

An optimized multilayer perceptron-based network intrusion detection using Gray Wolf Optimization

Asad Ali ^a, Muhammad Assam ^b, Faheem Ullah Khan ^b, Yazeed Yasin Ghadi ^c,
Zhumazhan Nurdaulet ^d, Alibiyeve Zhibek ^e, Syed Yaqub Shah ^f,
Tahani Jaser Alahmadi ^{g,*}

^a Department of Computer Science, University of Engineering and Technology, Peshawar, KPK Pakistan

^b Department of Software Engineering, University of Science & Technology Bannu, KPK Pakistan

^c Department of Computer Science and Software Engineering, Al Ain University, Al Ain, United Arab Emirates

^d Institute of Mechanics and Engineering, Almaty, Kazakhstan

^e Department of Software Engineering, Satbayev University, Almaty, Kazakhstan

^f Faculty of Informatics, Institute of Visual Computing & Human-Centered Technology, TU Wien, Austria

^g Department of Information Systems, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, Saudi Arabia

ARTICLE INFO

Keywords:

Intrusion Detection System
Multilayer Perceptron
Gray Wolf Optimization
Artificial Neural Network
KDD and UNSW datasets
Optimization

ABSTRACT

The exponential growth in the use of network services through the design of various network infrastructures, has led to increased complexities and challenges in the network. A major problem in computer networks is privacy and security breach. Cyber attackers exploit loopholes to infiltrate and disrupt the operation of the network through various attacks. Anomaly-based intrusion detection often employs Artificial Neural Network techniques like Multi-layer Perceptron (MLP) to classify malicious and legitimate traffic. Nevertheless, these techniques are vulnerable to overfitting and require extensive labeled data and computational resources. Consequently, this reduces the accuracy of intrusion detection systems and increases the error detection rate. To minimize the error detection rate of the intrusion detection system, it is necessary to optimize the connection parameters of the MLP neural network such as weights and biases. To this end, we proposed an optimized MLP-based Intrusion Detection using Gray Wolf Optimization (GWOMLP-IDS) to optimize the learning process of the MLP neural network by optimizing weights and biases. GWO aims to select an optimal connection parameter during the learning process to minimize the error rate of intrusion detection. Extensive simulations in Python reveal the effectiveness of the proposed approach in terms of designated performance metrics.

1. Introduction

A computer network consists of a group of nodes connected through communication media (wired or wireless), enabling end user to send and receive data. Building a computer network typically aims to achieves several goals such as sharing resources (e.g., data, software, and hardware), offering web and application services (e.g., World Wide Web and database), and communication services like social networking. Significant advancements has been made in the field of networking through the development of various types of

* Corresponding author.

E-mail address: tjahmadi@pnu.edu.sa (T.J. Alahmadi).

networks that are based on different architectures, protocols, and scale [1].

One of the primary challenges and serious problems in computer networks is privacy and security breaches [2]. Cyber attackers exploit loopholes to infiltrate and disrupt the normal operation of the network. One of the common security threats is a distributed denial of service (DDoS) attack, which involves the blockage of legitimate messages by flooding a server with malicious requests from various sources [3].

Intrusion is unauthorized access to network resources like hardware, software, and confidential data. It can take a variety of forms like eavesdropping, Man-in-Middle (MiM) attack phishing attacks, etc. There exists a set of policies and practices to prevent unauthorized entry, eavesdropping, and MiM attacks. An Intrusion Detection System (IDS) is a method that works as a gateway to monitor network traffic. The primary aim of the IDS is to identify abnormal behavior in a network and identify the privacy and security ruptures [4].

Various techniques exist to identify potential security breaches in a network, broadly, it is categorized into signature-based intrusion detection and anomaly-based intrusion detection systems. Signature-based detection methods use a database of known threats and attacks to be compared with the network traffic. Anomaly-based detection method identifies intrusions by investigating network traffic for deviations from a normal network behavior [5].

Anomaly-based detection method employs machine learning techniques that leverage large datasets to train models that can identify and recognize intrusions based on traffic patterns and anomalies. Recently, various machine learning techniques like reinforcement learning, deep learning, and artificial neural networks have been employed to create efficient network intrusion detection systems that can recognize known and unknown security threats. Multi-layer Perceptron (MLP) is a specific type of Feedforward Artificial Neural Network (ANN), commonly used as a learning algorithm to train the agent on a given data instance. During the learning process, MLP produces a new pattern that is used to envisage the output of the hidden part of the dataset [6]. However, these techniques require a large database of labeled data and computational resources. MLP models are vulnerable to overfitting and may struggle to adapt to rapidly changing attack vectors.

To minimize the inconsistency between legitimate and malicious traffic detection and the error rate of network IDS, researchers use Swarm Intelligence (SI) optimization techniques to optimize the learning process of the MLP by optimizing various parameters like activation function, weights and bias, etc.

SI techniques are a popular class of meta-heuristic algorithms. These are mostly inspired by the social relations and interactions of animals and birds such as ant colonies, pack of wolves, swarms of hawks and flocks of birds, etc. [7]. Based on their strong mathematical methodologies, they can solve various complex optimization problems like node clustering [8], task scheduling [9], image processing and cybersecurity [10]. Recently, various SI techniques are applied to MLP learning process to optimize MLP parameters [11–13]. These techniques select optimal MLP parameters to minimize the error rate of network IDS. However, due to unavoidable reasons such as computational intensive operations, lack of trade-off between exploration and exploitation phases, inefficient use of stochastic operators, and local stagnation, these techniques do not always obtain optimal solutions. This could reduce the efficiency of the network IDS by leaving malicious traffic as legitimate traffic in the network.

Given that optimizing MLP connection parameters is a complex optimization problem, we utilize the Gray Wolf Optimization (GWO) algorithm as a mathematical methodology to optimize the bias and weight parameters of the MLP learning process to minimize the error rate of network IDS. GWO belongs to the family of SI algorithms which is inspired by the leadership hierarchy and hunting maneuver of wolves [14]. The pack of wolves are divided into four groups i.e., alpha, beta, omega, and delta. Alphas are the leaders who decide on prey searching, hunting sleeping etc. Beta, omega, and delta are the subordinate wolves that follow the decision of the alpha wolves. GWO has shown efficient performance in solving various complex optimization problems due to its simplicity, high repertoire of stochastic operators, and balanced behavior between the search phases (exploration and exploitation phase). Moreover, GWO's high exploration and convergence rate enables the algorithm to quickly obtain the global optimal solution.

To the best of our knowledge, this is the first attempt to employ hybrid GWO and MLP for a network intrusion detection system. In the proposed hybrid approach, problem space is initialized with the population size and number of maximum iterations. GWO takes the biases and weight of the MLP learning process as input and considers them as potential candidate solutions. The fitness function is used to evaluate each potential solution, identifying the best MLP parameters in each iteration. Subsequently, a swarm of wolves reposition themselves according to the position updating strategy. Upon reaching the end of the iterations, the best MLP parameter is the optimal solution that contains the optimal bias and weight parameter of the MLP learning process. The proposed hybrid approach uses the optimal parameters to identify the potential intrusions and malicious traffic into the network.

The proposed hybrid approach is implemented using an open-source Python tool i.e., EvoloPy NN framework. It is a specialized framework designed for training MLPs using bio-inspired algorithms. The KDDCUP99 and UNSW datasets are used to evaluate the performance of the proposed approach in terms of accuracy, sensitivity, and specificity. The results of the proposed approach demonstrate the effectiveness in identifying potential malicious patterns compared to the state-of-the-art benchmark IDS schemes. Consequently, the security and reliability of the network increases. The main contributions of the paper are summed up as follows:

- We strategically employed the GWO algorithm into the feature vector to minimize the dimensionality of the feature vector. This accelerates the speed of the network intrusion detection system.
- We formulate the MLP architecture to use a specified number of hidden layers and nodes per layer to produce accurate results.
- We integrate the GWO algorithm into the MLP architecture to optimize the learning process by optimizing weights and biases to minimize the intrusion detection error rate.
- We strategically adjust the sensitive parameters to provide accurate predictions.

- We comprehensively evaluate the performance of the proposed IDS approach by utilizing network intrusion detection datasets i.e., KDDCUP19 and UNSW15.

The rest of the paper is organized as follows: Recent relevant literature are presented in [Section 2](#). [Section 3](#) contain the architecture of MLP and working operation of GWO as preliminaries. [Section 4](#) presents the proposed hybrid IDS methodology. The experimental simulation and results discussion are described in [Section 5](#). The conclusion and future direction are discussed in [Section 6](#).

2. Literature review

In this section, we present some of the recent research on the design of efficient network intrusion detection systems. Numerous methodologies are used to design an effective network IDS to protect the network from malicious attacks i.e., machine learning models, deep learning models, meta-heuristic algorithms, and hybrid models [15].

A hybrid Particle Swarm Optimization (PSO) and Fast Learning Network (FLN) based IDS technique is proposed by Ali et al. [16]. PSO-FLN used the KDD cup 99 dataset to test the model. The primary objective of the IDS technique is to optimize the neuron values like weights and biases in the hidden layer of the network architecture. PSO-FLN shows better performance in terms of accuracy compared to other similar benchmark algorithms. Due to a lack of training data, the algorithm may struggle to adapt to high traffic networks, thereby decreasing the detection rate for certain attack classes.

Fuzzy system based on multi-objective evolutionary algorithms for network IDS is proposed by Elhag et al. [17]. The focus of the algorithm is to select the optimal solution among multiple solutions according to the network features. The authors used a fuzzy system to handle the uncertainty and imprecision in the network traffic data. The multi-objective evolutionary algorithm is used to optimize a fuzzy rule base and explore the complex solution space to find an optimal trade-off among conflicting objectives to enhance the detection rate of the IDS technique. The authors used the KDD cup 99 dataset to test the model. The results show an efficient performance of the IDS technique in terms of false positive rate, computational cost, and detection rate.

Almomani et al. [18], proposed a meta-heuristic based feature selection technique for a network intrusion detection system. The technique employs four meta-heuristic algorithms: Firefly Algorithm (FA), Particle Swarm Optimization Algorithm (PSO), Gray Wolf Optimization algorithm (GWO), and Genetic Algorithm (GA) to enhance the capability of IDS in detecting unknown security threats. By employing multiple methods in combination with MI filter methods, the ratio of obtaining an optimal number of features for network IDS is increased, which aids in minimizing the processing time and avoids overfitting. The proposed model used a support vector machine, J48 machine learning classifier, and UNSW-NB15 dataset to evaluate the derived features in each iteration.

A hybrid PSO and Binary GWO (BGWO) technique is proposed by Alzubi et al. [19] to improve the detection accuracy and efficiency of network intrusion detection system. BGWO is used to select the features from the dataset to reduce the dimensionality and PSO is utilized to optimize the selected feature to escalate the detection accuracy of IDS.

Digital forensic classification using MLP neural network and Salp Swarm Algorithm (SSA) is proposed by Alazab et al. [20]. The primary objective is to enhance the classification accuracy in digital forensics by optimizing MLP parameters like bias and weights using SSA algorithm. SSA algorithm is used to enhance the learning process of MLP neural network by finding the optimal set of parameters. SSO algorithm globally explores the search space in search of potential optimal MLP parameters, thereby preventing local minima and slow convergence.

Naik et al. [21], proposed an intrusion detection system based on teaching-learning optimization (TLBO) and functional link neural network (FLANN). The IDS technique also incorporates mutation operations that can efficiently handle redundant parameters and prevent palindrome occurrence. By minimizing computational load, the IDS technique achieves a 96 % detection rate.

In [22], the authors used a hybrid Locust Swarm Optimization (LSO) and Feedforward Neural Network (FNN) to design an effective network intrusion detection system. The proposed network IDS technique obtains higher detection rate, reliability accuracy, and convergence speed because of high exploration of problem space which results in preventing local optima.

The authors in [23], used Genetic Programming (GP) and Support Vector Machine (SVM) to design an effective network intrusion detection system to enhance detection rate and accuracy. While maintaining detection accuracy, their proposed IDS technique achieves a higher detection rate for unknown malicious attacks including DoS attacks with 95 % detection rate.

In [24], the authors used the clustering technique to develop the IDS method. The method is a combination of a Decision Tree (DT) and cuttlefish algorithm. The cuttlefish optimization algorithm is used to improve the system performance by obtaining an optimal features set. The simulation results demonstrate the remarkable performance of the IDS method in terms of accuracy, detection rate, and minimum false positive rate.

In [11], the authors used a hybrid approach of the Harris Hawks Optimization (HHO) algorithm and MLP to design network IDS technique to detect known and uncommon malicious attacks. HHO is used to escalate the learning process of MLP by optimizing bias and weight parameters. KDD and UNSW 15 datasets are used to train the learning model. The simulation results demonstrate an efficient performance of the IDS method in achieving better accuracy, high detection rate, sensitivity level, and specificity.

In addition to the above literature, the researchers also employs other evolutionary algorithms combined with different type of ANN to design IDS methods like Moth Flame Optimization (MFO) based IDS method [25], Artificial Bee Colony [26], Genetic Algorithm [27], Grasshopper Optimization Algorithm (GOA) [28], BAT Optimization Algorithm (BOA) [29].

Overall, ANN combined with evolutionary algorithms demonstrates their effectiveness in designing IDS for various network architectures. However, most evolutionary algorithms may struggle to achieve higher accuracy and detection rates due to the unbalanced behavior of the algorithms, inefficient tuning of sensitive parameters, inefficient use of stochastic operators, and local stagnation. This creates an opportunity for the researchers to design an effective state-of-the-art solution for IDS. Moreover, the

simultaneous optimization of the number of feature sets and the parameters of the MLP neural network can generate a better classification model that can efficiently classify malicious and legitimate traffic.

GWO is chosen because of its strong capability of adaptive convergence, equilibrium behavior between the search stages, and alleviation of local optimal. The strong capability of GWO helps the optimizer to design an effective classification model to prevent local optima and minimize the error detection rate..

3. Preliminaries

In this section, we describe the basic architecture of multi-layer perceptron (MLP) neural network and the working operation of GWO.

3.1. MLP Neural Network Architecture

MLP neural network is a class of feedforward artificial neural network primarily used for classification and regression tasks [30]. It encompasses at least three layers of nodes or neurons i.e., one input layer, one or more hidden layers, and one output layer. Each layer's nodes or neurons are fully connected to every other node of the next layer. The strength of the connection is determined by the weight parameter. In addition, each layer's node (except the input layer) also has an associated bias which helps the detection model to fit the data better. The weight and bias parameters are learned during the training phase of the MLP neural network. MLP uses a backpropagation technique to propagate data from input nodes to the output nodes via hidden nodes.

The main functions are performed in the hidden layer of the MLP neural network, these are the summation function and activation function. Each node m receives signals from various nodes, combines them into aggregate sum using summation function, and then transforms into output result via activation function which are then further processed through computations within the network. The summation and activation functions are illustrated below:

$$S_m = \sum_{i=1}^N \omega_{im} \times F_n + B_m \quad (1)$$

Where ω_{im} indicate the connection's weight between the first node F_n and hidden node, while B_m represent the bias associated with the hidden node m .

$$Out_m = Func_{sig}(S_m) \quad (2)$$

Where Out_m indicate the output node m , while $Func_{sig}$ indicate sigmoid function and computed as below:

$$Func_{sig} = \frac{1}{1 + e^{-S_m}} \quad (3)$$

After aggregating signals from all the nodes in the hidden layer, the final output result of the node m is computed as follows:

$$AllS_m = \sum_{i=1}^m \omega_{im} \times Out_m + B_m \quad (4)$$

Now ω_{im} indicate the weight between the node m in the hidden layer and output node Out_m in the last layer, B_m indicate the bias associated with the node m in the output layer.

$$FinalOut_m = Func_{sig}(AllS_m) \quad (5)$$

Where $Func_{sig}$ is the same sigmoid function utilized in the computation of final output $FinalOut_m$.

3.2. Gray Wolf Optimization (GWO) algorithm

GWO belongs to the family of SI algorithms, inspired by the special leadership hierarchy and feeding maneuvers of gray wolves. Gray wolf's hierarchy is divided into four categories according to their dominance i.e., alphas, beta, delta, and omega. Alpha wolves are at the top of the hierarchy, also called leader or dominant wolves. They are the decision makers and their orders should be followed by the lower-level pack of wolves. Beta wolves are at the second level of the hierarchy, they play the role of an advisor and help the alphas in decision-making and other management activities. Omega wolves are at the lowest level of the hierarchy. These wolves are always follows the instructions and orders of dominant wolves. Delta wolves are called subordinate wolves, their responsibility is to keep an eye on the boundaries of the territory to protect the pack of wolves in case of emergency.

In addition to their dominant hierarchy, the hunting maneuver of the gray wolf is another interesting social behavior. According to the group feeding maneuver, gray wolves hunt their prey by chasing, encircling, and then attacking them. The leadership hierarchy and group hunting maneuver are mathematically modeled to perform optimization of MLP parameters during the learning process. This optimization utilizes exploration and exploitation phases, which are regulated by the coefficient vectors A and C . The vectors are calculated below:

$$\vec{A} = 2 \vec{l} * \vec{ran}_a - \vec{l} \quad (6)$$

$$\vec{C} = 2 \cdot \overrightarrow{ran}_c \quad (7)$$

Where $\vec{l}$ is the factor which is decreased linearly from 2 to 0 throughout the iterations, $\overrightarrow{ran}_a$ $\overrightarrow{ran}_c$ are the random numbers between 0 and 1.

Before attacking towards the prey, gray wolves (search agents) encircle prey by finding the possible location around the prey. This behavior is modeled using the following equations:

$$\vec{D} = \vec{C} * \vec{L}_p(ite) - \vec{L}_G(ite) \quad (8)$$

$$\vec{L}(ite+1) = \vec{L}_p(ite) - \vec{A} * \vec{D} \quad (9)$$

Where $\vec{L}_p(ite)$ and $\vec{L}_G(ite)$, indicate the prey location vector and the location vector of a search agent at current iteration ite respectively. $\vec{D}$ is the distance between the prey's location and the search agent's location.

According to the social hierarchy and dominance of the wolves, alpha (α) is considered the most optimal solution, beta (β) is the second optimal solution, and delta (δ) is the third optimal solution. The rest of the wolves update their location vector towards the location vector of alpha, beta, and delta search agents obtained so far. The following equations are used as a repositioning strategy:

$$\vec{D}_\alpha = \vec{C}_1 * \vec{L}_\alpha - \vec{L}_G, \vec{D}_\beta = \vec{C}_2 * \vec{L}_\beta - \vec{L}_G, \vec{D}_\delta = \vec{C}_3 * \vec{L}_\delta - \vec{L}_G \quad (10)$$

$$\vec{L}_1 = \vec{L}_\alpha - \vec{A}_1 * \vec{D}_\alpha, \vec{L}_2 = \vec{L}_\beta - \vec{A}_2 * \vec{D}_\beta, \vec{L}_3 = \vec{L}_\delta - \vec{A}_3 * \vec{D}_\delta, \quad (11)$$

$$\vec{L}(ite+1) = \frac{\vec{L}_1 + \vec{L}_2 + \vec{L}_3}{3} \quad (12)$$

3.2.1. Exploration search phase

In this phase, the search agents stochastically diverge from each other in search of potential optimal solutions. This is achieved when the coefficient vector $|\vec{A}|$ is greater than 1. Another variable that helps the optimizer in divergence is $\vec{C}$ that contain random

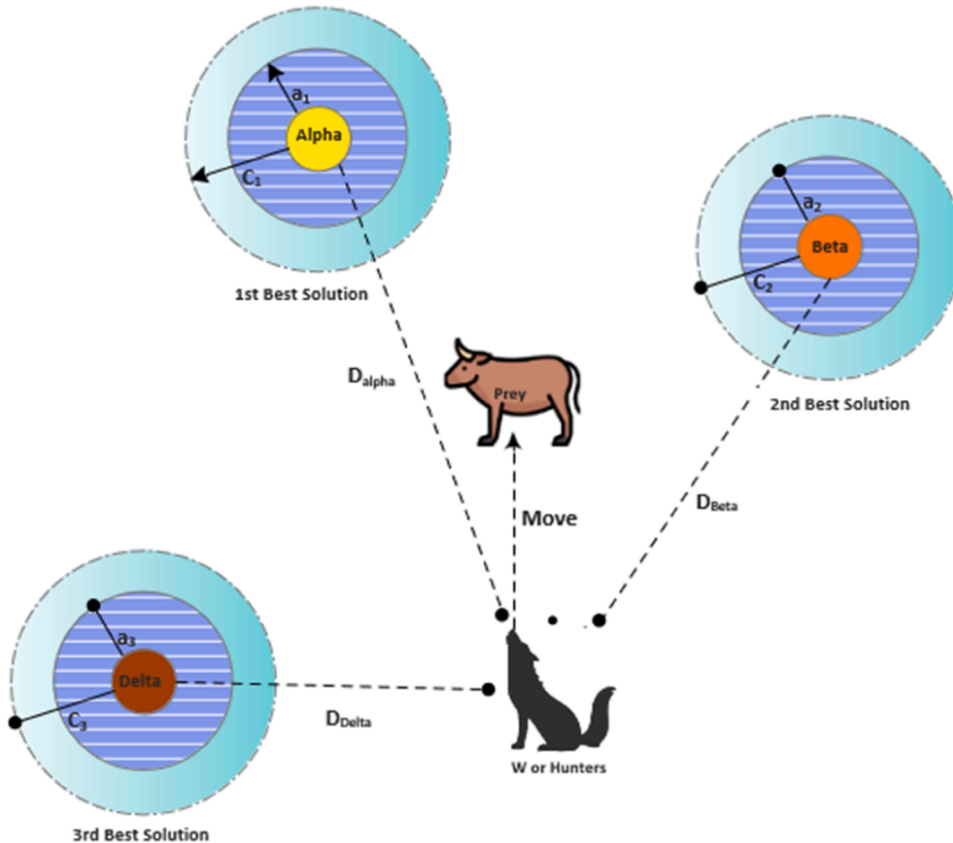


Fig. 1. Position Updating Mechanism in GWO.

values between 0 and 2. This variable emphasizes the optimizer to perform exploration over the course of iterations to prevent local stagnation.

3.2.2. Exploitation search phase

This is the attack phase, where search agents come closer to the prey during hunt. This is achieved by decreasing the value of linear factor $\vec{l}$. The value of the coefficient vector $|\vec{A}|$ is depends on $\vec{l}$ because, $|\vec{A}|$ contain random value between $[-2\vec{l}, 2\vec{l}]$. When the value of the coefficient vector $|\vec{A}|$ is less than 1, then the search agents reposition themselves between the location of best solution i.e., prey and current location of search agent. The position updating strategy is illustrated in Fig. 1.

4. Proposed hybrid IDS methodology

The proposed methodology is based on optimizing the MLP neural network parameters (weights and biases) using the GWO evolutionary algorithm. The primary objective is to reduce the error rate in detecting network intrusions. To this end, the proposed approach encompasses four stages i.e., 1) Data preprocessing 2) initialization 3) solution evaluation and 4) update position. Symbols used in equations and pseudo code are described in Table 1.

4.1. Data preprocessing

The data preprocessing stage involves two phases i.e., normalization and optimal feature selection. In the normalization phase, we reduce the network traffic values and make them normalized and standardized before the training. During normalization, all network data values are kept inside a certain range. This helps the algorithm in enhancing classification accuracy. The below equation is used to normalize the network traffic:

$$\sigma_V = \frac{\psi_V - L_B}{U_B - L_B} * (R_U - R_L) + R_L \quad (13)$$

Where σ_V, ψ_V indicate the normalized and non-normalized traffic value respectively. U_B, L_B represent the upper and lower bound of the network traffic respectively, R_U, R_L , indicate the limits of the normalized range. In case, the normalization range is $[0, 1]$, Eq. (13) will become:

$$\sigma_V = \frac{\psi_V - L_B}{U_B - L_B} \quad (14)$$

The next phase of data preprocessing is optimal feature selection which is carried out using the GWO algorithm. In this phase, the primary task of GWO is to reduce the dimensionality of the feature vector. To do this, we employed the GWO population to represent a feature vector as a candidate solution. A feature vector may contain a value of either 1 or 0. If it is 0, this means that the associated feature is not selected, and in the case of 1, it means the associated feature is selected. The below equation is used to represent a feature vector in the solution of GWO population.

$$GWO_k = [FV_1^1, FV_1^2, FV_1^3, FV_1^4, \dots, FV_1^D] \quad (15)$$

Where GWO_k, FV_1^i indicate the k^{th} solution and 1st feature of the i^{th} feature array respectively. By optimal feature selection, the proposed approach have fewest network traffic attributes which help them to minimum error rate in network intrusion detection. The below cost function is used to optimize feature vector.

$$C_F = a \times R_{Error} + b \times \frac{f_V}{F_V} \quad (16)$$

Table 1
List of Symbols.

Notations	Meaning
σ_V	Normalized traffic value
ψ_V	Non-normalized traffic value
γ_{Neu}	Number of neuron
F_{inp}	Length of the input data
R_{Error}	Intrusion detection error rate
$Func_{Sig}$	Sigmoid function
f_V	Selected feature vector
F_V	Set of all features vector
$w_1 b_1$	Weight and bias
Sol_{vec}	Solution vector
L_B	Lower bound
U_B	Upper bound
BestWeight	Best Weight

Where C_F is the cost value associated with each feature vector in the solution of GWO population. f_v , F_V indicate the selected feature and set of all features respectively. R_{Error} is the error rate of intrusion detection. Among the set of all feature vectors, the cost function aims to select a feature vector with minimum cost value having optimal feature attributes. The proposed hybrid IDS approach efficiently optimizes the feature vector while still maintaining an acceptable error detection rate. The optimized feature attributes are then used to train the MLP neural network.

4.2. Initialization

To initialize the search agents within the search space, we need to determine the number of neurons or nodes in the hidden layer of the MLP neural network because there is no standard way to determine their number. Therefore, a fixed architecture is used by the GWO with a defined number of nodes. The below equation is used to determine this number

$$\gamma_{Neu} = 2^* F_{inp} + 1 \quad (17)$$

Where γ_{Neu} , F_{inp} represent the number of neuron, input data respectively. Therefore, total connection parameter (weights and biases) can be determine using below equation.

$$n = F_{inp} \times \gamma_{Neu} + 2 \times \gamma_{Neu} + 1 \quad (18)$$

After determining the MLP training model, S search agents are randomly initialized as a potential candidate solution within the search space. Each search agent represents an optimized feature vector along with the associated weights and biases. The following equation is used to represent a solution vector of weights and biases.

$$Sol_{Vec} = \{Sol_1, Sol_2, Sol_3, \dots, Sol_n\} \quad (19)$$

$$Sol_{Vec}(i) = \{w_1 b_1, w_2 b_2, w_3 b_3, \dots, w_m b_k\} \quad (20)$$

Where $w_1 b_1$ indicate the weight and bias of 1st solution vector. Mapping of solution vector into MLP architecture is shown in Fig. 2.

4.3. Solution evaluation

Each search agent solution that encodes features vector along with the associated weights and biases is evaluated using the fitness function. The fitness function determines the quality of a solution. A solution with a low error rate represents a good solution and vice versa. The below fitness function is used to determine the goodness of a solution:

$$F_{Fitness} = \frac{1}{T} \sum_{j=1}^T [Y_j \log(\hat{y}_j) + (1 - Y_j) \log(1 - \hat{y}_j)] \quad (21)$$

Where Y_j , $\hat{y}_j$ indicate the actual output of the j^{th} training process and predicted output of j^{th} training instance of MLP neural

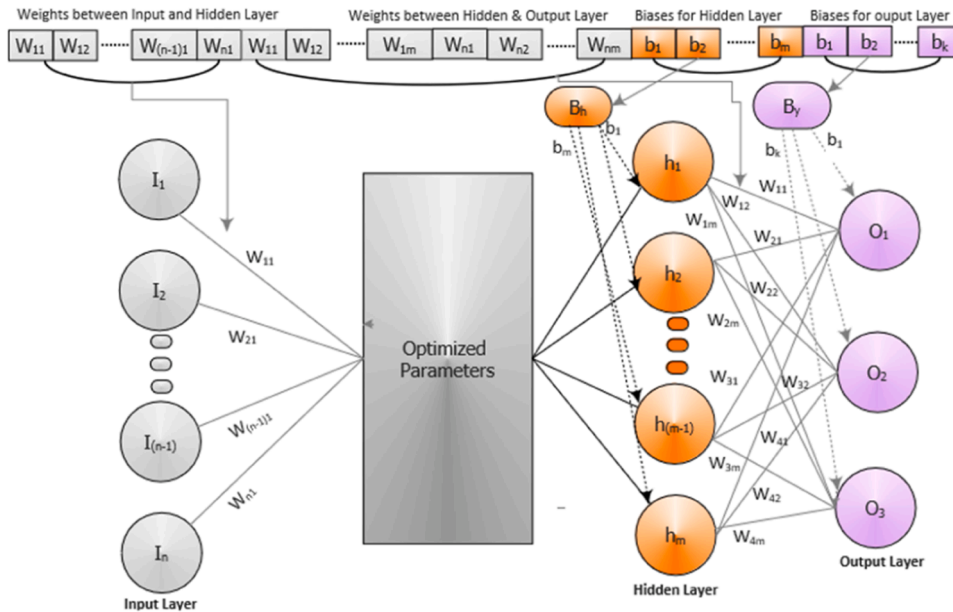


Fig. 2. Mapping of solution vector into MLP Architecture.

network with given weights and biases, respectively. A minimum fitness value indicates that the model is accurately predicting the output, resulting in better performance in terms of detecting malicious and legitimate traffic. Conversely, a higher fitness value indicates poor performance where the predication are far from the actual result, this is attributed to a high error rate in intrusion detections.

4.4. Update position

In this phase, the weights and biases of each feature vector are updated according to the position-updating strategy of the GWO algorithm. It is important to periodically update the weights and biases of each solution to come closer to the more optimal connection parameters i.e., weights and biases, and to minimize the error rate of IDS.

The connection parameters are updated according to the exploration and exploitation search strategy which is regulated by coefficient vectors A and C . The search agents employed an exploration search strategy in case of coefficient vector A is greater than 1. The search agents utilize an exploitation strategy when the coefficient vector A is <1 . Eq. (12) is used to update the connection parameters.

The optimization process stops when the algorithm reaches termination criteria i.e., maximum iteration, or when there is no improvement in the fitness value for a set of iterations. After the termination criteria, the optimal connection parameters i.e., weights and biases are then used to design the architecture of the MLP neural network.

Table 2

Pseudo code of Proposed GWOMLP-IDS.

Algorithm 1: Proposed Hybrid IDS algorithm

Input:

KDD and UNSW Datasets

Swarm size N and Number of maximum Iterations T

Output:

Optimal connection parameters i.e., weights and biases for MLP network architecture.

```

1. Begin
2. Initialize all the prerequisite parameters i.e., coefficient vector  $a$ ,  $A$ ,  $C$ .
3. Perform traffic normalization and optimization using Eq. 13 and 16 respectively
4. BestWeight =  $\infty$ , BestBias =  $\infty$ , BestSol(Loc) = {0,0}
5. Initialize search agents (GWO individual) within the search space
6. For (Iteration = 1  $\rightarrow$  MaxIteration) do {
7.   For (k = 1  $\rightarrow$  Swarm Size  $N$ ) do {
8.     i). Solvec(k) = { $w_1b_1, w_2b_2, w_3b_3 \dots \dots w_mb_m$ }
9.     ii). Evaluate the Solvec using Eq. 21
10.    Fitness =  $\frac{1}{T} \sum_{j=1}^T [Y_j \log(\hat{y}_j) + (1 - Y_j) \log(1 - \hat{y}_j)]$ 
11.    iv). if Fitness (Solvec(k) < BestWeight) do {
12.      i). BestWeight = Fitness (Solvec(k) }
13.    v). End if
14.    vi). if Fitness (Solvec(k) < BestBias) do {
15.      i). BestBias = Fitness (Solvec(k) }
16.    vii). End if
17.   End For
18.   Update coefficient vector  $A$ ,  $a$ ,  $C$ 
19.   For (j=1  $\rightarrow$  Number of search agents) do {
20.     i) update the solution vector using exploration and exploitation mechanism
21.     ii) if  $|A| < 1$  do {
22.       i). Use exploitation to update the solution vector
23.       ii). NewSolvec = UpdateSolution (BestWeight, BestBias)
24.     Else
25.       i). Use exploration to update the solution vector
26.       ii). NewSolvec = UpdateSolution (BestWeight, BestBias)
27.     End if
28.   End For
29.   Iteration++}
30. End For
31. Optimal weight = BestWeight
32. Optimal Bias = BestBias
33. Train the MLP network using optimal weights and biases
34. Evaluate the model against Intrusion detection

```

4.5. Pseudo code of proposed hybrid IDS approach

Table 2 presents the pseudo-code of the proposed hybrid IDS approach. The inputs of the algorithm include KDD and UNSW datasets, swarm size N , and a number of maximum iterations. Line # 1–2, initialize the prerequisite parameters like coefficient vectors. Line # 3, performs data preprocessing where network traffic values are normalized and optimized to enhance classification accuracy by maintaining a minimum error detection rate. In Lines # 4–5, we initialized the search agents within the search spaces and variables (e.g., best weight, best bias, and location of best solution) to their default values. Line # 7–16 is a loop where we encode random weights and biases in the feature vector of each search agent as candidate solutions to represent the population of GWO. In the same loop, we evaluate the candidate solution using the fitness function as given in Eq. (21). This determines the best feature vector containing the best weight and biases. Line # 18–26 is a loop, where we update the feature vector (weights and biases) according to the solution updating mechanism of GWO. Line # 29–32, assign the best

The operational procedure of the proposed IDS approach is illustrated in Fig. 3. The entire working operation consists of four phases. In the first phase, data is processed to normalized and standardize the network traffic values. The second phase involves initialization where candidate solution are randomly initialized within the search space. Each candidate solution represent the feature vector along with the corresponding weights and biases. The next step evaluate the candidate solution to determine the quality of each feature vector. The weights and biases associated with the feature vector is updated if it has best fitness value in the current iteration. The final phase, update the candidate solution according to the exploration and exploitation method of GWO algorithm.

5. Experimental setup and result discussion

The proposed hybrid approach is implemented using an open-source python tool i.e., EvoloPy NN framework on a LAPTOP with 16 GM RAM and core i7 2.56 GHz Intel processor. It is a specialized framework designed for training MLPs using bio-inspired algorithms. The KDD and UNSW datasets are used to investigate the proposed IDS approach. The datasets contain a collection of network traffic generated by the network nodes. KDD dataset has a total of 42 features and consists of malicious and legitimate network traffic. KDD dataset contains four types of malicious attacks i.e., DoS attack, Probing and surveillance attack, user-to-root attack, and remote-to-local attack. The UNSW dataset contains nine malicious attacks i.e. Backdoor, DoS, Shellcode, Exploits, Reconnaissance, Worms, Fuzzers, Generic, and Analysis. The distribution ratio of each type of attack and simulation parameters are given in Table 3 and Table 4 respectively.

5.1. IDS Performance Metrics

The performance of the proposed GWOMLP-IDS approach is analyzed using three performance metrics i.e., accuracy, specificity, and sensitivity. Accuracy indicates the overall correctness of the system by computing the ratio of accurate predictions of network traffic (including both legitimate and malicious traffic) to the total instances of network traffic. Sensitivity represents the proportion of

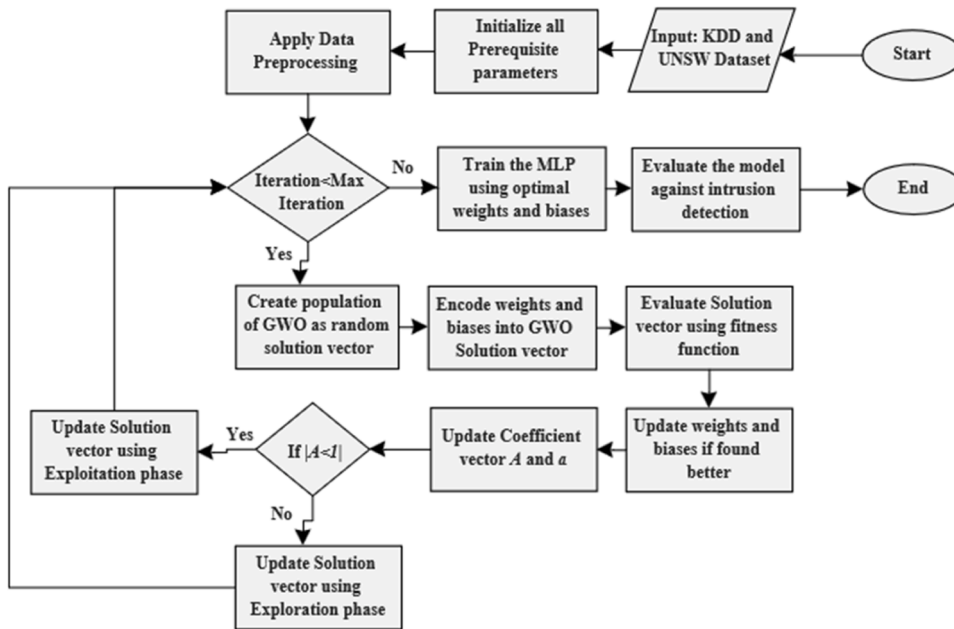


Fig. 3. Flowchart of Proposed GWOMLP-IDS method weight and best bias value to optimal weight and bias. Afterward, MLP neural network are trained using optimal values.

Table 3
Attack distribution ratio in KDD dataset.

Dataset	Attack Category	Instances	Ratio
KDD99	Normal	97,277	19.69 %
	DoS	391,458	79.24 %
	R2L	1126	0.22 %
	U2L	52	0.01 %
	Probe	4107	0.84 %
	Total	494,019	100 %

Table 4
Simulation Parameters.

Parameters	GWOMLP-IDS
Max Iterations	100
Swarm size	15–30
Hidden layers	2
Neuron per layer	5
Coefficient vector A	[2,0]
Coefficient vector C	[0, 2]
Dimension	2

correctly identified network traffic as intrusions, it is also called recall or true positive rates. Specificity indicates the proportion of correctly identified network traffic as legitimate traffic. The accuracy evolution measure of IDS is calculated below:

$$Acc_{IDS} = \frac{T_{Pos-ID} + T_{Neg-ID}}{T_{Pos-ID} + T_{Neg-ID} + F_{Pos-ID} + F_{Neg-ID}} \quad (22)$$

Where T_{Pos-ID} , T_{Neg-ID} represents true positive and true negative respectively, while F_{Pos-ID} , F_{Neg-ID} indicates false positive and false negative. The sensitivity and specificity performance metric of IDS scheme are calculated using Eq. (23) and Eq. (24) respectively.

$$Sen_{IDS} = \frac{T_{Pos-ID}}{T_{Pos-ID} + F_{Neg-ID}} \quad (23)$$

$$Spec_{IDS} = \frac{T_{Neg-ID}}{T_{Neg-ID} + F_{Pos-ID}} \quad (24)$$

5.2. Swarm size vs error rate prerequisite

This experimental framework analyzes the effect of swarm size on IDS error rate. The experiments are performed over 50 iterations with a varying swarm size ranging from 10 to 50. The results in Fig. 4 demonstrate that while increasing the population size, the error

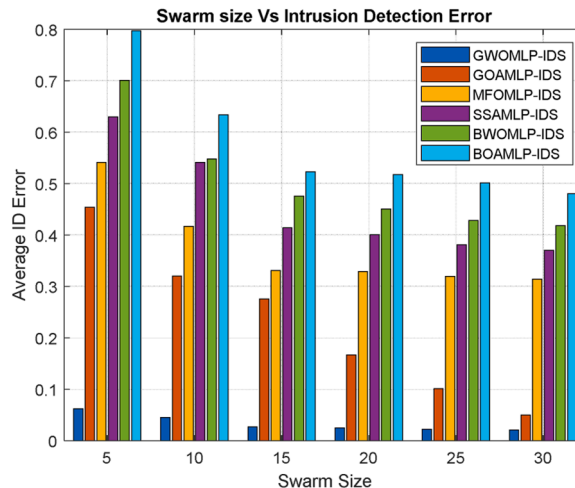


Fig. 4. Swarm size verses Intrusion Detection Error.

detection rate is linearly decreased. This implies that the proposed IDS scheme achieves a minimum error detection rate for the largest population size. Furthermore, irrespective of the population size, the stochastic connection

5.3. Performance measurement on KDD and UNSW datasets

In this simulation setup, we analyze the performance of the GWOMLP-IDS approach concerning the accuracy, sensitivity, and specificity of IDS. The evaluation of these metrics is carried out using KDD and UNSW datasets. The results in Fig. 5 demonstrate the efficient performance of the proposed hybrid IDS approach over the KDD dataset in terms of accuracy, true positive rate, and true negative rate compared to the learning methods without hybridization with meta-heuristic algorithms. This implies that combining MLP with GWO, significantly improves the network intrusion detection rate by achieving the highest accuracy (99.45 %), sensitivity (98.82 %), and specificity (99.29 %) level.

The next experiment is performed over the UNSW dataset to determine the efficiency of the proposed approach in detecting network intrusions. The outcomes displayed in Fig. 5, reveal that the proposed hybrid IDS approach yields efficient performance in terms of obtaining high accuracy, sensitivity, and specificity compared to other benchmark algorithms.

Fig. 6 illustrates the convergence curve of the proposed approach and other IDS approaches. The simulation outcomes reveal that the proposed approach exhibits stable and smooth convergence behavior towards achieving optimal connection parameters of the MLP neural network i.e., weights and biases. Moreover, the convergence speed of the proposed approach is faster towards obtaining optimal fitness value compared to other approaches.

The comparative analysis between the proposed IDS approach and other IDS algorithms over KDD and UNSW datasets unfolds in Table 5 and Table 6 respectively.

The convergence performance of the proposed algorithm concerning other approaches is shown in Table 7. The table has three main attributes, indicating the best fitness value, maximum fitness value, convergence speed, and standard deviation (S.D). S.D reflects the smoothness, stability, and consistency of the algorithm against multiple rounds. As per the numerical results, the proposed algorithm exhibits better convergence behavior in terms of S.D. Having a lower value of S.D, the proposed GWOMLP-IDS algorithm converges quickly and consistently with fewer variations towards minimum optimal fitness value. This implies that the proposed algorithm significantly enhances the error detection rate of network intrusions compared to the benchmark algorithms.

While measuring the performance of the IDS approaches, an important evaluation criterion is IDS speed which indicates the time taken by the algorithm to detect intrusions within the network. The proposed algorithm optimizes the features vector in the pre-processing stage to select an optimal feature attribute within the network traffic. Consequently, the hybrid IDS approach performs a training process on the optimized part of the network traffic. This results in a significant increase in the speed of detecting network intrusions compared to state-of-the-art algorithms. Fig. 7 illustrates the outcomes of the proposed algorithm in terms of IDS time. The results demonstrate the effectiveness of the proposed approach concerning intrusion detection speed.

Overall, we can safely infer that the proposed IDS approach efficiently handles the complexities of intrusion detection, surpassing alternative benchmark algorithms. This is attributed to the inherent characteristics of the GWO algorithm (e.g., stochastic operators, fine-tuning of sensitive parameters, adaptive convergence, and strong exploration and exploitation ability). These features enable the proposed approach to skillfully balance the multiple iterations between diversification and intensification search strategies. The strategic interplay of the position updating mechanism prevents local minima and provides a global optimal solution, resulting in a minimum error detection rate for the network intrusion detection system. Moreover, during the learning process, GWO continuously

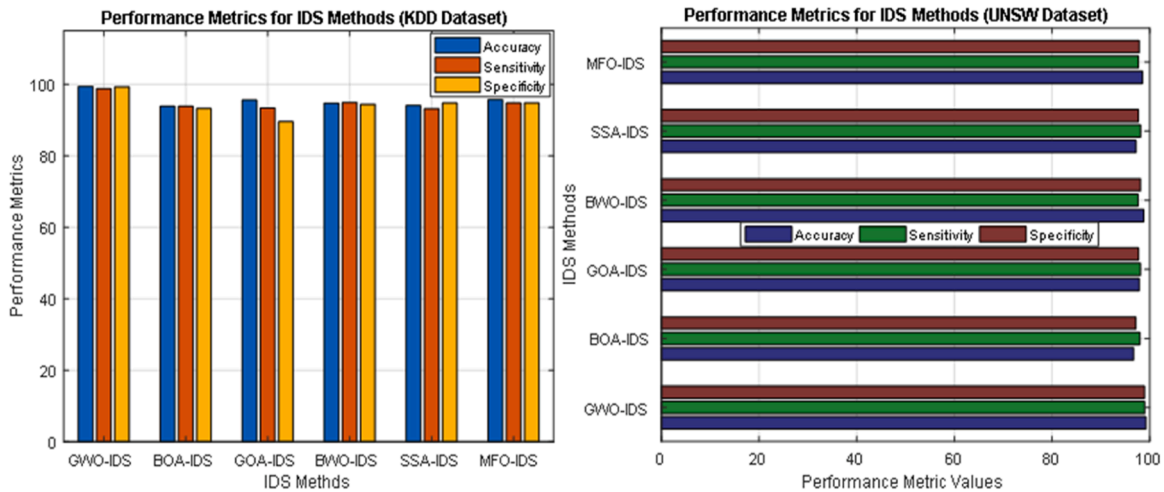


Fig. 5. Performance Metric values for IDS methods over KDD and UNSW datasets. The MLP neural network converge to the optimal weights and biases throughout iteration. The optimality of weights and biases is determined by considering the stabilization of the fitness function. Stabilization means when there is no or minimum improvement in the fitness value for a set of iterations.

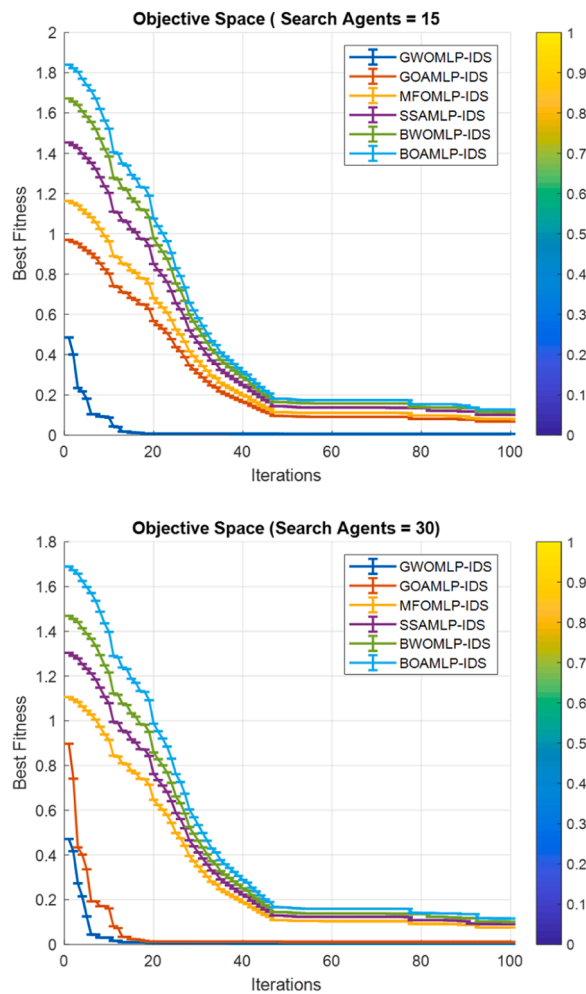


Fig. 6. Convergence behavior of IDS Methods against swarm size 15 and 30.

Table 5

Comparative analysis of Performance Metrics of IDS methods over KDD dataset.

IDS Method	ID-Time	Accuracy	Sensitivity	Specificity	Average Error Rate
GWOMLP-IDS	0.15	99.45	98.82	99.29	0.033
BOAMLP-IDS	1.5	93.92	93.89	93.24	0.227
GOAMLP-IDS	0.3	95.61	93.37	90.58	0.375
BWOMLP-IDS	1.2	94.76	94.96	94.44	0.456
SSAMLP-IDS	0.8	94.11	93.21	94.77	0.503
MFOMLP-IDS	0.4	95.75	94.78	94.82	0.575

Table 6

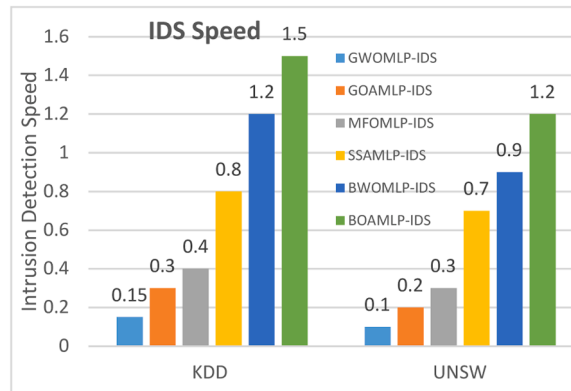
Comparative analysis of Performance Metrics of IDS methods over UNSW dataset.

IDS Method	ID-Time	Accuracy	Sensitivity	Specificity	Average Error Rate
GWOMLP-IDS	0.1	99.23	98.98	98.93	0.036
BOAMLP-IDS	1.2	96.71	97.98	97.14	0.210
GOAMLP-IDS	0.2	97.84	98.11	97.71	0.319
BWOMLP-IDS	0.9	98.78	97.61	98.11	0.409
SSAMLP-IDS	0.7	97.19	98.12	97.67	0.489
MFOMLP-IDS	0.3	98.52	97.67	97.84	0.505

Table 7

Convergence behavior of GWOMLP-IDS and other methods.

IDS Method	Swarm Size	Max Fitness	Best Fitness	S.D	Swarm Size	Max Fitness	Best Fitness	S.D
GWOMLP-IDS	15	0.485	0.0060	0.0720	30	0.4716	0.0038	0.0707
GOAMLP-IDS		0.969	0.0671	0.2837		0.8973	0.0111	0.1332
MFOMLP-IDS		1.163	0.0801	0.3406		1.1052	0.0764	0.3256
SSAMLP-IDS		1.454	0.1004	0.4253		1.3029	0.0897	0.3814
BWOMLP-IDS		1.672	0.1149	0.4895		1.4687	0.1014	0.4296
BOAMLP-IDS		1.839	0.1273	0.5385		1.6890	0.1160	0.4944

**Fig. 7.** ID Speed of IDS methods over KDD and UNSW dataset.

optimizes the parameters of MLP making the IDS model more flexible and adaptable to new network traffic. Based on new data traffic, the model parameters are continuously updated. This continuous updating enables the proposed algorithm to adapt to changes in attack patterns, enhancing its ability to respond to new cyber-attacks.

The computational complexity of GWO is relatively low compared to other evolutionary algorithms, making it more suitable for large-scale problems. Its convergence speed is high even in large search spaces that enable it to optimize extensive datasets or large network environments without significant increase in computational complexity and overhead.

6. Conclusion

In this study, we introduce a novel hybrid GWO and MLP based intrusion detection system, tailored for computer networks. The proposed hybrid IDS approach aims to optimize the connection parameters of the MLP neural network i.e., weights and biases to minimize error detection rate. The proposed approach is implemented using an open-source python tool i.e., EvoloPy NN framework, and investigated over KDD99 and UNSW15 datasets. The simulation outcomes validate the efficacy of our proposed IDS approach concerning accuracy, specificity, sensitivity, convergence behavior, and IDS time. This efficiency is attributed to the strategic interplay of exploration and exploitation phases within the GWO algorithm. Due to the inherent characteristics of GWO and MLP, (e.g., continuous training, low complexity, and continuous parameter optimization) the proposed IDS model is more flexible, scalable, and adaptable to new large network traffic. This enables the model to detect new cyber-attacks in large-scale networks.

Future work avenues include exploring other evolutionary algorithms such as Arithmetic Optimization Algorithm (AOA) and Cheetah Optimizer (CO), with advance neural network architecture like Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), could further improve the model's capability in detecting unknown cyber-attacks. This hybrid model can optimize various connection parameters like learning rate, dropout rate, number of hidden layers, number of neuron per layer, etc.

We can extend this work by utilizing deep learning models across various network infrastructures such as the Internet of Things (IoT), Flying Adhoc Network (FANET), and Fog Computing. This would enhance the model's real-world applicability and scalability.

Moreover, we can also employ Federated Learning (FL) in a distributed network environment to train the IDS model across distributed devices without centralizing the data. Each device trains the model locally on its data and sends only the model updates like optimized parameters (e.g., weights and biases, etc.) to a central server for aggregation. This allows the proposed model to handle massive amounts of data and detect intrusion in real time across different locations. Since only the model updates are shared with the central server, privacy is preserved.

CReditorship statement

Conceptualization, A.A. and M.A.; methodology, A.A and M.A. F.U.K; software, M.A, Y.Y.G and S.Y.S.; validation, T.J.A., A.Z and F. U.K.; formal analysis, Z.N, S.Y.S and A.Z.; investigation, Y.Y.G, T.J.A; data curation, Z.N and A.Z; writing—original draft preparation,

A.A, M.A, Y.Y.G; writing—review and editing, S.Y.S, F.U.K.; funding acquisition, T.J.A.

Declaration of competing interest

The authors declare that they have no known conflict of interest to be reported in this paper.

Acknowledgement

The Funding of research is provided by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R513), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data availability

Data will be made available on request.

References

- [1] Injadat M, et al. Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Trans Netw Serv Manag* 2020;18(2):1803–16.
- [2] Guo H, et al. A survey on space-air-ground-sea integrated network security in 6G. *IEEE Commun Surveys Tutorials* 2021;24(1):53–87.
- [3] Cai T, et al. ADAM: an adaptive DDoS attack mitigation scheme in software-defined cyber-physical system. *IEEE Trans Industr Inform* 2023.
- [4] Kumar S, Gupta S, Arora S. Research trends in network-based intrusion detection systems: a review. *IEEE Access* 2021;9:157761–79.
- [5] Javeed D, et al. An intelligent intrusion detection system for smart consumer electronics network. *IEEE Trans Consum Electron* 2023.
- [6] Heidari, A.A., et al., *Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks*. Nature-Inspired Optimizers: Theories, Literature Reviews and Applications, 2020: p. 23–46.
- [7] Sun W, et al. A survey of using swarm intelligence algorithms in IoT. *Sensors* 2020;20(5):1420.
- [8] Ali A, et al. Harris hawks optimization-based clustering algorithm for vehicular ad-hoc networks. *IEEE Trans Intell Transp Syst* 2023;24(6):5822–41.
- [9] Ali A, et al. Multi-Objective Harris Hawks Optimization Based Task Scheduling in Cloud-Fog Computing. *IEEE Internet Things J* 2024.
- [10] Haghnegahdar L, Wang Y. A whale optimization algorithm-trained artificial neural network for smart grid cyber intrusion detection. *Neural Comput Appl* 2020;32(13):9427–41.
- [11] Alazab M, et al. An effective networks intrusion detection approach based on hybrid Harris Hawks and multi-layer perceptron. *Egypt Inf J* 2024;25:100423.
- [12] Goyal A, Kumar C. GA-NIDS: a genetic algorithm based network intrusion detection system. Northwestern University; 2008. p. 210. URL, <http://www.cs.northwestern.edu/~ago>.
- [13] Mahboob AS, Moghaddam MRO. An anomaly-based intrusion detection system using butterfly optimization algorithm. In: 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS). IEEE; 2020.
- [14] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw* 2014;69:46–61.
- [15] Ahmad Z, et al. Network intrusion detection system: a systematic study of machine learning and deep learning approaches. *Trans Emerging Telecommun Technol* 2021;32(1):e4150.
- [16] Ali MH, et al. A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access* 2018;6:20255–61.
- [17] Elhag S, et al. A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems. *Soft comput* 2019;23:1321–36.
- [18] Almomani O. A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. *Symmetry (Basel)* 2020;12(6):1046.
- [19] Alzubi QM, et al. Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization. *Expert Syst Appl* 2022;204:117597.
- [20] Alazab M, et al. Digital forensics classification based on a hybrid neural network and the salp swarm algorithm. *Electronics (Basel)* 2022;11(12):1903.
- [21] Naik B, et al. Intelligent secure ecosystem based on metaheuristic and functional link neural network for edge of things. *IEEE Trans Industr Inform* 2019;16(3):1947–56.
- [22] Benmessahel I, et al. A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization. *Evol Intell* 2019;12:131–46.
- [23] Pozi MSM, et al. Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming. *Neural Process Lett* 2016;44:279–90.
- [24] Mohammadi S, et al. Cyber intrusion detection by combined feature selection algorithm. *J Inf Secur Appl* 2019;44:80–8.
- [25] Alazab M, et al. A new intrusion detection system based on Moth-Flame Optimizer algorithm. *Expert Syst Appl* 2022;210:118439.
- [26] Gulia N, et al. Intrusion Detection System Using the G-ABC with Deep Neural Network in Cloud Environment. *Sci Program* 2023;2023(1):7210034.
- [27] Zhao X, et al. Genetic algorithm based intrusion detection system for software-defined network architecture. In: 2020 IEEE International Conference on Progress in Informatics and Computing (PIC). IEEE; 2020.
- [28] Moghanian S, et al. GOAMLP: network intrusion detection with multilayer perceptron and grasshopper optimization algorithm. *IEEE Access* 2020;8:215202–13.
- [29] Ghanem WA, Jantan A. A new approach for intrusion detection system based on training multilayer perceptron by using enhanced Bat algorithm. *Neural Comput Appl* 2020;32(15):11665–98.
- [30] Sarker IH. Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective. *SN Comput Sci* 2021;2(3):154.