Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

An efficient DDoS attack detection method in IoT based on an optimized recurrent neural network using hybrid swarm intelligence methods

Dr.S.Anitha[1*], Dr T.Ramaprabha[2], Mrs.S. Chandrakala[3], Mr. N. Nijanthan[4]

[1]Assistant Professor, Department of Computer Science and Applications , Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam, Tamil Nadu, India, Email: selvianitha1974@gmail.com

[2]Associate Professor, Department of Computer Science, Nehru Arts and Science College, Coimbatore-641105, Tamilnadu, India, Email: nascramaprabhacs@nehrucolleges.com

[3]Assistant Professor/Cyber Security , Paavai Engineering College (Autonomous)- Pachal,Namakkal, Tamilnadu, India, Email: chandrakalasubramaniampec@paavai.edu.in

[4]Assistant Professor/Cyber Security, Paavai Engineering College (Autonomous), Pachal,Namakkal, Tamilnadu, India, Email: nijanthannagarajanpct@paavai.edu.in

Abstract

The recent growth of the Internet of Things (IoT) has worsened security concerns, and one of the most common threats today is the Distributed Denial of Service (DDoS) protection because of the modest computational capabilities and the heterogeneity of the devices comprising the IoT. Conventional intrusion detection systems frequently fail to distinguish between legitimate and malicious flows very well, where the former results in high false alarms and low adaptability to changing patterns of attacks. This paper has suggested an effective DDoS attack detection algorithm based on an optimized Elman Recurrent Neural Network (ERNN) with an improved algorithm, the Improved Grey Wolf Optimization (IGWO) algorithm, to overcome these limitations. The ERNN is utilized to learn the time dependencies in the IoT traffic, and the IGWO is utilized to optimize the weights, biases, and hyperparameters of the framework,including adaptive convergence control, opposition-based learning, and enhanced search strategies. Experimental analyses of various datasets of IoT devices, such as Fridge, Garage, GPS, Modbus, Light Motion, and Thermostat. The IGWO-ERNN is more accurate, precise, and recalls specific information, and has a higher F-measure, faster convergence, and lower misclassification. These findings demonstrate the strength and effectiveness of the IGWO-ERNN architecture, which would be suitable in real-time and resource-constrained IoT settings and can contribute significantly to countering DDoS attacks.

Keywords: Internet of Things; Elman recurrent neural network; grey wolf optimization; simplex method; opposition-based learning; convergence rate; population diversity;

1. Introduction

The IoT is a network of physically linked devices, including embedded software, sensors, and other technologies that enable them to collect and share data. Numerous industries are using this technology, including healthcare, home automation, and transportation. Smart thermostats, wearable devices, industrial sensors, and automated cars are a few examples of IoT devices [1]. In recent years, IoT devices have increased and extended to encompass everyday appliances and systems. It's projected that there will be an incredible 24.6 billion connected devices on the planet by 2030[2, 3].Owing to the great number of IoT devices, there is a higher chance of cyberattacks, which may be intentionally or unintentionally caused by IoT devices. One type of assault that is connected to IoT devices is a DDoS attack. DDoS attacks have become increasingly sophisticated and in number at a startling rate [4].

IoT devices are essential to many DDoS attacksdue to their Internet connectivity, lack of firewalls, and other security factors that make them possible[5]. Cyberattacks can trigger catastrophic occurrences, such as blackouts, failures in military equipment, and the exposure of confidential information. These attacks may also affect phone and computer networks, stopping users from accessing data and hindering systems[6]. IoT devices are also more susceptible to hacking as they lack the computing power required for adequate protection[7]. They can be used to launch large-scale attacks without the device owner's knowledge, which could lead to the creation of botnets. The more connected IoT devices there are, the more security measures are required to protect IoT infrastructure; this is becoming increasingly evident as the number of connected IoT devices increases daily [8]. Therefore, it's imperative to lessen the impact of DDoS attacks on IoT networks to maintain the security, reliability, and integrity of these interconnected systems, ensure the continuous delivery of essential services, and protect the interests of individuals as well as businesses. Therefore, there has been a focus on enhancing security and resistance against these types of attacks,and Figure 1 shows the overview of DDoS attacks [9].

An example of a neural network architecture is the ERNN, which is intended to process sequential input by maintaining a type of memory over time. It has been widely used in several domains where sequential dependencies are crucial since Jeffrey Elman's 1990 suggestion[10]. Elman RNNs have recurrent connections that allow them to store information from previous time steps in a state known as the hidden state. This is in contrast to feedforward neural networks, which only permit information to flow in one way (from input to output). As a result, they can capture temporal dependencies in sequential data. The hidden state of an Elman RNN serves as a memory, holding information about the history of the sequence. Since the hidden state is modified at each time step depending on the input and the hidden state preceding it, it can learn and represent complex temporal patterns. Training parameters (learning rate, batch size) and architecture (number of layers and units) must be carefully considered during design and training for Elman RNNs to function as optimally as possible. The weights and biases, learning rate, and hidden neurons are the main elements that have a major impact on how well ERNN performs. The conventional approach initializes these parameters at random. This adds to the level of uncertainty over ERNN's performance. Recently, several evolutionary optimization (EO) and swarm intelligence (SI) strategies have been used to expand ERNN performance[11].

Fig. 1 : DDoS attack in IoT environment

The present research work develops a new variant of ERNN based on IGWO for DDoS attacks in an IoT environment. An OBL is used to initialize the populations of GWO for enhancing global convergence ability, and a simplex method is used to find the local optimal values of neighborhood space to increase the likelihood of leaving local optima. This IGWO aids in improving the tradeoff between the GWO's capacity for exploration and exploitation. The suggested approach makes use of an updated GWO's improved optimization efficiency as well as the temporal pattern detection abilities of ERNN.The ERNN's hyperparameter optimization's convergence speed and accuracy are greatly increased by this improvement.

Extensive tests on benchmark datasets unique to the IoT show that the enhanced ERNN model works better than current detection techniques. The model's potential for use in actual IoT security systems is demonstrated by its effectiveness in the real-time detection of several DDoS attack vectors. By providing a strong and effective neural network-based method for DDoS attack detection, this work makes a substantial addition to IoT cybersecurity and improves the security and resilience of IoT networks.

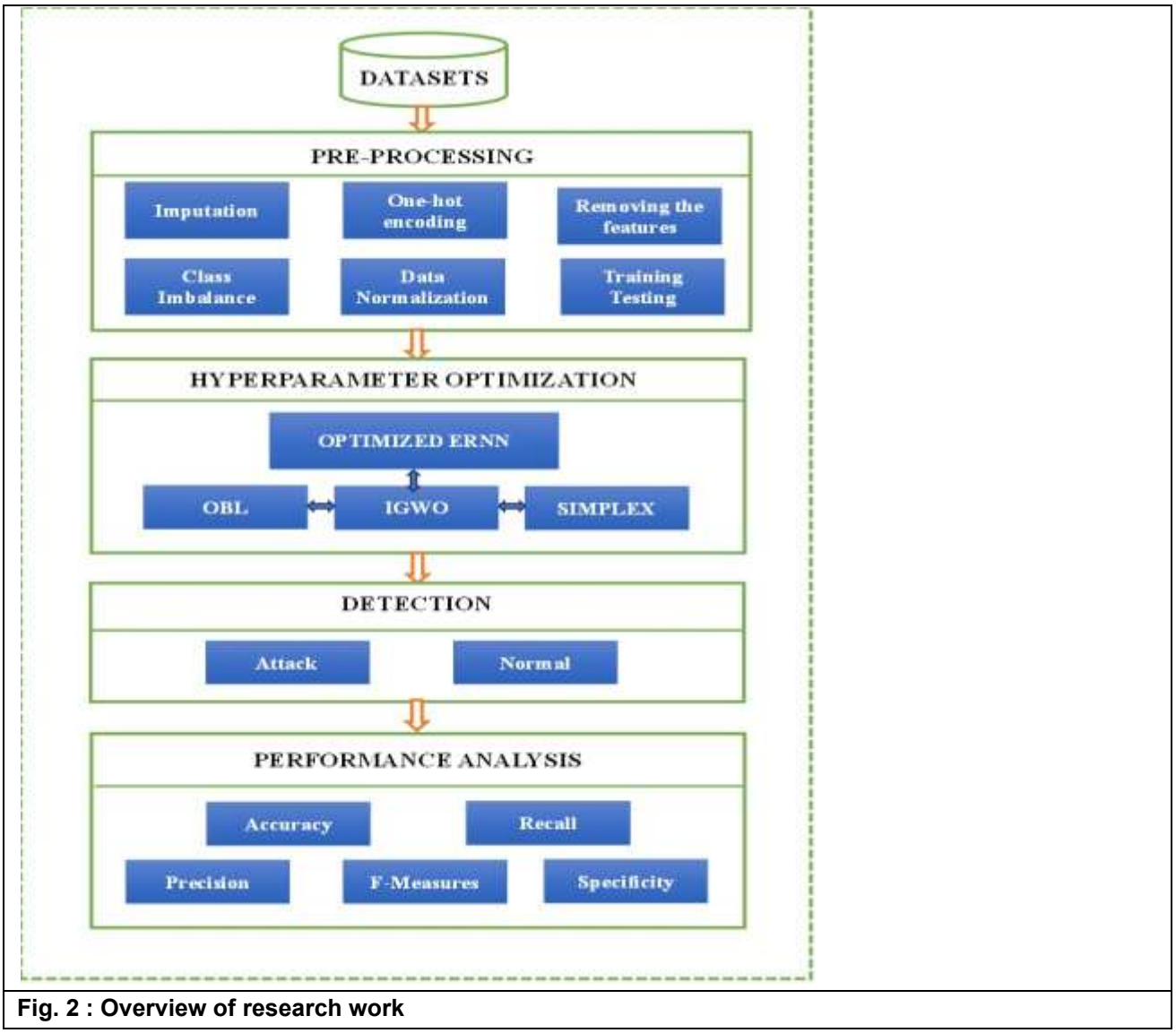

Fig. 2 : Overview of research work

Figure 2 shows the overview of the research work. The research's contribution is as follows:

- The recommended IGWO-ERNN is utilized to detect malicious movement from incoming data in an IoT setting.
- To get the right parameters, a novel method known as IGWO is created to enhance ERNN's generalization performance.

- This work presents a hybrid GWO employing OBL to address the limited population diversity and SM to enhance the convergence rate.
- The generated detection algorithm was put to the test against a few benchmark detection techniques to analyze its performance.

The structure of the paper is as follows: Some of the most current related works are covered in Section 2. Research methods and proposed IGWO-ERNN are covered in Sections 3 and 4.The experimental findings and the research project's conclusion are covered in Sections 5 and 6.

2. Related works

Reviews of the literature offer a thorough summary of all the studies that have been done on a subject. This aids in understanding the current state of knowledge and identifies knowledge gaps that can be filled by future study.In the past, numerous research projects have been undertaken to lessen the threat posed by DDoS attacks. A portion of this research and endeavors are authentic, whilst others are antiquated. These features fit the profile of DDoS botnets that exist today. DDoS malware comes in a wide variety, and some of them are constantly evolving, making it nearly hard to identify them. On infected devices, certain DDoS software may hibernate without causing problems, creating the illusion that nothing out of the ordinary is occurring until the hacker initiates the attack. Deep learning has produced fresh approaches and plans of action to address DDoS and other issues. The present sectorconverses with some current papers associatedwith DDoS attacks.

D. Negesse et al. (2026) [12]developed a new deep learning modelto detect DDoS attacks based on LSTM. The experiment results illustrated that our proposed LSTM attained 99.53% accuracy,where the LSTM can take benefit of its capability to model chronological attack signatures.B. B. Gupta et al. (2025) [13]presented a hybrid DL model for detecting cyber-attacks in the IoT context. The proposed approach employed the feature selection technique to pick efficient features and the Ant Lion Optimization algorithm to tune the hyperparameters. This hybrid method model trains for five epochs and detects attack traffic with 97% accuracy, making it both efficient and lightweight for IoT applications.R. Kumar et al. (2025) [14]present QuIDS, which uses a Quantum SVM to categorize assaults in an IoT network. QuIDS, unlike ML or DL, takes extremely minimal training data to train and effectively identify attacks in an IoT network. QuIDS captures eight flow-level features from IoT network traffic and uses over four quantum bits for training.H. A. Sakr et al. (2024) [15] developed a two-level IDS based on DL models with binary and multiclass classifiers to detect DDoS attacks in IoT networks. Developed a method that enhanced the detection and mitigation of potential security risks in IoT networks. To improve performance, we did preprocess operations on the dataset, including random subset selection, feature elimination, duplication removal, and normalization. The real-time effectiveness of numerousDL models, as well as detection performance, has been assessed.S. Kalvikkarasi et al. (2024) [16]proposed a new DDoS attack detection method based on an optimized ERNN based on bacterial colony optimization with centroid opposition-based learning (COBL). The traditional BCO lacks population diversity and falls into local optima due to its random initialization and population update. To improve population diversity and prevent local optima problems, COBL is utilized for population initialization and population update. The COBCO algorithm enhances the ENN's ability to explore and exploit the solution space by mimicking bacterial foraging behavior, which raises the network's convergence speed and detection accuracy.

M. T. Hussan et al. (2023) [17] developed a novel approach to DDoS attack detection: CBCO-ERNN, an enhanced ERNNusing chaotic BCO (CBCO). The suggested approach makes use of CBCO to determine the ideal ERNN architecture's structure,such as several hidden neurons and parameters,such as weights and biases. Through bacterial population initialization and chemotaxis step size value selection, the chaos theory is utilized to enhance BCO's exploration and exploitation capabilities. To improve the convergence rate and prevent local optima, the ERNN model is trained using the CBCO method.S. Yadav et al. (2024) [18]suggest a threat mitigation technique for IoT networks based on the RNN algorithm. RNN classifies characteristics linked to assaults by using pre-processed and feature-extracted data. The min-max scaling strategy is used to preprocess the datasets before the XBoost model selects features.M. Ali et al. (2024) [19]developed a botnet detection method based on ACLR n to improve security measuresfor IoT systems. ACLR, the combination of an artificial neural network (ANN), a convolutional neural network (CNN), a long short-term memory (LSTM), and the RNN

stacking method for botnet identification, is suggested (ACLR).V. Saravanan et al. (2024) [20]suggested using the RNN model in conjunction with the Blockchain-based African Buffalo (BbAB) system to identify intrusions and improve security. Additionally, the system gathers and trains datasets from both malware and regular users. Identity-Based Encryption (IBE) is used to encrypt the dataset. The blockchain in the cloud safely stores the encrypted data. From that point on, RNN was used to identify the intrusion in a cloud environment. In the RNN prediction stage, African Buffalo Optimization (ABO) was applied to provide ongoing intrusion monitoring.

N. F. Syed et al. (2023) [21]suggest a two-step procedure to make DL-based IDS for fog nodes easier to implement. To accomplish this, the time-series-based IoT network data is first divided into attack classes, converting multi-class problems into binary-class problems. The next step is to reduce the amount of data needed to train the deep learning models by using basic feature reduction techniques like Chi-Square Statistic, Mutual Information, and Group Method of Data Handling (GMDH). After the DL algorithm has been trained using the smaller datasets, an optimized DL model is installed on edge or fog nodes to identify IoT assaults. These actions decrease computation time and network latency while enabling the dispersal of DL workloads.S. A. Khanday et al. (2023) [22]developed a DDoS attack detection method based on feature selection. To extract feature importance from the input feature vector using an ensemble strategy and then employ feature importance and impurity, a novel dataset pre-processing method. Next, features with the least amount of importance are dropped using the feature drop-out technique. The manuscript presents a set of classifiers for anomaly identification in IoT network traffic data, encompassing both ML and DL models, along with a comparative performance study of each classifier. Detecting DDoS attack patterns and the dissemination of malware capable of causing DoS attacks continues to be the main focus of the research.V. Gaur et al. (2022) [23]use a hybrid technique to choose features, using machine learning classifiers to apply feature selection techniques for the early identification of DDoS attacks on IoT devices.M. B. Farukee et al. (2021)[24]created a novel way of detecting IoT device vulnerabilities using multilayer perceptrons and CNN. Random Forest is combined with another feature selector to form the basis of the models. Accurately and promptly detecting DDoS attacks is the primary goal of the suggested models.U. M. Badamasi et al. (2020) [25]propose a DL-based CUDA-enabled LSTM technique for the detection of DDoS attacks. Wani et al. (2020)[26]proposed an SDN-based security technique for identifying and thwarting DDoS assaults in IoT networks. SDN separates data planes from control planes and provides a flexible method of managing and controlling networks. It makes networks programmable, which can be used to offer a practical defense against catastrophic attacks in IoT networks.

J. Galeano-Brajones et al. (2020)[27]recommended testing of an entropy-based technique for identifying and reducing DoS and DDoS assaults in IoT settings using a stateful SDN data layer. The obtained results prove the effectiveness for the first time in targeting real-world IoT data flow. S.-H. Lee et al. (2022)[28]proposed an autonomous security system that can identify the type of attack and detect if an IoT data server is being attacked by a DDoS using edge computing and a two-dimensional CNN. A trained two-dimensional CNN can achieve 99.8% and 99.5% accuracy for training packet characteristics and packet traffic, respectively.M. Shurman et al. (2020)[29]suggested two methods for identifying Distributed Reflection DoS (DDoS) assaults in the IoT. To identify IoT-DoS attacks, the first methodology makes use of a hybrid IDS. Based on LSTM and trained with the most recent dataset for these types of DDoS, the second methodology makes use of DL models.R. Doshi et al. (2018) [30]have shown that high-accuracy DDoS detection in IoT network data can be achieved with a range of ML techniques by exploiting IoT-specific network behaviors to influence feature selection. M. Roopak et al. (2020) [31]suggested a feature selection (FS) approach that uses multi-objective optimization to detect DDoS attacks in IoT networks. To decrease the dimensionality of the data and enhance the IDS's effectiveness, FS is necessary. To solve the optimization challenge, they applied the non-dominated sorting algorithm with its modified jumping gene operator. They then used an extreme learning machine (ELM) as the classifier for FS based on six crucial goals for an IoT network. F. A. F. Silveira et al. (2020) [32]suggest an IoT controller detection module that classifies network traffic using ML algorithms. Three real, well-known datasets from the literature were used to test the system on an emulated platform, which was created in the context of SDN.

3. Research methods

3.1 Elman recurrent neural network

Elman (1990) created an RNN variant known as an ENN. The context layer or recurrent layer in an ENN provides feedback directly from the hidden layer outputs. The feedback connection can hold state information and display the time lag between input and output patterns. ENN has a local memory function as a result. Time series prediction, sequence analysis, and wind forecasting are just a few of the numerous fields in which ENN is frequently employed. An input layer, a hidden layer, an output layer, and a recurrent or context layer make up an ENN. A nonlinear function of the weighted sum of the input samples is calculated by one or more neurons in each layer to transfer information or samples from one layer to another. The input layer's mathematical model is described as follows:

$X_{it}(k) = \sum_{i=1}^{n} X_{it}(k-1)$ (1)

where X_{it} is the collection of input vectors at time t and n is the number of neurons in the input layer. The expression for the input model of every neuron in the hidden layer is:

$net_{jt}(k) = \sum_{i=1}^{n} W_{ij} X_{it}(k-1) + \sum_{j=1}^{p} C_j R_{jt}(k)$ (2)

In this case, $C_j$is the weight between the hidden and recurrent layers, and $W_{ij}$is the weight between the input and hidden layers. The following is the definition of the hidden layer's output:

$Z_{jt}(k) = f(net_{jk}(k)) \sum_{i=1}^{n} W_{ij} X_{it}(k-1) + \sum_{j=1}^{p} C_j R_{jt}(k)$ (3)

The forecast model can contain dynamic feedback and storage thanks to the recurrent layer. The recurrent layer's output is calculated as follows:

$R_{jt}(k) = Z_{jt}(k-1)$ (4)

The layer's output is computed as follows:

$Y_t(k) = f\left(\sum_{j=1}^{p} V_j Z_{jt}(k)\right)$ (5)

3.2 GWO

Animals with a rigid social hierarchy are grey wolves. In general, they are classified into four levels: α, β, δ, and ω. Their social order descends from the top to the bottom. Grey wolves' social structure and hunting habits serve as the foundation for the mathematical modeling of GWO. The main tasks of hunting are to track, encircle, and assault animals. The following is a definition of tracking and encircling modeling:

$G = \left|C \otimes X_p(t) - X(t)\right|$ (6)

$X(t+1) = X_p(t) - A \otimes G$(7)

The distance between the grey wolf and the prey is represented by G, and the current iteration number is indicated by t. The positions of the prey and the grey wolf are denoted by X_p and $X(t)$, respectively. The coefficient vectors A and C are computed as follows:

$A = 2a * r_1 - a$ (8)

$C = 2r_2$ (9)

where the random vectors $r_1$and $r_2$have uniform distributions, and each component is in the interval [0,1]. In the series of iterations depicted in the following equation, the value of ais lowered linearly from 2 to 0:

$a = 2 - 2 * (t/t_{max})$ (10)

The following equations demonstrate that ω wolves update their locations under the leadership of α, β and δwolves, who are said to have the finest understanding of the position of prey, making them the best, second-best, and third-best wolves, respectively.

$\boldsymbol{G}_\alpha = |C_1 \otimes X_\alpha(t) - X(t)|$ (11)

$\boldsymbol{G_\beta} = \left|C_2 \otimes X_\beta(t) - X(t)\right|$ (12)
$\boldsymbol{G_\delta} = |C_3 \otimes X_\delta(t) - X(t)|$ (13)
$X_1 = X_\alpha(t) - A_1 \otimes G_\alpha$ (14)
$X_2 = X_\beta(t) - A_2 \otimes G_\beta$ (15)
$X_3 = X_\delta(t) - A_3 \otimes G_\delta$ (16)
$X(t+1) = (X_1 + X_2 + X_3)/3$ (17)

Where $\boldsymbol{G_\alpha}$,$\boldsymbol{G_\beta}$, and $\boldsymbol{G_\delta}$ stand for the separations, respectively, between the present grey woland δ wolves. The relative positions based on α, β, and δwolves are represented by the symbols $\boldsymbol{X_1}$,$\boldsymbol{X_2}$, and $\boldsymbol{X_3}$.A multiplication by entry is shown by $\otimes$. According to the explanations above, GWO typically explores when |A|>1 and exploits when |A|<1. Furthermore, after every iteration, ω changes to become one of the top three agents. GWO can therefore perform effective local searches. The grey wolf's difficulty near the prey is determined by parameter C. |C|<1 is advantageous to the local search in the late stage of the search, whereas |C|>1 is advantageous to the global search in the early search stage.

3.3 Opposition-based learning (OBL)

OBL is a novel notion in artificial intelligence that Tizhoosh first introduced and has since been used in several optimization techniques. An effective demonstration of the opposite operation utilized in DE to solve the optimization problem has been made. Evolutionary optimization techniques generally begin with some preliminary solutions and work toward improving them to some ideal answers[33].When a few predetermined conditions are met, the search procedure comes to an end. When we don't know the answer ahead of time, we typically begin with educated assumptions. The gap between these initial estimations and the best solution influences several factors, including computing time. By concurrently verifying the opposing solution, we might increase the likelihood that we will begin with a closer solution. This allows the better one to be selected as the starting solution. In actuality, a guess is 50% more likely to be off than the answer than its opposite guess,

Algorithm 1: GWO

Step 1:Initialize the population at random and set parameters parameters a, A, and C

Step 2:Determine each search agent's fitness value and choose X_α,X_β, and X_δ

Step 3: For all search agent

Step 3.1: Each search agent's position should be updated

Step 4: End for

Step 5: Limit each agent's updated position boundaries

Step 6:Determine each agent's updated fitness value and update $\boldsymbol{X_\alpha}$,$\boldsymbol{X_\beta}$, and $\boldsymbol{X_\delta}$using the greedy approach.

Step 7: Update A, C, and a

Step 8:Return to step 3 if the termination state is not fulfilled

according to probability theory. Thus, accelerating convergence may be possible if the closer of the two guesses is started first. Often, population-based optimization approaches begin by making a collection of solutions.

Several studies have been carried out to mitigate these shortcomings by utilizing the benefits of the OBL method for population initialization and updating.Over the past few years, the OBL has gained popularity as a method for improving the efficiency of many machine learning algorithms. It is claimed that the OBL can increase population variety and boost the capacity for worldwide searching. For instance, the OBL method is used by several neural networks [34, 35]and nature-inspired optimizations [36-41]techniques to accelerate their rate of convergence. An approach to develop optimization methods that move against the current solution is provided by the OBL algorithm. After contrasting it with the other option, the current solution is determined to be the best choice. This OBL strategy yields a solution that rapidly gets close to the optimal answer. The OBL method is described in the ensuing subsections.

a) Opposite number

Let's say that x is the true number. It is located between $m$and n : $x \in [m, n]$. By $\bar{x}$, we obtain the opposing number x. Here is a clear definition of the opposing solution, $\bar{x}$:

$\bar{x} = (m + n) - x$ (18)

For multidimensional space: Let x be a sample in D - dimensional space, where $x = (`x_1, x_2, \ldots, x_D)$. Whereas and $x_1, x_2, \ldots, x_D \in R$ and $x_i[m_i, n_i]$ $\forall i \in \{1,2, \ldots . D\}$. Thus, in D -dimensional space, the opposite estimate $\bar{x}$ is defined by $\bar{x}_i, \ldots . \bar{x}_D$

$\bar{x}_i = m_i + n_i - x_i, \quad i = 1,2, \ldots, D$(19)

b) Opposition-based optimization

Let $g(.)$be a suitable evaluation function and let $f(x)$be the function. If x is a random beginning value in the range of $x \in [a, b]$and $\tilde{x}$ is x's opposite value. For each iteration, estimate the values of $f(x)$and $f(\tilde{x})$.After if $g(f(x) \geq g(f(\tilde{x})$, the learning process proceeds with x. Otherwise with $\tilde{x}$. The evaluation function $g(.)$is represented using the algorithm that is provided.

3.4 Simplex method

Spendley et al. (1962)[42] first presented the simplex approach, which is characterized by a set of points that is one greater than the set of dimensions of the search space. Numerous features of the simplex method include its quick search speed, small computer footprint, and strong local searching capability[43, 44]. The following is a description of the SM method's detailed procedure:

Step 1: Examine every solution (bacteria) in the population. Choose the top-performing bacteria globally, X_g and the second-best bacteria, X_b assuming that X_s is needs to be switched. The definitions of these three fitness value locations are $f(X_g)$,$f(X_b)$, and$f(X_s)$.

Step 2: To find the intermediate location X_c between two points, likeX_g and X_b apply the formula below.

$X_c = \frac{X_g + X_b}{2}$ (20)

Step3: The following is how $X_r$determines the reflection point:

$X_r = X_c + \alpha(X_c - X_s)$ (21)

Here, α represents the reflection coefficient, with a value of 1.

Step 4: A comparison procedure is carried out between the global best and the reflection point. Provided that $f(X_r) < f(X_g)$to solve the following equation:

$X_e = X_c + \gamma(X_r - X_c)$ (22)

where the extension coefficient, denoted by γ, is typically set to 2. Next, compare the global best $X_g$with the fitness value of the extension point X_e. If $(X_e < X_g)$ then X_s should be used in place of X_e. X_s will then take the place of X_r.

Step 5: There are comparisons made between X_r and X_s. The comparison operation should be completed using the following equations if $f(X_r) > f(X_s)$,

$X_t = X_c + \beta(X_s - X_c)$ (23)

where the condense coefficient, denoted by β, is typically set to 0.5. Next, the fitness values of the condense point $X_t$and the point X_s are compared. In the event that $f(X_t) <$
$f(X_s)$, $X_s$should be used in place of X_r. If not, $X_s$will take X_r's position.
Step 6: To get the condense point X_w, $f(X_g) < f(X_r) < f(X_s)$, shrink operations are performed. This has the following definition:
$X_w = X_c - \beta(X_s - X_c)$ (24)
Here, β is the coefficient of shrinkage. $X_s$must be swapped for X_w if $f(X_w) < f(X_s)$. *If not, $X_r$will* take the place of X_s.

3.5 Improved GWO (IGWO)

The purpose of this work is to overcome the slow convergence and population diversity of the classic GWO algorithm by proposing animproved GWO with an OBL and SM technique. The suggested algorithm's convergence is accelerated by the SM technique to enhance population diversity and by the OBL method to

avoid local optima. To speed up the search and prevent the algorithm from running more iterations without seeing any improvement, we introduced an IGWO approach in this study.

3.5.1 Population initialization

An OBL helps optimization algorithms perform better by taking into account both the opposite of a potential solution and its own. Applying OBL to GWO population initialization can improve variety and raise the likelihood of discovering a global optimum.Through the creation of a population, the OBL algorithm enhances the provided answer. The opposite answer, $\tilde{x}_i$, for x_i, to create an initial population, X. The optimal starting values for the initial population of a given solution can be found by computing the fitness function values for both current and opposite values and then comparing the fitness values. To choose the new population of the optimal solution—known as the starting population—fitness function values are employed.

3.5.2 Simplex method

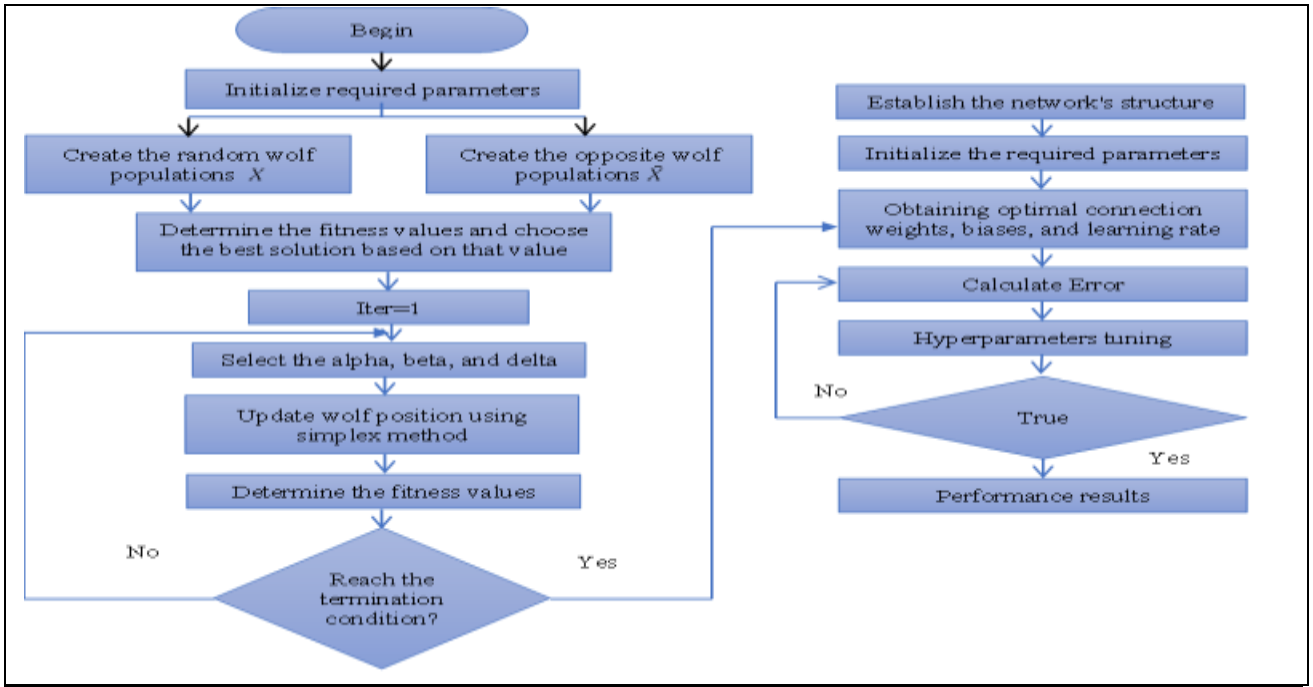

Fig. 3 : Overview of the IGWO-ERNN detection method

Algorithm 2: Improved GWO

Step 1: Initialize the population of grey wolves

Step 2: Evaluate the fitness of each wolf

Step 3: Compute the opposite population

Step 4: Evaluate the fitness of the opposite population

Step 5: Select the best individuals from the initial and opposite populations

Step 6: While the termination condition is met do

Step 6.1: Update the positions of alpha, beta, and delta wolves

Step 6.2: For each wolf do

a) Initialize a simplex around the current position

b) Perform the simplex operations (reflection, expansion, contraction, and shrinkage)

c) Evaluate the fitness of the new positions

d) Update the position to the one with the best fitness

Step 6.3: End for

Step 6.4: Update alpha, beta, and delta based on the new positions

Step 7: End while

A population is directed toward the global best value by the optimal solution, which is a major factor in GWO throughout iterations. However, population search stagnation is likely to occur if the ideal solution gets stuck at a local optimal value. The simplex contracts to the final minimum after adjusting to the local environment, as per the analysis presented. Consequently, we employ the complicated technique disturbance on the ideal solution during repetitions to investigate the local optimal value's neighborhood space and increase the probability of emerging from the local optimum. We can more efficiently use search space and locate better people faster because of this disruption of the local perfect solution.

4. Proposed IGWO-ERNN

Elman initially chooses the initial values of the parameters at random, then uses network training to continuously update the sample space until he finds the optimal set of parameters that best matches thefeatures of the sample set. The prediction effect of the network predictor is diminished and the training process is more likely to veer toward local extremes as a result of the blind selection of starting parameters during the training phase. Finding the ideal parameters at the outset is so essential to develop. A stronger network architecture. In the iterative procedure, the network structure can be more effectively trained using the ideal network parameters. To enhance the ERNN for DDoS attack detection in IoT, a new IGWO is trying to do so. For assessing its performance, mean square error (MSE) is typically employed as anobjective function measure. A reduced MSE signifies enhanced DDoS precision. Because of this, the suggested IGWO algorithm uses the minimization of MSE as its objective function. The algorithm known as IGWO is used to give the ERNN's hyperparameters. By adjusting the hyperparameters, IGWO minimizes the MSE of the model. With the α wolf having a significant influence on hyperparameter tuning, the IGWO searches for the optimal weight to maximize ELMAN neural network error minimization while simultaneously improving convergence and accuracy. Figure 3 shows the overview of the proposed IGWO-ERNN detection method

5. Experimental results and analysis

The DDoS attacks, in which a large number of devices—such as IoT devices—are used as bots to submit fake requests to services, causing them to become unavailable. A trustworthy process ofdetection based on suitable techniques is required to identify and detect if such attacks have taken place in a network or not. The present research work focused on a new variant of optimized ERNN proposed to detect DDoS attacks in IoT environments. The ToN-IoT dataset is considered for analyzing the performance of DDoS attack methods[45]. The performance analysis of the proposed IGWO-ERNN method is compared with GWO-ERNN[46], APSO-ERNN[47], PSO-ERNN[48], GA-ERNN[48], ERNN[49], BPNN[50], and SVM[51].

5.1 Datasets details

The ToN-IoT datasets include a diversity of data sources, including operating system logs and network traffic from IoT/IIoT services, as well as heterogeneous data collected from telemetry data. These sources were obtained from a realistic representation of a medium-scale network built at the UNSW Canberra's Cyber Range and IoT Labs. This work primarily focuses on the suggested dataset of IoT/IIoT service telemetry data and its attributes. You can get the ToN-IoT datasets at the ToN-IoT repository. The suggested datasets were labeled with two features: a type feature that indicated the attack sub-classes for multi-class classification tasks and a label feature that indicated whether an observation is normal or an attack. Nine (9) different cyberattack types, including “scanning, denial-of-service (DoS), distributed denial-of-service (DDoS), ransomware, data injection, backdoor, cross-site scripting (XSS), password cracking assault, and man-in-the-middle (MITM)”, were launched against different IoT and IIoT sensors throughout the IIoT network.The dataset's details are available on the UNSW website.There are seven (7) Train-Test IIoT datasets in total, one for each of the IIoT devices: Thermostat, Garage Door, Motion Light, GPS Tracker, and Fridge. A variety of IoT/IIoT scenarios were simulated in the testbed to create the dataset. Smart cities, smart homes, and smart manufacturing are a few common IoT/IIoT applications that could use these scenarios. For example, most smart houses have sensors for motion lights, garage doors, and refrigerators. One can find the GPS sensor in smart cities. Most industrial and smart manufacturing applications include Modbusand thermostats. A detailed description of the datasets is shown in Table 1.

Algorithm 3: Detection approach

Input: Testing data

Output: Labeled data

Step 1: While x in X Test do

Step 2:$x' \leftarrow$ Model Predict(x)

Step 3:$\delta \leftarrow RE(x, x')$

Step 4: If $\delta \leq th$ then

Step 4.1: Label as normal

Step 5: Else

Step 5.1: Labelas an attack

Step 6:End if

Step 7:End while

Table 1 : Datasets details						
Types	**Fridge**	**GPS**	**Garage**	**Thermostat**	**Motion_light**	**Modbus**
Normal	15,000	15,000	15,000	15,000	15,000	15,000
Password	5,000	5,000	5,000	5,000	5,000	5,000
Scanning	0	550	529	61	1,775	529
XSS	2,042	577	1,156	449	449	577
DDoS	5,000	5,000	5,000	0	5,000	0
Ransomware	2,902	2,833	2,902	2,264	2,264	0
Injection	5,000	5,000	5,000	5,000	5,000	5.000
Backdoor	5,000	5,000	5,000	5,000	5,000	5,000
Total	***39,944***	***38,960***	***39,587***	***32,774***	***39,488***	***31,106***

5.2 Data preprocessing

Cleaning and preparation are the most important steps in ensuring excellent performance before feeding the data into detection techniques. The dataset presents numerous hurdles for our research, including missing values, categorical characteristics, and class imbalance. There are superfluous characteristics that could degrade the performance of the chosen ML algorithm. We used a variety of preparation strategies, drawing from the literature, and assessed the chosen detection approaches using combinations of different preprocessing and normalization strategies.

i. ***Imputation :*** Missing values are a common occurrence in the massive ToN-IoT dataset. Correct handling of these values is necessary to build a valuable analysis. In the suggested model, the most frequent value in each feature—which comprises missing values—replaces the imputation of missing values.

ii. ***Converting categorical features:*** The ToN-IoT dataset has multiple properties that fall into categories. It is necessary to translate the category qualities into numerical values. One-hot encoding was used to achieve this. In this work, categorical characteristics are transformed via label encoding. For example, '0' and '1' were assigned to the temperature feature in the Fridge dataset, which has categorical values 'high' and 'low'.

iii. ***Removing features:*** Date, time, and timestamp were removed from feature vectors because they have the potential to cause some machine learning techniques to overfit the training set.

iv. ***Class imbalance:***Distributions with class imbalances afflict the ToN-IoT dataset. The imbalanced problem was addressed by proposing oversampling, under-sampling, and hybrid techniques. Replicating the minority class points is known as oversampling. It has been utilized by several researchers.Nevertheless, this method's drawback is that it exaggerates these points. Some employ under-sampling, which deducts a certain number of points from the dominating class. The challenge with this approach is that some aspects that are removed can be essential for accurately portraying the class. Use a hybrid approach where some majority class points are subtracted, and minority class scores are repeated. The Synthetic Minority Oversampling Technique (SMOTE) [52]tackles the oversampling problem that may occur with simple random oversampling and provides synthetic minority class samples to enhance basic random oversampling.Because SMOTE creates original data points as opposed to replicating already existing ones. To generate additional minority data points, a linear combination of two comparable minority samples is employed. New feature values are reliably interpolated between the nearest neighbors of the minority sample.

v. ***Data normalization:*** A model may be biased toward the large feature values when certain features have larger values than others, which might produce unreliable findings. Therefore, by scaling features within a range between [0.0,1.0] without altering the normalcy of data behavior, data normalization plays a crucial role in preventing features with big values from outweighing the features with smaller values. The feature values are scaled within [0.0,1.0] using the min-max normalization technique.

$$x_{New} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (25)$$

$x_{new}, x, x_{min}$and x_{max} are is the normalized, present, minimum, and is the maximum value of features.Initially, we separated the dataset into two segments (i.e., train and test split). Eighty percent of the data was utilized for training and assessing the chosen machine learning techniques, while the remaining twenty percent of the data

was reserved for the testing dataset, which allowed us to assess the models (i.e., trained classifiers) using additional, unseen data.

5.3 Hyperparameters optimization

Enhancing the performance and generalization capacity of ERNNs requires careful consideration of hyperparameter adjustment. The present research work uses the improved GWO for obtaining optimal hyperparameters of ERNN in order to enhance accuracy. IGWO strikes a balance between hyperparameter space exploration and exploitation. Finding the optimal hyperparameters in the high-dimensional space usually connected to ERNNs requires striking this equilibrium. By preventing local optima and conducting a global search, IGWO's capability guarantees the discovery of optimal hyperparameters. The IGWO is used to find and tune the hyperparameters of ERNN, such as weights and biases, number of hidden neurons, andLearning rate. The strength of the connections between neurons is represented by the ***weights***. The network can identify intricate patterns and connections in the input data by properly adjusting the weights. ***Biases*** allow the activation functions to be altered, which gives the model more flexibility. The model's capacity to match the data may be enhanced by this adaptability. The ***hidden layer's neuron*** count has an impact on the model's ability to recognize intricate patterns. Neurons in excess can produce overfitting, whereas an inadequate number might result in underfitting. The model's parameters are updated based on the loss gradient to the extent that is determined by the ***learning rate***. A model that has the right learning rate will converge more quickly and perform better; a model with the wrong learning rate will diverge or converge too slowly. For ERNNs, GWO's function in hyperparameter optimization is essential to accomplishing successful and efficient model training. GWO is a time and resource-efficient method of improving model performance, robustness, and generalization by methodically exploring and utilizing the hyperparameter space. Because of this, GWO is an effective method for optimizing intricate neural network designs, such as ERNNs.The value ranges and details are shown in Table 2. The optimalparameter values are obtained average values of 20 independent runs.

Table 2 : Parameters settings

ERNN		GWO	
Parameter	**Value**	**Parameter**	**Value**
Activation functions	TanH, Sigmoid	Population size	30
Loss function	MSE	Maximum iterations	500
Learning rate	0.01	Control parameter $\vec{a}$	0.5
Epochs	1000	Population communication coefficient c_1	0.5
Expected error	0.0005	Individual memory coefficient c_2	0.5
Weight range	-1 and 1	w	0.8
Dropout	0.1		
Batch size	32		
Numbers of hidden neurons	10-100		

5.4 Performance measures

Performance metrics are essential for identifying DDoS attacks that employ optimized ERNN. These metrics aid in assessing how well the neural network model identifies between legitimate and malicious traffic in terms of effectiveness, efficiency, and dependability.Several performance metrics are used to assess the efficacy of detection techniques in the context of identifying DDoS attacks in an IoT environment. These metrics are essential for evaluating the detection system's precision, effectiveness, and overall performance. The prediction algorithm's performance was evaluated in this study using five performance measures: accuracy, specificity, recall, precision, and f-measure. These metrics are calculated using a confusion matrix.

- ***Accuracy*** isthe percentage of successfully identified instances (attacks and non-attacks) out of all occurrences is known as accuracy, and it expresses the detection system's overall correctness.

  $$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} (26)$$

- ***Specificity*** quantifies the percentage of genuine non-attacks (or legitimate traffic) that the detection system accurately classifies as such:

$$Specificity = \frac{TN}{TN+FP} \quad (27)$$

- ***Recall*** quantifies the percentage of real DDoS attacks that the system accurately identifies. It shows how well the system can detect actual attacks.

  $$Recall = \frac{TP}{TP+FN} (28)$$

- The percentage of real attacks, or true positive detections, among all positive detections (including false positives) is quantified by ***precision***. It displays the system's accuracy in determining which of the elevated alarms are actual attacks.

  $$Pr\,e\,cision = \frac{TP}{TP+FP} (29)$$

- The harmonic mean of recall and precision is the F-measurethat strikes a balance between precision and recall as follows,

  $$F - measure = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad (30)$$

- **True positive (TP) is t**he number of times a DDoS attack has been accurately detected.False Positive **(FP)** is the quantity of cases in which the detection system misclassifies regular traffic as a DDoS attack.True Negative (TN) is the quantity of accurately detected cases in which there is regular network traffic and no DDoS attack taking place.False Negative (FN) is the number of cases in which a DDoS attack is misclassified as regular traffic by the detection system because it is unable to recognize it.

5.5 Results analysis

The experimental results for each device's datasets are displayed in this section.An essential part of the research and development process for DDoS attack detection utilizing optimized ERNN is the analysis and discussion of the results. They shed light on the model's functionality, point out its advantages and disadvantages, and provide suggestions for future developments.We conducted multiple trials with various values of hyperparameters to achieve the best outcomes. Choosing the finest values of the hyperparameters is essential to establishing an effective DL design, in light of the behavior of the trained model based on these parameters. We assessed the performance of the model by obtaining optimal hyperparameters using IGWO for ERNN. The final findings were calculated as the average value of all the evaluation metrics. The results analysesareconducted based on training, testing, computation time, and convergence analysis.Training entails gathering the dataset, identifying and extracting characteristics, and selecting and fine-tuning the model. Its goal is to develop a model that can precisely identify DDoS attacks in IoT.To verify the model's efficacy, it must be evaluated and validated using a variety of metrics and tested in real-world situations.

Tables 3, 4, 5, 6, and 7 show the training and testing results for accuracy, precision, recall, specificity, and f-measure respectively. The developed method that the IGWO-ERNN outperformed previous detection techniques based on training results for all datasets such as Fridge, Garage, GPS, Modbus, Motion light, and Thermostat, respectively. For example, the Fridge dataset, performed the best, with training accuracy, precision, recall, specificity, precision, and F-Score values of 98.22%, 98.84 %, 98.10 %, 98.37%, and 98.45 %, respectively.The garage dataset, which performed the best, has the following values for accuracy, precision, recall, specificity, precision, and F-Score: 100%, 100%, 99.89%, 99.99%, and 99.94%, respectively. The dataset with the best performance, GPS, has the following values for accuracy, precision, recall, specificity, precision, and F-Score: 99.46%, 96.68%, 94.32%, 89.44%, and 95.39%. The Modbus dataset, which performed the best, had the following values for accuracy, precision, recall, specificity, precision, and F-Score: 82.74%, 83.81%, 82.28%, and 82.80%, respectively. The accuracy, precision, recall, specificity, precision, and F-Score values for the Motion light dataset, which had the best performance, were 73.15%, 72.59 %, 73.89 %, 73.06%, and 73.23 % respectively. The Thermostat dataset, which performed the best, had the following values for accuracy, precision, recall, specificity, precision, and F-Score: 77.31%, 85.78%, 75.59%, and 86.60%, and 82.66% respectively. Similarly, the testing results proved that the developed IGWO-ERNN method formed high performance when compared with another detection method for all datasets.

Convergence analysis guarantees a steady and effective training procedure that produces a perfect model. To get the optimum performance, it entails keeping an eye on learning curves and adjusting hyperparameters. Figures 4, 5, 6, 7, 8, and 9 show the convergence analysis of compared methods for Fridge, GPS, Garage, Thermostat, Motionlight, Weather, and Modbus datasets respectively. The best performance is defined as having the least MSE. Consequently, the new IGWO-ERNN method has produced minimal error when compared to other previously optimized methods. The new and enhanced ERNN technique features a more adaptive real-time detector that can assess dangerous data from incoming data, according to the assessment of complete performance. Computation time makes sure the model can be taught and used in realistic time frames by focusing on the effectiveness of the prediction and training processes. Figure 20 shows the computational time analysis for all datasets. The developed method such as optimized ERNN produces low computational time to achieve high detection accuracy.Our findings demonstrated that employing the ideal hyperparameters greatly enhanced the performance of our model. This emphasizes how crucial it is to select hyperparameters for deep learning models properly.

Table 3 : Performance results-based accuracy

	Fridge		Garage		GPS		Modbus		Motion light		Thermostat	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
IGWO-ERNN	**98.22**	**98.58**	**100.00**	**100**	**95.46**	**96.10**	**82.74**	**80.35**	**73.15**	**73.24**	**77.31**	**80.80**
GWO-ERNN	98.69	89.47	99.08	94.47	94.57	94.11	73.12	78.09	69.43	73.00	72.29	79.07
APSO-ERNN	97.42	93.00	98.92	88.68	84.91	92.67	80.60	80.86	72.12	61.98	71.53	80.02
PSO-ERNN	88.06	84.83	95.57	93.79	86.81	91.18	69.55	74.80	71.04	66.62	75.74	75.67
GA-ERNN	93.42	87.20	97.70	98.08	89.54	79.72	65.31	72.48	67.41	68.87	72.87	67.92
ERNN	92.56	88.27	93.29	93.01	87.07	88.08	72.80	72.48	67.18	65.13	72.68	67.32
BPNN	92.36	85.55	87.64	83.32	81.46	78.80	67.29	72.19	67.54	65.94	70.32	63.20
SVM	92.68	85.36	85.92	84.25	88.08	78.65	72.78	68.07	61.71	59.23	72.73	72.30

Table 4 : Performance results-based precision

	Fridge		Garage		GPS		Modbus		Motion light		Thermostat	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
IGWO-ERNN	**98.84**	**98.97**	**100.00**	**99.95**	**96.48**	**95.67**	**83.81**	**83.43**	**72.59**	**72.51**	**85.98**	**86.15**
GWO-ERNN	98.22	97.16	99.59	96.74	93.99	84.28	83.19	78.12	71.31	72.02	84.44	84.74
APSO-ERNN	90.52	96.52	90.82	90.84	89.98	93.33	80.85	80.22	71.21	71.69	84.44	84.20
PSO-ERNN	93.80	96.63	97.42	87.98	84.91	92.08	75.69	65.34	70.96	71.78	75.65	80.27
GA-ERNN	92.48	86.39	89.36	97.11	86.51	91.92	70.64	72.24	67.63	60.92	81.93	75.64
ERNN	91.11	92.80	96.15	92.97	90.35	82.06	73.29	68.76	69.86	70.90	81.23	73.64
BPNN	90.16	86.48	95.37	95.56	90.58	86.87	68.43	65.96	69.94	71.14	80.59	72.35
SVM	90.71	87.69	95.93	93.58	86.56	83.92	69.84	68.71	69.89	66.72	80.11	71.89

Table 5 : Performance results-based recall

Methods	Fridge		Garage		GPS		Modbus		Light motion		Thermostat	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
IGWO-ERNN	**98.10**	**98.40**	**99.89**	**98.79**	**94.32**	**95.61**	**81.82**	**86.97**	**73.89**	**73.89**	**79.59**	**81.87**
GWO-ERNN	96.66	96.37	97.56	93.34	93.45	85.89	75.95	82.50	71.98	72.45	77.45	77.94
APSO-ERNN	90.61	88.22	98.77	94.89	94.31	90.57	70.51	80.17	71.84	68.87	74.83	76.85
PSO-ERNN	91.33	85.90	93.55	95.69	88.35	92.57	75.85	75.54	71.43	71.19	70.84	67.92
GA-ERNN	91.74	92.94	97.50	88.13	91.17	91.16	74.74	68.80	70.98	66.10	74.57	73.71
ERNN	84.16	91.08	91.98	97.49	90.09	89.74	73.10	71.65	68.68	68.29	70.88	67.92

BPNN	85.98	92.29	90.63	93.74	85.84	83.29	73.05	73.73	68.44	68.50	73.61	70.62
SVM	87.48	83.02	90.89	97.63	89.56	89.65	73.42	73.19	67.89	68.91	64.47	72.65

Table 6 : Performance results-based specificity

	Fridge		Garage		GPS		Modbus		Light motion		Thermostat	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
IGWO-ERNN	**98.37**	**98.45**	**99.99**	**99.92**	**89.44**	**95.89**	**83.28**	**83.62**	**73.06**	**72.48**	**86.60**	**86.49**
GWO-ERNN	88.98	96.98	98.87	95.89	85.06	94.49	74.42	81.48	71.89	73.26	84.62	84.29
APSO-ERNN	91.21	94.49	98.41	94.35	92.90	93.29	69.89	80.21	69.78	63.23	83.48	82.78
PSO-ERNN	82.68	91.76	97.89	92.06	87.20	88.63	67.24	69.47	72.38	66.94	82.69	83.05
GA-ERNN	84.48	84.30	96.30	91.85	86.31	91.90	74.80	75.45	71.20	71.68	82.28	82.39
ERNN	86.49	87.54	88.12	89.28	80.98	85.62	73.13	69.80	70.94	64.25	81.48	73.95
BPNN	93.40	93.20	95.94	91.45	90.55	81.26	64.24	70.89	63.08	69.25	80.92	72.17
SVM	92.69	88.29	93.67	87.89	91.18	80.78	72.86	69.88	66.17	67.48	81.43	71.89

Table 7 : Performance results-based f-measure

	Fridge		Garage		GPS		Modbus		Light motion		Thermostat	
Methods	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
IGWO-ERNN	**98.45**	**98.62**	**99.94**	**99.39**	**95.39**	**96.04**	**82.80**	**85.36**	**73.23**	**73.23**	**82.66**	**83.87**
GWO-ERNN	97.44	97.28	98.56	96.36	93.72	89.76	79.40	82.84	71.64	71.88	80.80	81.06
APSO-ERNN	90.56	89.35	94.63	92.81	92.10	90.28	75.33	80.51	71.52	70.02	79.35	80.47
PSO-ERNN	92.55	89.68	95.45	96.55	86.60	88.58	75.77	75.61	71.19	71.08	73.16	71.57
GA-ERNN	92.11	92.71	93.25	88.74	88.78	88.77	72.63	69.71	69.26	66.80	78.08	77.61
ERNN	87.49	91.09	94.02	96.81	90.22	90.05	73.20	72.46	69.26	69.07	75.70	73.98
BPNN	88.02	91.21	92.94	94.55	88.15	86.78	70.66	70.98	69.18	69.21	76.94	75.27
SVM	89.06	86.69	93.34	96.77	88.03	88.07	71.58	71.48	68.88	69.39	71.45	76.20

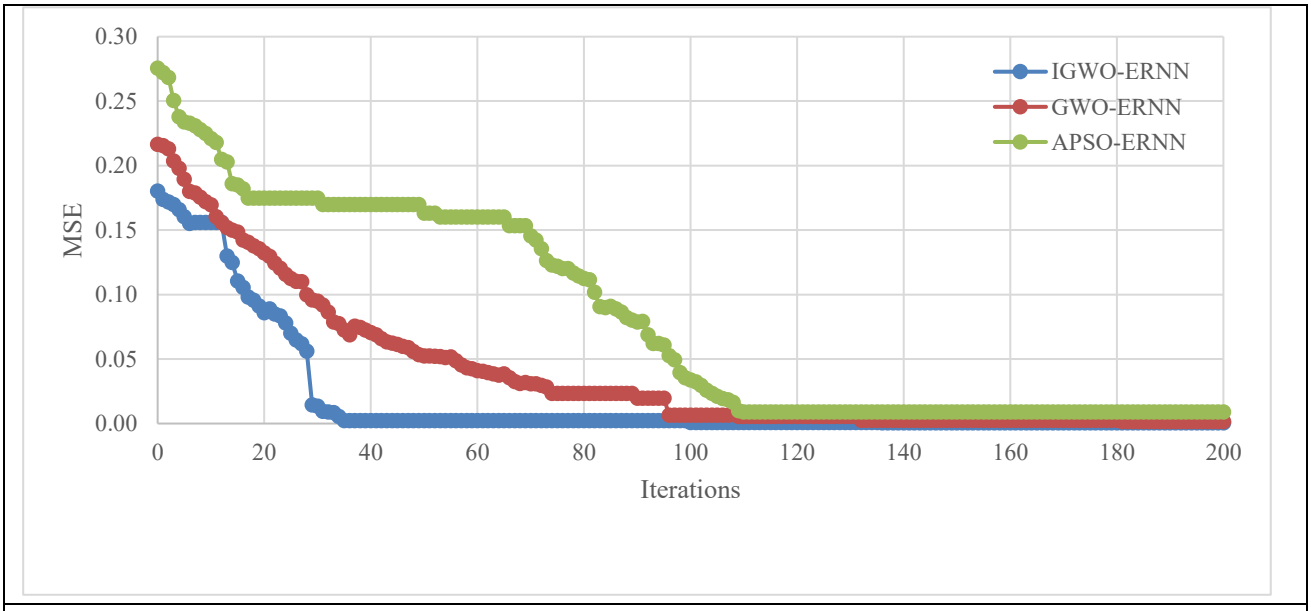

Fig. 4 : Convergence analysis for the Fridge dataset

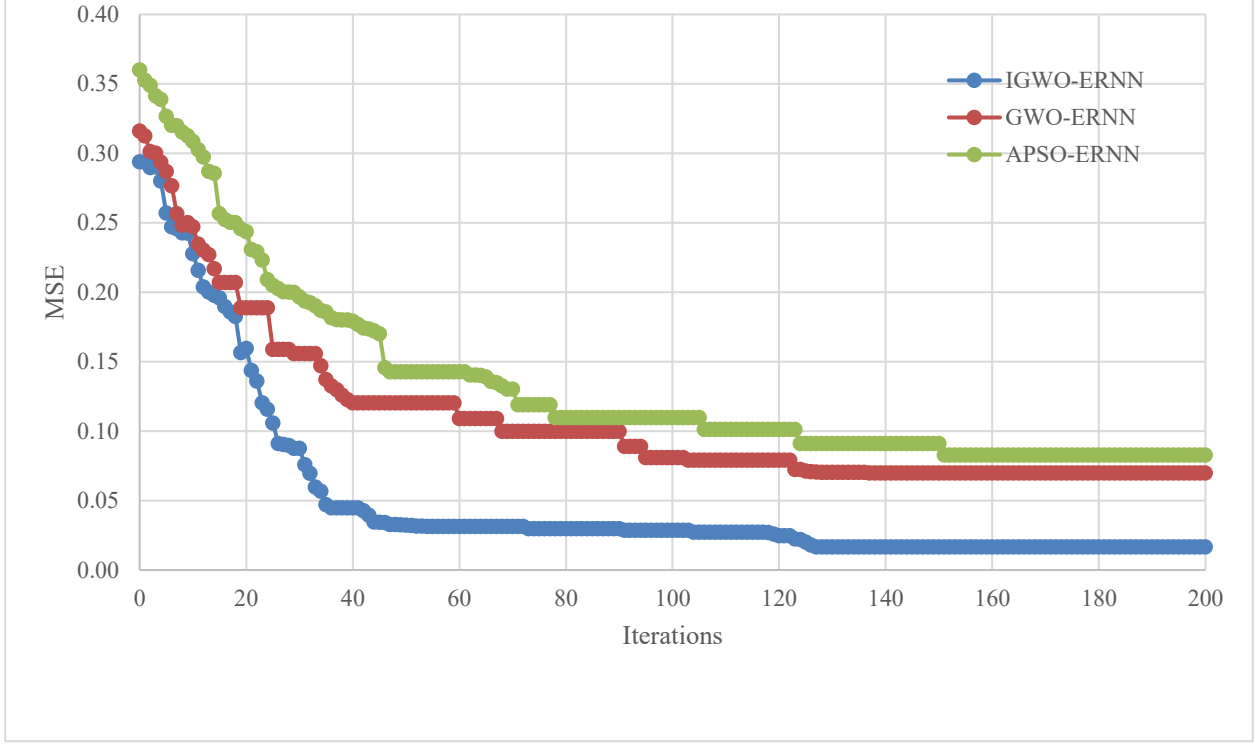

Fig. 5 : Convergence analysis for the Garage dataset

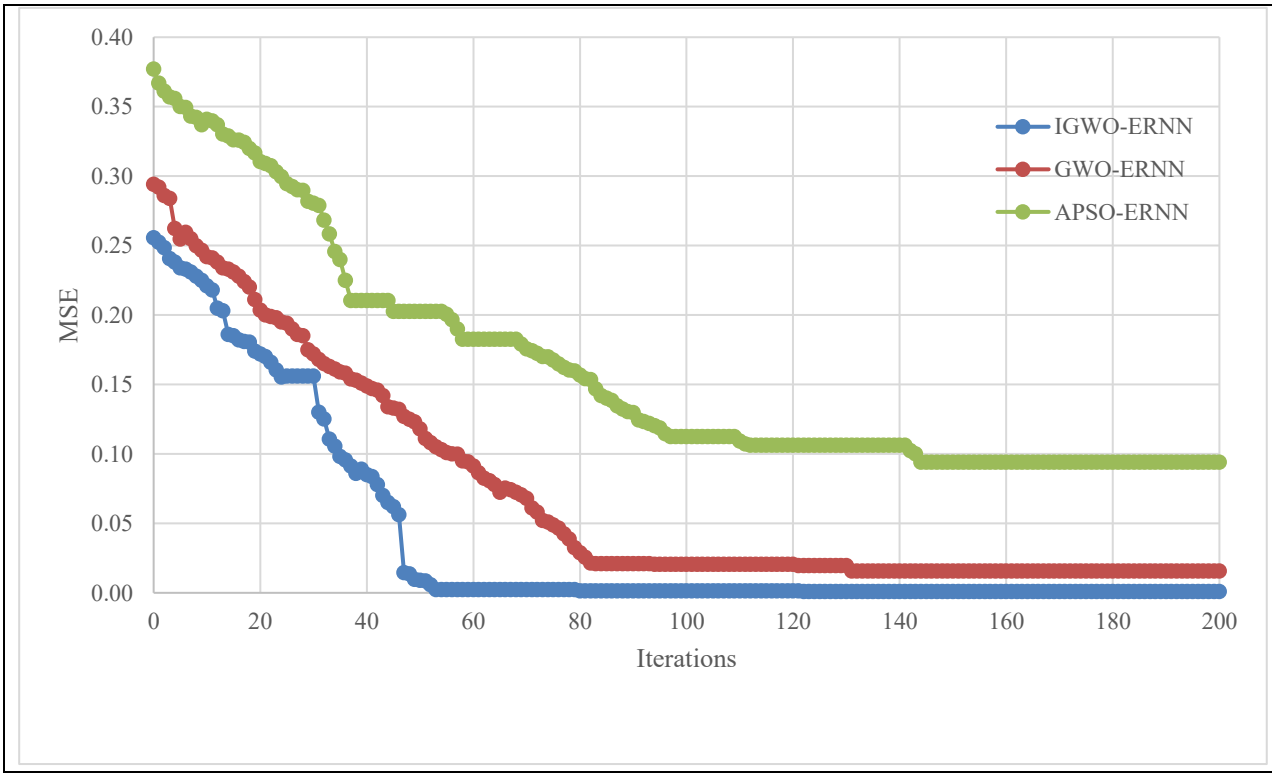

Fig. 6 : Convergence analysis for the GPS dataset

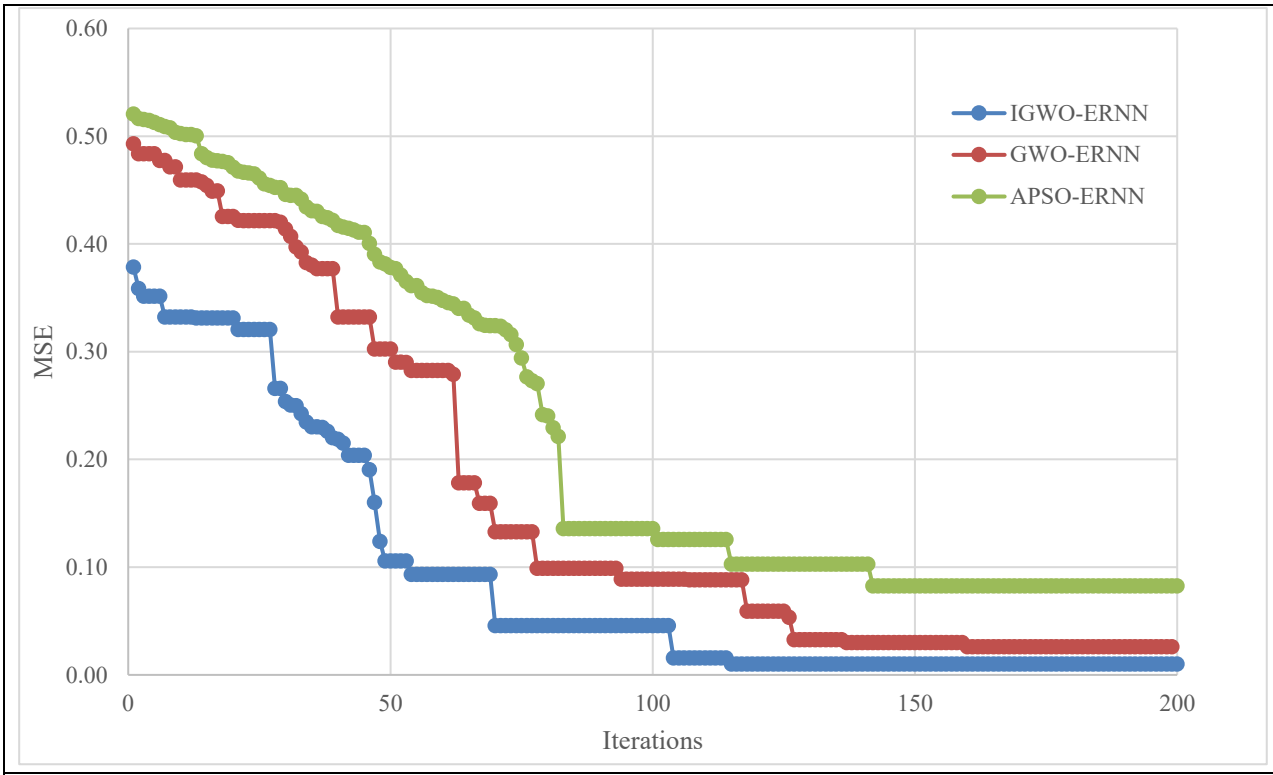

Fig. 7. Convergence analysis for Modbus dataset

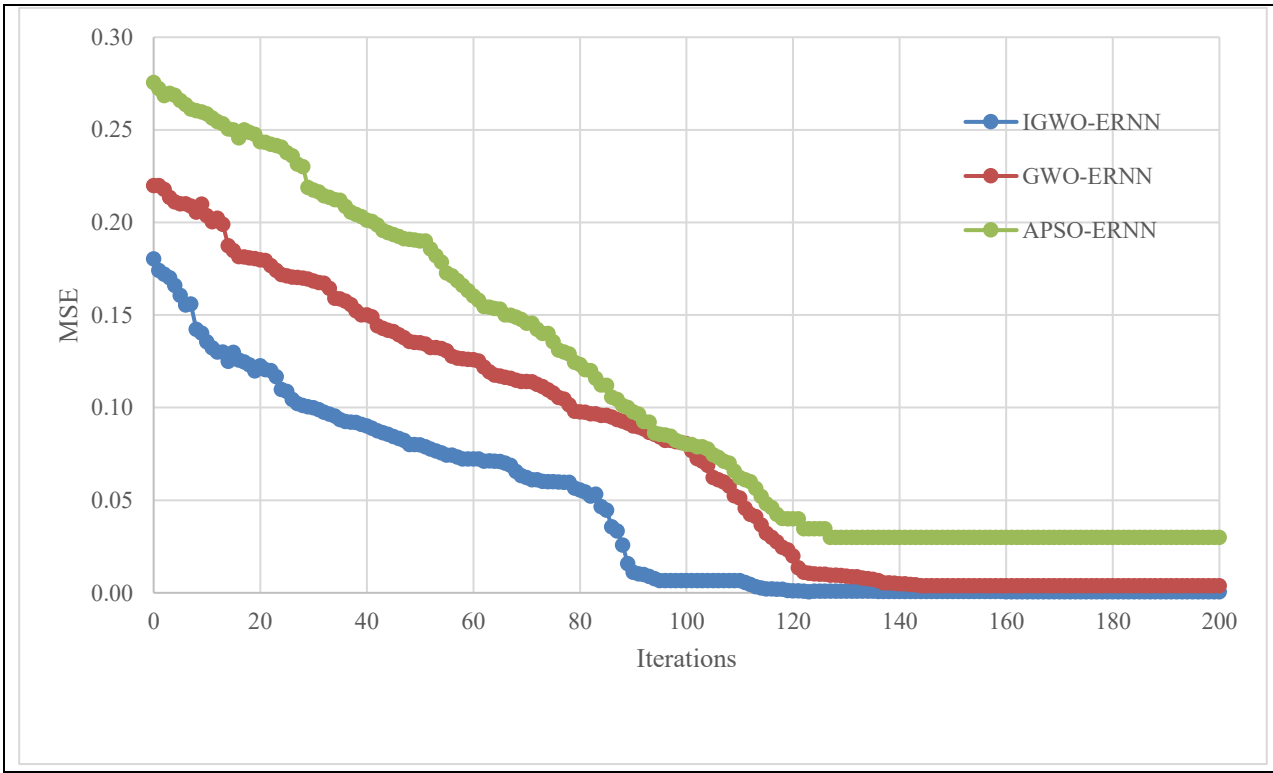

Fig. 8 : Convergence analysis for Light Motion dataset

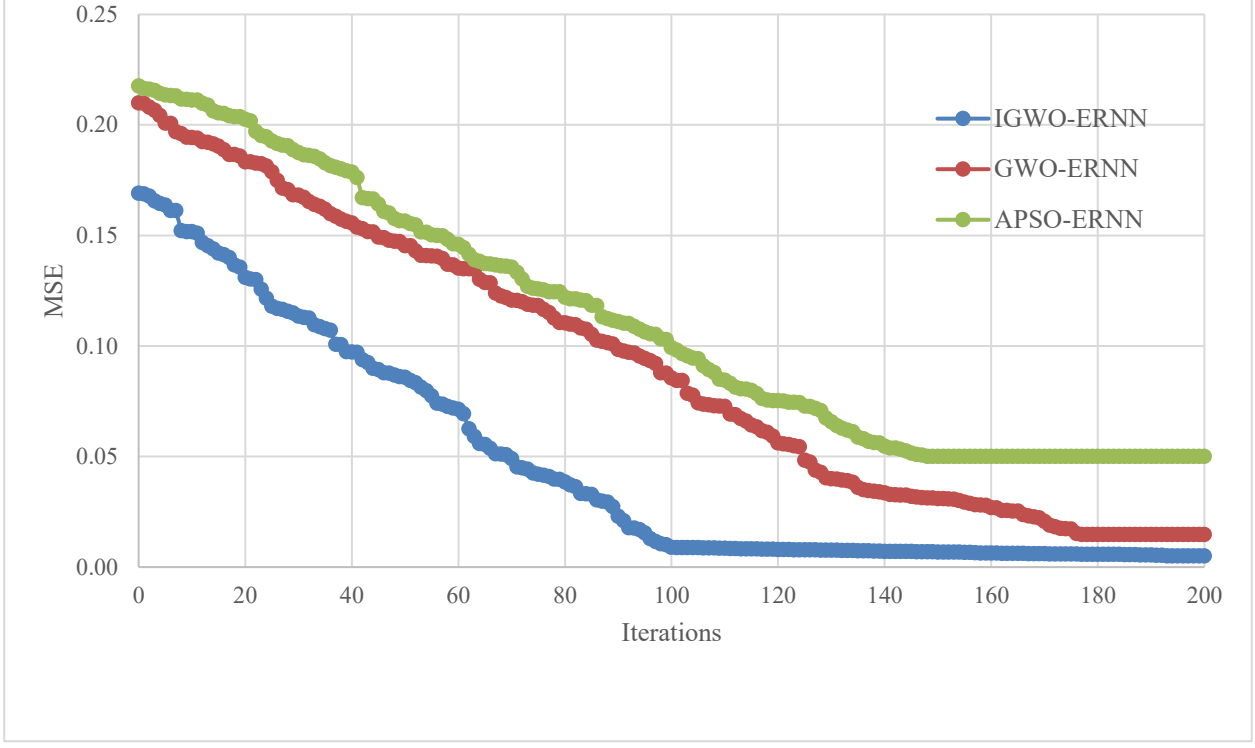

Fig. 9 : Convergence analysis for Thermostat dataset

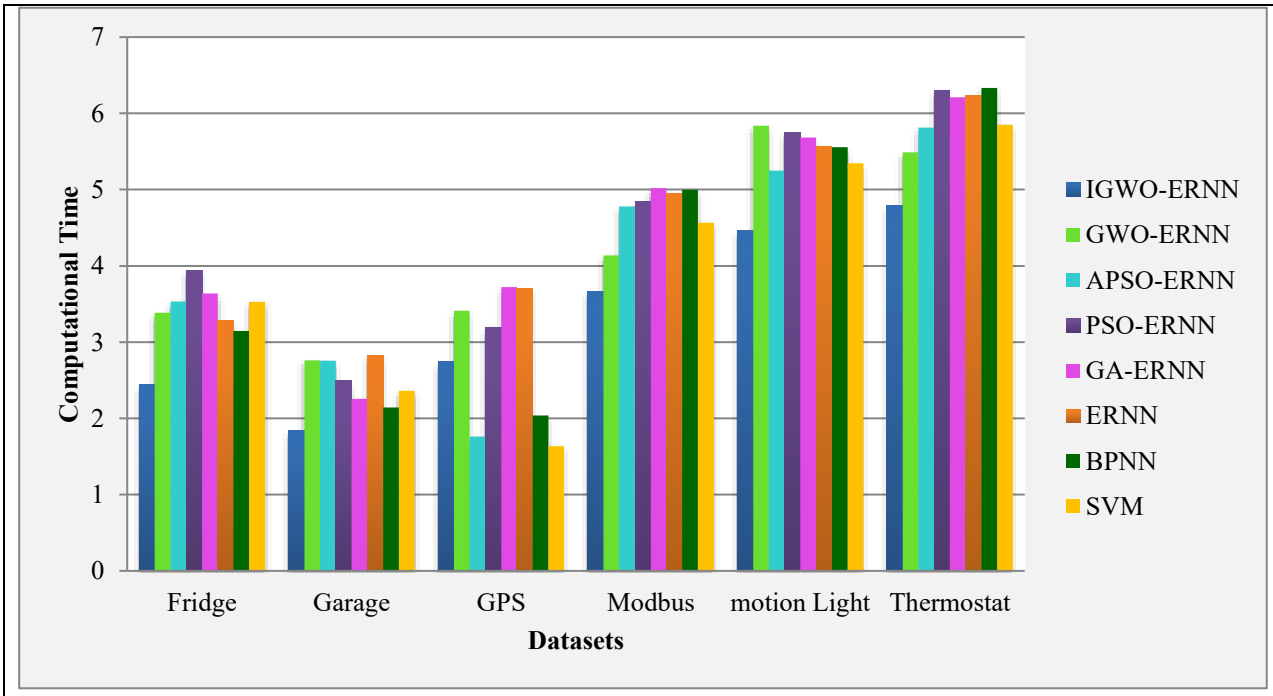

Fig. 10 : Time complexity analysis of detection methods

6. Conclusions

In this work, we developed an optimized ERNN with an IGWO called IGWO-ERNN to present a novel framework for detecting DDoS attacks in IoT scenarios. A very successful detection model was produced by combining the enhanced GWO's better optimization performance with the ERNN's potent temporal sequence learning capabilities. IGWO, which incorporates adaptive mechanisms and a better balance between exploration and exploitation, addresses numerous shortcomings of the conventional GWO. Through this modification, the

ERNN's hyperparameter tuning considerably increased in speed and accuracy, making it better able to catch the intricate patterns linked to DDoS attack traffic in IoT networks. Experimental assessments on benchmark datasets unique to the IoT revealed that our refined ERNN model routinely beat current state-of-the-art detection techniques. The model demonstrated its capacity to quickly and accurately detect several DDoS attack vectors by achieving excellent results. The suggested architecture offers a reliable, effective, and scalable approach that not only improves DDoS attack detection but also advances the subject of IoT cybersecurity. The encouraging outcomes show that our method may be successfully implemented in practical IoT security systems, greatly enhancing the security and resilience of IoT infrastructures against DDoS attacks.Successive research activities will focus on enhancing the optimization techniques, investigating alternative neural network topologies, and expanding the framework's capability to identify diverse cyber risks in IoT.

Declarations

Ethical Approval: Not applicable

Competing interests: The authors have disclosed no conflicts of interest.

Authors' contributions: The algorithms and development, as well as the paper, were contributed to by Anitha, Ramaprabha, Chandrakala, and Nijanthan. The final manuscript has been read and agreed upon by all authors.

Funding – No organizations or financial sources are helping to fund this study.

Availability of data and materials: The datasets can be downloaded from the following links

Datasets available: https://research.unsw.edu.au/projects/toniot-datasets

References

1. M. Deivakani, "Anomaly Detection in IoT Network Traffic Using Bidirectional 3D Quasi-Recurrent Neural Network Optimize With Coati Optimization Algorithm," *Transactions on Emerging Telecommunications Technologies,* vol. 36, no. 1, p. e70026, 2025.
2. R. Kavitha, K. Saravanan, S. A. Jebakumari, and K. Velusamy, "Machine learning algorithms for IoT applications," in *Artificial Intelligence for Internet of Things*: CRC Press, 2022, pp. 185-214.
3. E. Osei Owusu *et al.*, "A Systematic Review of Machine Learning and Deep Learning Techniques for DDoS Attack Mitigation in IoT and Edge Computing Systems," *Security and Privacy,* vol. 9, no. 1, p. e70147, 2026.
4. T. S. C, S. S. R.S, M. K. M, and G. P. P, "IoT-Enabled Flood Monitoring System for Enhanced Dam Surveillance and Risk Mitigation," *International Research Journal of Multidisciplinary Technovation,* vol. 6, no. 3, pp. 144-153, 05/13 2024, doi: 10.54392/irjmt24311.
5. N. Dash, S. Chakravarty, A. K. Rath, N. C. Giri, K. M. AboRas, and N. Gowtham, "An optimized LSTM-based deep learning model for anomaly network intrusion detection," *Scientific Reports,* vol. 15, no. 1, p. 1554, 2025.
6. M. Malini and N. Chandrakala, "DDoS attack detection in the cloud environment using an optimised long short-term memory with an improved firefly algorithm," *International Journal of Communication Networks and Distributed Systems,* vol. 32, no. 1, pp. 58-87, 2026.
7. B. Fathimamary, M. Nasreen, and K. Velusamy, "An Efficient DDoS Attack Detection Using Optimized Long Short-Term Optimization Based on Improved Brainstorm Optimization," *Indian Journal of Science and Technology,* vol. 19, no. 5, pp. 298-312, 2026.
8. D. N, J. Katiravan, and S. S.P, "Botnet Attack Detection in IoT Devices using Ensemble Classifiers with Reduced Feature Space," *International Research Journal of Multidisciplinary Technovation,* vol. 6, no. 3, pp. 274-295, 05/22 2024, doi: 10.54392/irjmt24321.
9. A. A. Alahmadi *et al.*, "DDoS attack detection in IoT-based networks using machine learning models: a survey and research directions," *Electronics,* vol. 12, no. 14, p. 3103, 2023.
10. H. Karamollaoğlu, İ. Yücedağ, İ. A. Doğru, S. Toklu, and İ. Atacak, "CBM-IDS: An Advanced Hybrid Deep Learning Model for DDoS Attack Detection in IoT Networks," *Journal of Universal Computer Science,* vol. 32, no. 1, p. 108, 2026.
11. A. Darwish, A. E. Hassanien, and S. Das, "A survey of swarm and evolutionary computing approaches for deep learning," *Artificial intelligence review,* vol. 53, no. 3, pp. 1767-1812, 2020.
12. D. Negesse, K. Gemeda, and G. Gianini, "DDoS attack detection and classification for the MQTT-IoT protocol using LSTM models," *Discover Applied Sciences,* 2026.

13. B. B. Gupta *et al.*, "A hybrid Ant Lion Optimization algorithm based lightweight deep learning framework for cyber attack detection in IoT environment," *Computers and Electrical Engineering,* vol. 122, p. 109944, 2025.
14. R. Kumar and M. Swarnkar, "QuIDS: A Quantum Support Vector machine-based Intrusion Detection System for IoT networks," *Journal of Network and Computer Applications,* vol. 234, p. 104072, 2025.
15. H. A. Sakr, M. M. Fouda, A. F. Ashour, A. Abdelhafeez, M. I. El-Afifi, and M. R. Abdellah, "Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems," *Egyptian Informatics Journal,* vol. 28, p. 100540, 2024.
16. S. Kalvikkarasi and A. Saraswathi, "DDoS Attack Detection in Cloud Computing Using Optimized Elman Neural Network Based on Bacterial Colony Optimization and Centroid Opposition-Based Learning."
17. M. T. Hussan, G. V. Reddy, P. Anitha, A. Kanagaraj, and P. Naresh, "DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization," *Cluster Computing,* pp. 1-22, 2023.
18. S. Yadav, H. Hashmi, and D. Vekariya, "Mitigation of attacks via improved network security in IOT network environment using RNN," *Measurement: Sensors,* vol. 32, p. 101046, 2024.
19. M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran, and I. Ashraf, "Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment," *IEEE Access,* 2024.
20. V. Saravanan, M. Madiajagan, S. M. Rafee, P. Sanju, T. B. Rehman, and B. Pattanaik, "IoT-based blockchain intrusion detection using optimized recurrent neural network," *Multimedia Tools and Applications,* vol. 83, no. 11, pp. 31505-31526, 2024.
21. N. F. Syed, M. Ge, and Z. Baig, "Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks," *Computer Networks,* vol. 225, p. 109662, 2023.
22. S. A. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks," *Expert Systems with Applications,* vol. 215, p. 119330, 2023/04/01/ 2023, doi: https://doi.org/10.1016/j.eswa.2022.119330.
23. V. Gaur and R. Kumar, "Analysis of machine learning classifiers for early detection of DDoS attacks on IoT devices," *Arabian Journal for Science and Engineering,* vol. 47, no. 2, pp. 1353-1374, 2022.
24. M. B. Farukee, M. Z. Shabit, M. R. Haque, and A. S. Sattar, "Ddos attack detection in iot networks using deep learning models combined with random forest as feature selector," in *Advances in Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December 8-9, 2020, Revised Selected Papers 2*, 2021: Springer, pp. 118-134.
25. U. M. Badamasi, S. Khaliq, O. Babalola, S. Musa, and T. Iqbal, "A Deep Learning based approach for DDoS attack detection in IoT-enabled smart environments," *International Journal of Computer Networks and Communications Security,* vol. 8, no. 10, pp. 93-99, 2020.
26. A. Wani and S. Revathi, "DDoS detection and alleviation in IoT using SDN (SDIoT-DDoS-DA)," *Journal of The Institution of Engineers (India): Series B,* vol. 101, no. 2, pp. 117-128, 2020.
27. J. Galeano-Brajones, J. Carmona-Murillo, J. F. Valenzuela-Valdés, and F. Luna-Valero, "Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: An experimental approach," *Sensors,* vol. 20, no. 3, p. 816, 2020.
28. S.-H. Lee, Y.-L. Shiue, C.-H. Cheng, Y.-H. Li, and Y.-F. Huang, "Detection and prevention of DDoS attacks on the IoT," *Applied Sciences,* vol. 12, no. 23, p. 12407, 2022.
29. M. Shurman, R. Khrais, and A. Yateem, "DoS and DDoS attack detection using deep learning and IDS," *Int. Arab J. Inf. Technol,* vol. 17, no. 4A, pp. 655-661, 2020.
30. R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018: IEEE, pp. 29-35.
31. M. Roopak, G. Y. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Networks,* vol. 9, no. 3, pp. 120-127, 2020.
32. F. A. F. Silveira, F. Lima-Filho, F. S. D. Silva, A. d. M. B. Junior, and L. F. Silveira, "Smart detection-IoT: A DDoS sensor system for Internet of Things," in *2020 international conference on systems, signals and image processing (IWSSIP)*, 2020: IEEE, pp. 343-348.
33. H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 2005, vol. 1: IEEE, pp. 695-701.
34. K. Kalaiselvi, K. Velusamy, and C. Gomathi, "Financial prediction using back propagation neural networks with opposition based learning," in *Journal of Physics: Conference Series*, 2018, vol. 1142, no. 1: IOP Publishing, p. 012008.

35. D. Bairathi and D. Gopalani, "Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks," in *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 2017: IEEE, pp. 438-444.
36. S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary computation,* vol. 12, no. 1, pp. 64-79, 2008.
37. O. P. Verma, D. Aggarwal, and T. Patodi, "Opposition and dimensional based modified firefly algorithm," *Expert Systems with Applications,* vol. 44, pp. 168-176, 2016.
38. G. Xiong, J. Zhang, D. Shi, and Y. He, "Oppositional Brain Storm Optimization for Fault Section Location in Distribution Networks," in *Brain Storm Optimization Algorithms*: Springer, 2019, pp. 61-77.
39. A. A. Ewees, M. Abd Elaziz, and D. Oliva, "A new multi-objective optimization algorithm combined with opposition-based learning," *Expert Systems with Applications,* vol. 165, p. 113844, 2021.
40. H. Muthusamy, S. Ravindran, S. Yaacob, and K. Polat, "An improved elephant herding optimization using sine–cosine mechanism and opposition based learning for global optimization problems," *Expert Systems with Applications,* vol. 172, p. 114607, 2021.
41. Z. Zhang, Z. Xu, S. Luan, and X. Li, "A Hybrid Max–Min Ant System by Levy Flight and Opposition-Based Learning," *International Journal of Pattern Recognition and Artificial Intelligence,* p. 2151013, 2021.
42. W. Spendley, G. R. Hext, and F. R. Himsworth, "Sequential application of simplex designs in optimisation and evolutionary operation," *Technometrics,* vol. 4, no. 4, pp. 441-461, 1962.
43. J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal,* vol. 7, no. 4, pp. 308-313, 1965.
44. Y. Zhou, Y. Zhou, Q. Luo, and M. Abdel-Basset, "A simplex method-based social spider optimization algorithm for clustering analysis," *Engineering Applications of Artificial Intelligence,* vol. 64, pp. 67-82, 2017.
45. A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *Ieee Access,* vol. 8, pp. 165130-165150, 2020.
46. S. Kumar Chandar, "Grey Wolf optimization-Elman neural network model for stock price prediction," *Soft Computing,* vol. 25, pp. 649-658, 2021.
47. L. Yang, F. Wang, J. Zhang, and W. Ren, "Remaining useful life prediction of ultrasonic motor based on Elman neural network with improved particle swarm optimization," *Measurement,* vol. 143, pp. 27-38, 2019.
48. P. Dixit, R. Kohli, A. Acevedo-Duque, R. R. Gonzalez-Diaz, and R. H. Jhaveri, "Comparing and analyzing applications of intelligent techniques in cyberattack detection," *Security and Communication Networks,* vol. 2021, no. 1, p. 5561816, 2021.
49. X. Tong, Z. Wang, and H. Yu, "A research using hybrid RBF/Elman neural networks for intrusion detection system secure model," *Computer physics communications,* vol. 180, no. 10, pp. 1795-1801, 2009.
50. R. M. Saad, M. Anbar, S. Manickam, and E. Alomari, "An intelligent icmpv6 ddos flooding-attack detection framework (v6iids) using back-propagation neural network," *IETE Technical Review,* vol. 33, no. 3, pp. 244-255, 2016.
51. T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence,* vol. 13, no. 2, pp. 283-294, 2020.
52. J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding," in *2006 8th international Conference on Signal Processing*, 2006, vol. 3: IEEE.
53. A. Kamalaveni. (2026). A Variational and Computational Framework for Constrained Nonlinear Optimization Problems. *Frontiers in Mathematical and Computational Research*, 9-15.
54. Gaurav Tamrakar. (2025). Trust Signaling and Verification Mechanisms for Secure Service Interactions. *Journal of Advanced Antenna and RF Engineering*, 18–24.
55. R. Rudevdagva, & G.C. Kingdone. (2025). Lip-Reading-Guided Speech Enhancement via Self-Aligning Cross-Attention Networks. *National Journal of Speech and Audio Signal Processing*, 17-25. https://doi.org/10.17051/NJSAP/01.04.03
56. F. Rahman. (2025). Adaptive Statistical Signal Processing Framework for Robust Pattern Recognition in Noisy Environments. *Journal of Integrated VLSI and Signal Intelligence*, *1*(1), 9-17.