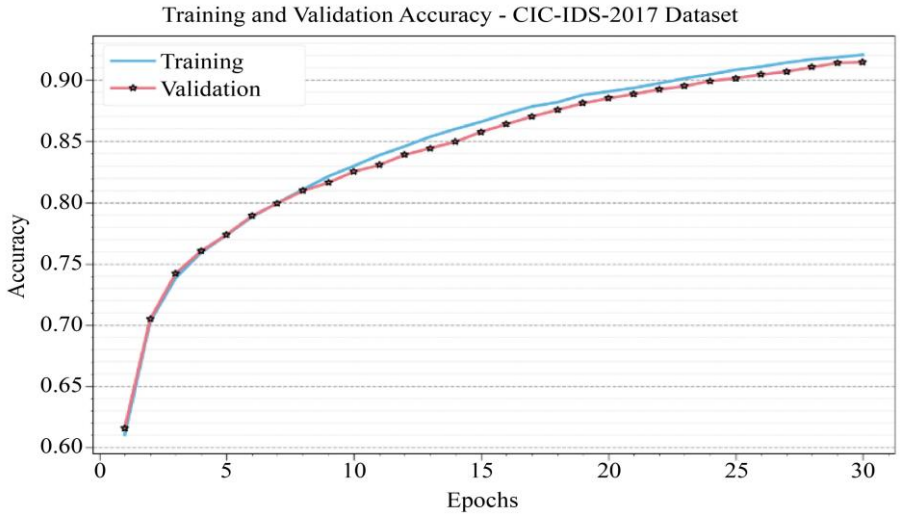

Fig. 10 $Accur_y$ curve of HDLZAD-TTMOA approach under the CIC-IDS-2017 dataset

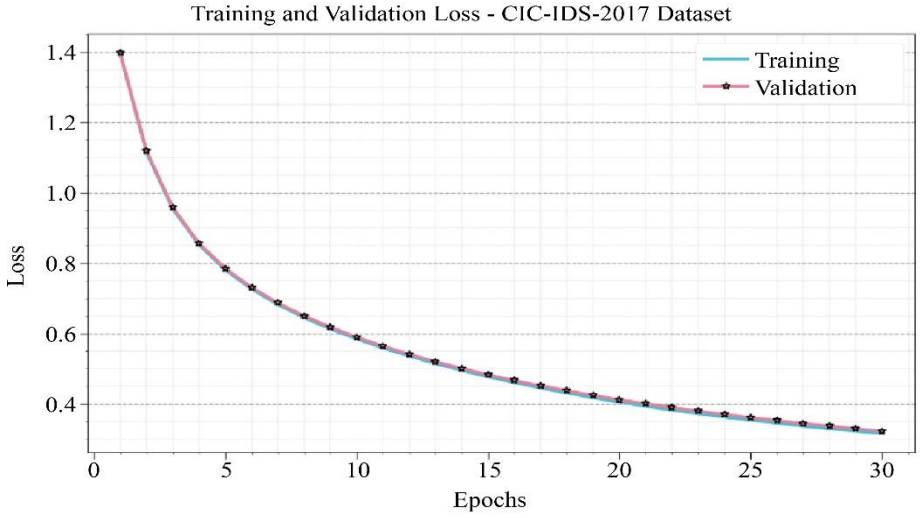

Fig. 11 Loss curve of HDLZAD-TTMOA under the CIC-IDS-2017 database

In Table 3 and Figures 12-13, the general comparison and Computational Time (CT) of the HDLZAD-TTMOA methodology are shown [6, 32-33]. According to $accu_y$, the HDLZAD-TTMOA methodology provides maximum $accu_y$ of 97.73%, whereas the Naïve Bayes (NB), CNN, DNN, KNN, SVM, RF, and Decision Tree (DT) models attain lesser $accu_y$ of 61.31%, 83.15%, 90.91%, 92.95%, 78.32%, 94.02%, and 82.97%, respectively. Similarly, according to $prec_n$, the HDLZAD-TTMOA model delivers improved $prec_n$ of 92.01%, whereas the NB, CNN, DNN, KNN, SVM, RF, and DT models attain $prec_n$ of 60.84%, 78.86%, 69.89%, 73.97%, 70.05%, 71.26%, and 76.49%, respectively. Lastly, according to $F1_{score}$, the HDLZAD-TTMOA model provides enhanced $F1_{score}$ of 91.91% whereas the NB, CNN, DNN, KNN, SVM, RF, and DT models obtain minimum $F1_{score}$ of 75.54%, 77.50%, 68.89%, 71.20%, 76.79%, 74.66%, and 75.71%, respectively. Finally, the HDLZAD-TTMOA model attained a lower CT of 0.95 over other methods. Table 4 depicts the ablation study of the HDLZAD-TTMOA methodology. The HDLZAD-TTMOA approach attained higher performance, while combinations such as HCNN+WOA, HCNN+WOA+ECOA, TCN+WOA, TCN+WOA+ECOA, LSTM+WOA, and LSTM+WOA+ECOA reached lower outcomes. Table 5 represents the computation performance assessment of the HDLZAD-TTMOA approach based on Floating-Point Operations (FLOPs), Graphics Processing Unit (GPU), and inference time [34]. The HDLZAD-TTMOA approach portrayed superior efficiency with a FLOP of 0.008 G, GPU memory utilization of 849 M, and an inference time of 1.22 seconds, outperforming existing models in speed and resource consumption. The CLAD, AutoEncoder, and DUAD consumed 6.610 G FLOP, 4194 M GPU, and 4.81 seconds, 0.010 G FLOP, 2876 M GPU, and 8.13 seconds, and 1.060 G FLOP, 2865 M GPU, and 6.09 seconds. Furthermore, the Deep SVDD required 0.170 G FLOP, 4245 M GPU, and 8.41 seconds, whereas Renoir needed 7.030 G FLOP, 4721 M GPU, and 8.85 seconds, MLP utilized 1.330 G FLOP, 2185 M GPU, and 8.63 seconds, and the Siamese Network consumed 0.400 G FLOP, 4133 M GPU, and 4.03 seconds. This indicates that the HDLZAD-TTMOA model achieved a substantial decrease in computation cost and memory.

Table 3. Results of the HDLZAD-TTMOA model with existing methods

Method	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{Score}$	CT
NB	61.31	60.84	68.48	75.54	2.70
CNN Model	83.15	78.86	69.49	77.50	2.55
DNN Algorithm	90.91	69.89	71.92	68.89	2.34
KNN Classifier	92.95	73.97	70.91	71.20	5.40
SVM Model	78.32	70.05	70.84	76.79	2.65
RF	94.02	71.26	77.86	74.66	7.99
DT	82.97	76.49	68.86	75.71	5.03
HDLZAD-TTMOA	97.73	92.01	91.84	91.91	0.95

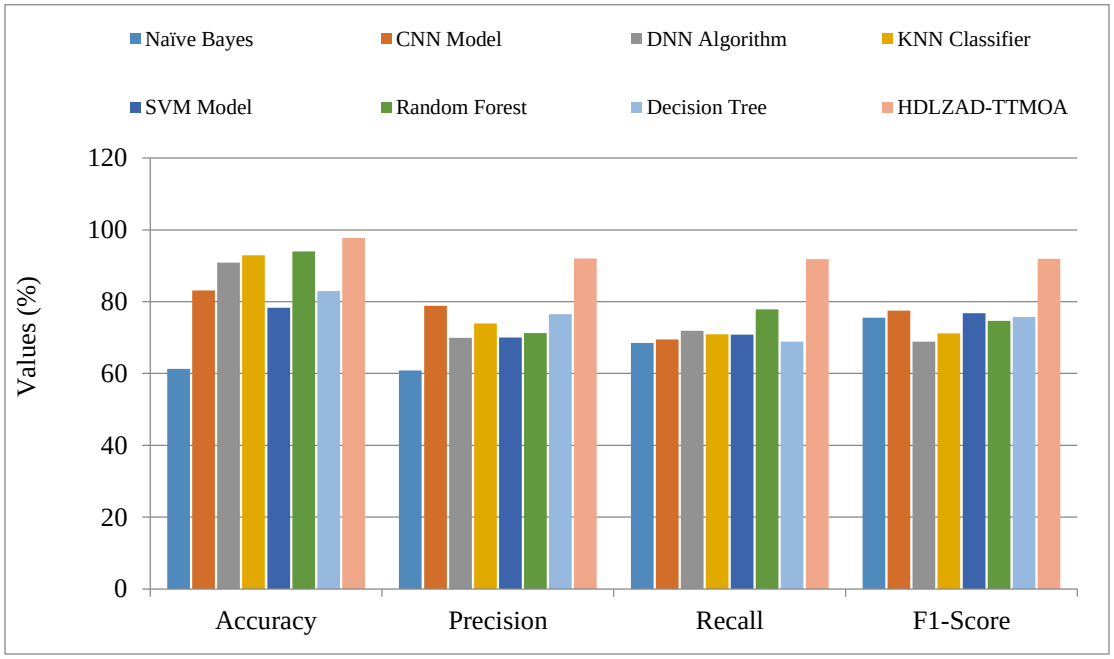

Fig. 12 Comparison of the HDLZAD-TTMOA system with other systems

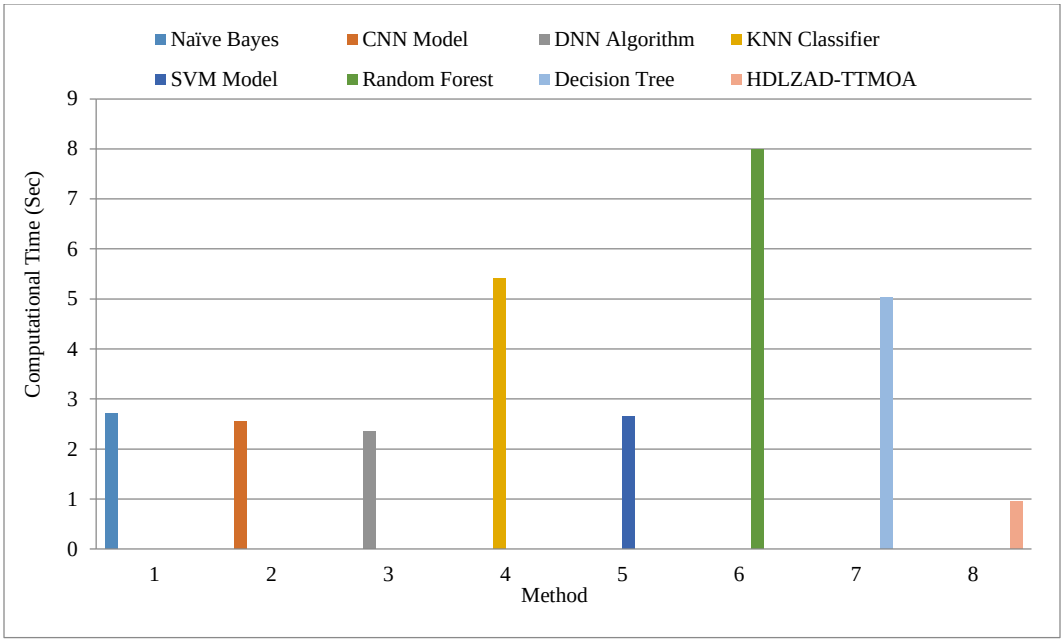

Fig. 13 CT analysis of HDLZAD-TTMOA model with existing approaches

Table 4. Ablation study analysis of the HDLZAD-TTMOA model with existing approaches

Method	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{Score}$
HCNN+WOA (With FS without optimization, TCN, and LSTM)	97.73	92.01	91.84	91.91
HCNN+WOA+ECOA (With FS and optimization without TCN and LSTM)	97.73	92.01	91.84	91.91
TCN+WOA (With FS without optimization, HCNN and LSTM)	97.73	92.01	91.84	91.91
TCN+WOA+ECOA (With FS and optimization without TCN and LSTM)	97.73	92.01	91.84	91.91
LSTM+WOA (With FS without optimization, TCN, and HCNN)	97.73	92.01	91.84	91.91
LSTM+WOA+ECOA (With FS and optimization without TCN and HCNN)	97.73	92.01	91.84	91.91
HDLZAD-TTMOA (Hybrid model with FS and ECOA optimization)	97.73	92.01	91.84	91.91

Table 5. Evaluation of the HDLZAD-TTMOA method based on different measures

Model	FLOPs (G)	GPU (M)	Inference Time (s)
CLAD	6.610	4194	4.81
AutoEncoder	0.010	2876	8.13
DUAD	1.060	2865	6.09
Deep SVDD	0.170	4245	8.41
Renoir	7.030	4721	8.85
MLP	1.330	2185	8.63
Siamese Network	0.400	4133	4.03
HDLZAD-TTMOA	0.008	849	1.22

5. Conclusion

In this manuscript, the HDLZAD-TTMOA model is proposed. The aim is to improve a new zero-day attack detection and classification mechanism using advanced optimization models. Here, the FS process is performed by using WOA. Moreover, a hybrid CNN-TCN-LSTM network has been used for attack detection. Finally, the ECOA-based hyperparameter selection is accomplished. The efficiency of the HDLZAD-TTMOA technique is examined against two benchmark datasets. The comparison study of the HDLZAD-TTMOA technique illustrated improved efficacy.

The limitations include restrictions in real-world applicability and scalability when handling large-scale IoT and IoV environments. There also exists a deployment issue with limited edge devices. It is also required to examine lightweight, adaptive frameworks to maintain high detection performance.

References

[1] Benedetto Marco Serinelli, Anastasija Collen, and Niels Alexander Nijdam, "On the Analysis of Open Source Datasets: Validating IDS Implementation for Well-Known and Zero Day Attack Detection," *Procedia Computer Science*, vol. 191, pp.192-199, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Rotem Bar, and Chen Hajaj, "Simcse for Encrypted Traffic Detection and Zero-Day Attack Detection," *IEEE Access*, vol. 10, pp. 56952-56960, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xiaoyan Sun et al., "Using Bayesian Networks for Probabilistic Identification of Zero-Day Attack Paths," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2506-2521, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mahmut Tokmak, "Deep Forest Approach for Zero-Day Attacks Detection," *Innovations and Technologies in Engineering*, pp. 45-56, 2022. [Google Scholar]

[5] Antonio Gonzalez Pastana Lobato et al., "An Adaptive Real-Time Architecture for Zero-Day Threat Detection," *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, pp. 1-6, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Nerella Sameera, and Mogalla Shashi, "Deep Transductive Transfer Learning Framework for Zero-Day Attack Detection," *ICT Express*, vol. 6, no. 4, pp. 361-367, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Agathe Blaise et al., "Detection of Zero-Day Attacks: An Unsupervised Port-based Approach," *Computer Networks*, vol. 180, pp. 1-19, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Segun I. Popoola et al., "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930-3944, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Umme Zahoora et al., "Zero-Day Ransomware Attack Detection using Deep Contractive Autoencoder and Voting based Ensemble Classifier," *Applied Intelligence*, vol. 52, no. 12, pp.13941-13960, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ahmed Sleem, "Intelligent and Secure Detection of Cyber-Attacks in Industrial Internet of Things: A Federated Learning Framework," *Full Length Article*, vol. 7, no. 1, pp. 51-61, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mohanad Sarhan et al., "From Zero-Shot Machine Learning to Zero-Day Attack Detection," *International Journal of Information Security*, vol. 22, no. 4, pp. 947-959, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mahdi Soltani et al., "An Adaptable Deep Learning-based Intrusion Detection System to Zero-Day Attacks," *Journal of Information Security and Applications*, vol. 76, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yali Wu et al., "An Active Learning Framework using Deep Q-Network for Zero-Day Attack Detection," *Computers and Security*, vol. 139, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mahmoud M. Badr et al., "Comparative Analysis Between Supervised and Anomaly Detectors Against Electricity Theft Zero-Day Attacks," *2024 International Telecommunications Conference (ITC-Egypt)*, Cairo, Egypt, pp. 706-711, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Belal Ibrahim Hairab et al., "Anomaly Detection based on CNN and Regularization Techniques against Zero-Day Attacks in IoT Networks," *IEEE Access*, vol. 10, pp. 98427-98440, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Alok Kumar Shukla, "An Efficient Hybrid Evolutionary Approach for Identification of Zero-Day Attacks on Wired/Wireless Network System," *Wireless Personal Communications*, vol. 123, no. 1, pp.1-29, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Christopher Redino et al., "Zero Day Threat Detection using Graph and Flow based Security Telemetry," *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, pp. 655-662, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Abubakar Wakili, and Sara Bakkali, "ZeroDefense: An Adaptive Hybrid Fusion-based Intrusion Detection System for Zero-Day Threat Detection in IoT Networks," *Journal of Electronic Science and Technology*, vol. 24, no. 1, pp. 1-14, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[19] Mona Almofarreh et al., "Boosting Cybersecurity: A Zero-Day Attack Detection Approach using Equilibrium Optimiser with Deep Learning Model," *CMES Computer Modeling in Engineering and Sciences*, vol. 145, no. 2, pp. 2631-2656, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ahmed Hasan Dakheel, Ali Hasan Dakheel, and Anas Qays Flayyi, "A Bayesian-Optimized Random Forest Framework for Zero-Day Threat Detection in IoT Environments," *International Journal of Intelligent Engineering and Systems*, vol. 19, no. 1, pp. 803-822, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ahmed A. Mohamed et al., "Zero-Day Exploits Detection with Adaptive WavePCA-Autoencoder (AWPA) Adaptive Hybrid Exploit Detection Network (AHEDNet)," *Scientific Reports*, vol. 15, no. 1, pp. 1-27, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Aamir S. Ahanger et al., "A Deep Learning Approach for the Detection of Zero-day Attacks," *Deep Learning for Intrusion Detection: Techniques and Applications*, pp. 267-283, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[23] Nahla J. Abid, Nawaf Alhebaishi, and Turki Althaqafi, "Robust Zero-Day Attack Detection with Optimal Deep Learning for Securing Internet of Things Environment," *Journal of Intelligent Systems and Internet of Things*, vol. 16, no. 1, pp. 118-131, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Rahila Rahim, and Mohammad Ahsan Chishti, "Deep Learning-based Intrusion Detection in Wireless Networks," *Deep Learning for Intrusion Detection: Techniques and Applications*, pp. 209-232, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[25] Amal Mirza et al., "ZDBERTa: Advancing Zero-Day Cyberattack Detection in Internet of Vehicle with Zero-Shot Learning," *Computers*, vol. 14, no. 10, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[26] Md. Johirul Islam et al., "Application of Min-Max Normalization on Subject-Invariant EMG Pattern Recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Abdul Wadood et al., "Design of a Novel Fractional Whale Optimization-Enhanced Support Vector Regression (FWOA-SVR) Model for Accurate Solar Energy Forecasting," *Fractal and Fractional*, vol. 9, no. 1, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[28] Farhad Hosseinali et al., "Evaluation of A Hybrid CNN-TCN-LSTM Model for Traffic Flow Prediction," *Earth Observation and Geomatics Engineering*, vol. 7, no. 2, 2023. [Google Scholar]

[29] Wenping Xiang et al., "Research on End-Effector Position Error Compensation of Industrial Robotic Arm based on ECOA-BP," *Sensors*, vol. 25, no. 2, pp. 1-25, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[30] CIC-ToN-IoT: UNSW-ToN-IoT, with CICFlowmeter Features, by the University of Queensland, Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/dhoogla/cictoniot

[31] Network Intrusion Dataset (CIC-IDS-2017): Anomaly Detection in Network Dataset, Kaggle, 2017. [Online]. Available: https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset

[32] Belal Ibrahim Hairab et al., "Anomaly Detection of Zero-Day Attacks based on CNN and Regularization Techniques," *Electronics*, vol. 12, no. 3, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Mingcan Cen et al., "Zero-Ran Sniff: A Zero-Day Ransomware Early Detection Method based on Zero-Shot Learning," *Computers and Security*, vol. 142, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Jack Wilkie et al., "A Novel Contrastive Loss for Zero-Day Network Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 2064-2076, 2026. [CrossRef] [Google Scholar] [Publisher Link]