Academic Journal of International University of Erbil

Journal Homepage: https://ue.edu.krd/ojs/index.php/public

PRINT-ISSN: 2519-6928

Optimizing intrusion detection using intelligent feature selection with Gray wolf-based FOX algorithm

Hawkar Saeed Ezat1[1]*, Nawroz Fadhil Ahmed[2], Rizhin Nuree Othman[3], Zana Azeez Kakarash[4]

[1]Kurdistan Institution for Strategic Studies and Scientific Research, Sulaimani, KRG, Iraq

[2,4]Department of Information Technology, Kurdistan Technical Institute, Sulaimani, Kurdistan Region, Iraq

[3]Department of Medical Laboratory Science, Lebanese French University, Kurdistan Region, Iraq

*Corresponding Author: Hawkar Saeed Ezat

Received 13 Jan 2025; Accepted 01 Feb 2025; Available online 06 Mar 2025

ABSTRACT: Intrusion Detection Systems (IDS) are crucial in protecting computer networks against malicious activities. However, the performance of IDS can be improved by selecting the most relevant features from the vast amount of network traffic data. This article proposes an innovative approach to optimizing intrusion detection using intelligent feature selection with the FOX algorithm based on Grey Wolf optimization. In this study, intrusion detection is conducted using the KDDCup99 database. Then, processed features are selected. After preprocessing and preparing the dataset for data mining, they are fed into an MLP neural network. Each of the features and findings play a significant role in intrusion detection and prediction. In other words, not all features are equally valuable. Determining the value and role of each feature in intrusion detection is crucial. In this study, the value and role of each of these features are optimized and intrusion is identified by the Grey Wolf Optimization (GWO) algorithm. The proposed method's suitable accuracy compared to other classification algorithms used in this research such as Support Vector Machines and Decision Trees, demonstrates the efficiency and superiority of the proposed method.

Keywords: Intrusion Detection, Feature Selection, Gray Wolf Optimization, FOX Algorithm, Network Security, Cybersecurity

1. INTRODUCTION

Today, security is one of the key issues in modern computer systems. One of the significant challenges in these systems is intrusion detection, which refers to a set of activities aimed at compromising the integrity, reliability of the system, and unauthorized access to a specific resource. Therefore, intrusion detection systems are one of the techniques used to maintain security in computer networks. The increasing use of computer network services on one side and attacks on these networks on the other side have turned intrusion detection into a vital research area in securing these networks. Intrusion detection system is an effective security tool deployed in computer networks that examines and restricts user access using a set of predefined rules, based on experts' knowledge. In today's world, computer networks connected to the internet play a significant role in communications and information exchange. Among them, profit-driven individuals with access to valuable information of specific entities or others' information have attempted to gain unauthorized access to computer systems for intrusion, exerting pressure, or even disrupting the system's order. Intrusion detection systems are security systems that identify ongoing disruptions in the network. Intrusion detection systems (IDS) are responsible for identifying any unauthorized use of the system, misuse, or damage by internal and external users. Decision trees are one of the methods used for intrusion detection in a network [1-2]. Decision trees are not suitable in cases where the goal is learning an estimation function with continuous values. Moreover, in cases where there are many classifications and few training samples, the likelihood of error is high. Pruning decision trees is costly. In case of node overlap, the number of terminal nodes increases. If the tree is large, errors may accumulate from one level to another. Also, there is a possibility of creating incorrect relationships. Data mining techniques such as multi-layer perceptron neural networks (MLP) can be used for intrusion detection in networks. Among the challenges of predicting intrusion in computer

networks, increasing detection accuracy is crucial. Low accuracy in predicting network intrusions has irreparable consequences. In this study, to address this challenge, improving the performance of the neural network (MLP) with evolutionary optimization methods is proposed. The grey wolf optimizer (GWO) algorithm is one of the newest evolutionary optimization methods [3-4].

This algorithm is designed and implemented based on simulating the social behavior and hunting behavior of gray wolves [5]. This approach has been compared with evolutionary algorithms such as Particle Swarm Optimization (PSO). Both quantitative and qualitative results show that the Gray Wolf Optimization (GWO) algorithm can provide better results for some problems compared to the algorithms being compared [6]. The GWO algorithm is a new approach that has the potential to increase accuracy and reduce errors in prediction with fewer computations and less time complexity compared to similar works. Due to the existence of various parameters and features, correct intrusion detection in computer networks is challenging [7]. Therefore, in this research, by improving the performance of Multilayer Perceptron neural network (MLP) with the GWO algorithm, the accuracy of intrusion detection and classification for predicting intrusions in computer networks is enhanced. The dataset (KDDCup99) was examined in this study, and processed variables were selected [8].

One of the data mining techniques like Multi-Layer Perceptron (MLP) can be used for intrusion detection in networks. One of the challenges of predicting network intrusion is improving detection accuracy. Low accuracy in predicting network intrusions has irrecoverable losses [9-10]. In this study, to address this challenge, improving the performance of Multi-Layer Perceptron (MLP) using evolutionary optimization methods is proposed. The Grey Wolf Optimizer (GWO) algorithm is one of the newest evolutionary optimization methods. The optimization process in the GWO algorithm is based on a powerful process with random guidance [11]. This method is based on gradual evolutionary theory. Due to the presence of different parameters and features, accurate intrusion detection in computer networks is difficult. Therefore, in this study, by using the GWO algorithm to optimize the parameters of Multi-Layer Perceptron (MLP), detection accuracy and classification for predicting network intrusions in computer networks are improved [12].

This research uses one of the most common datasets for evaluating intrusion prediction systems in computer networks. The KDDCup99 dataset consists of 4898431 records or connection vectors, each with 41 features. These features are divided into four categories: basic features, content features, time traffic features, and host traffic features. Connections are classified into two types: malicious and normal. The test dataset also includes 311027 records. In this research, the dataset is first examined and processed variables are selected. After preprocessing and preparing the dataset for data mining, network intrusion is predicted using the GWO algorithm to optimize the parameters of the Multilayer Perceptron (MLP) [13-15]. Extensive research has been conducted in the field of network intrusion detection. Various simulations and tools have been used in these studies based on the proposed approach. Some of the most important simulations include Matlab, Nime, Klmentine, and RapidMiner. These simulations have been used in many research works related to predicting network intrusion detection [12]. Given Matlab's software comprehensiveness, intrusion detection data will be described, simulated, and analyzed using Matlab 2016 software. In this research, feature extraction and selection will be performed on the dataset (KDDCup99). To compare the proposed model with other methods, metrics such as accuracy, sensitivity, feature, accuracy, and measurement size based on the following relationships will be used.

$$\text{Accuracy} = (TP + TN) / \text{All} \quad (1)$$

$$\text{Sensitivity} = TP / (TP + FN) \quad (2)$$

$$\text{Specificity} = TN / (FP + TN) \quad (3)$$

TP: The number of diagnoses correctly identified as positive.

TN: The number of diagnoses correctly identified as negative.

FP: The number of diagnoses incorrectly identified as positive.

FN: The number of diagnoses incorrectly identified as negative.

The reset of the paper is structured in the following manner: Section 2 related works relevant to this study, while Section 3 outlines the research Methodology employed. Section 4 proposed method, followed by a result the proposed method in Section 5. The final section contains Recommendations and the Conclusion.

2. Related works:

Several studies have explored the use of machine learning (ML) models for the diagnosis of heart disease, showcasing various techniques, datasets, and evaluation metrics. Traditional ML methods, such as decision trees, support vector machines, and random forests, have been widely applied to classify heart disease based on patient data, including demographic information, clinical tests, and medical histories. For instance, [16] demonstrated the effectiveness of decision trees in heart disease prediction, highlighting their interpretability but noting limitations in terms of accuracy when compared to more complex models. Other studies have focused on using neural networks for heart disease diagnosis, with models like deep learning achieving superior accuracy at the cost of interpretability [17-18]. These studies

often emphasize the trade-off between accuracy and model complexity. Furthermore, research [19] has shown that while deep learning models outperform traditional methods in terms of diagnostic accuracy, they face challenges in terms of training time and the need for large labeled datasets.

Moreover, research on ensemble learning approaches has shown promising results in heart disease classification. [20] Tested several ensemble methods, including boosting and bagging algorithms, and found that ensemble models significantly outperformed single classifiers in terms of accuracy and generalization. These models have proven to be particularly beneficial in handling noisy data, a common issue in medical datasets. Despite this progress, challenges remain in heart disease prediction, especially with imbalanced datasets, where the majority of cases are negative, leading to biased predictions. Techniques like oversampling and synthetic data generation have been explored to mitigate this issue [21]. Additionally, [22] demonstrated the success of using generative adversarial networks (GANs) for augmenting small datasets, thereby improving the overall performance of predictive models.

Recent advancements in hybrid models, such as combining convolutional neural networks (CNNs) with traditional ML models, have shown potential in enhancing both diagnostic accuracy and efficiency [23]. These hybrid approaches aim to leverage the strengths of multiple algorithms, addressing the complexity of heart disease data and improving prediction rates. Hybrid models that combine unsupervised learning with supervised techniques have also gained attention due to their ability to extract important features from raw medical data without extensive preprocessing [24]. Despite these innovations, the need for real-time prediction systems and model explain ability remains a significant challenge in the clinical adoption of machine learning for heart disease diagnosis. As such, there is an ongoing need for research that can bridge these gaps, ensuring that machine learning models are not only accurate but also practical for use in medical settings [25].

Furthermore, the integration of patient-specific factors, such as genetics and lifestyle data, into machine learning models has recently become a point of interest. Research [26] has shown that personalized models, which consider individual patient histories and genetic predispositions, could significantly improve the accuracy of heart disease diagnoses. These personalized approaches are expected to be crucial in providing more tailored and effective treatments for patients, highlighting the future direction of ML in healthcare [27].

3. Research Methodology

Extensive research has been done on the importance of infiltration in organizations. Given that the issue of intrusion detection entails many features and due to the importance of information and organizations requiring high accuracy, our suggestion is to use data mining methods and techniques.

Among the available neural networks, we chose the Multilayer Perceptron neural network (MLP) to solve our problem of high-accuracy intrusion detection. This network is inspired by the human brain. As you know, the human brain is the most complex system and no supercomputer can compete with this powerful system. For input data to our neural network, we used the dataset (KDD cup99) which is a standard dataset containing intrusion features and classes, and for training this neural network, we used the gray wolf optimization algorithm which is an inspired metaheuristic algorithm from the life and hunting of gray wolves. The operation should be performed on the data, which will be explained further, and the algorithm structure for predicting intrusion detection in the network will also be demonstrated.

3.1 Phases of predicting intrusion detection on the network

Each of the stages of data collection, preprocessing, dividing the dataset into training and testing sets, feature selection, model generation, model optimization, and classification in predicting intrusion detection on the network are shown in Figure (1).

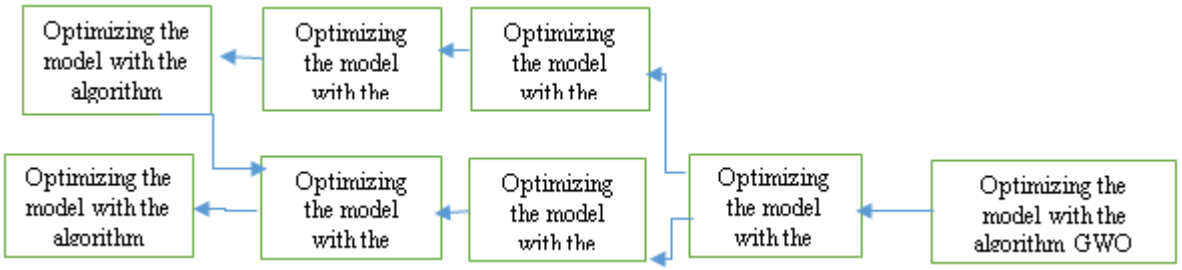

Figure 1: Stages of the algorithm for predicting intrusion detection on the network

3.2 Implementation

To run the proposed method, we used the Weka software with the following parameters: a neural network with 42 neurons in the input layer (for the number of features and intrusion detection parameters), three hidden layers, and one output neuron. The network was designed to predict intrusion. The MLP network was used for prediction, and different threshold functions were evaluated to find its optimal state. (4) Logistic sigmoid function

$$Y_i = \frac{1}{1+exp(X)_j} \qquad (4)$$

Tangent sigmoid function

$$Y_i = \frac{2}{1+exp(-2X_j))-1} \qquad (5)$$

The Grey Wolf algorithm was used for training, and the mean square error of this algorithm in training the neural network is much better than similar algorithms such as GA, PSO.

3.3 Evaluation Tools:

Bayes methods, neural networks, support vector machines, and decision tree algorithms are used for classification. Genetic algorithms, artificial bee colony clustering, grey wolf optimizer, and GBC are used for optimizing classification results.

The (K-fold) method is a common technique for evaluating the performance of classifiers. This algorithm receives a set of m training data and follows the following steps after each run:

1) Shuffles the training data randomly.

2) Divides the training data into K parts (where K is approximately K / m data).

3) Trains classifier K each time using one of the parts as the test set and the rest as the training set, then calculates the number of misclassified samples (ni) at each stage.

4) Returns the classifier error rate.

5) To obtain an accurate estimate of the classifier's accuracy, cross-validation K-fold is run several times, and decisions are made based on the average results.

In short, training, providing feedback to the algorithm, adjusting the predictive power of the classifier, and testing are the processes for determining the actual classification accuracy generated by the algorithm. During the experiment, data that has never been part of the training is classified. Usually, after each training stage, a validation process is performed to help determine the classification.

In this study, the data is first divided into approximately 10 equal parts using the K-Fold cross-validation method (K=10) and then the neural network is trained 10 times. Each time 9 parts are used for training and 1 part for testing, and the error is calculated at each stage.

4. Proposed Method:

4.1 Intrusion Detection Data Collection:

In this study, available data from the KDD Cup 1999 database has been selected for description, simulation, and analysis. The task of a self-learning intrusion detection system is to create a probabilistic model (for example, classification) that can distinguish between malicious or attack connections and normal and good connections. The DARPA Intrusion Detection Evaluation Program 1998 was prepared and managed by the MIT Lincoln Laboratory. The goal was to assess and evaluate intrusion detection systems. A standard set of data, including a wide range of simulated intrusions in a military environment, was examined. The intrusion detection dataset (KDD 1999) has been used in this research. Tables 1 and 2 show a complete list of records for a connection.

Table 1: Traffic features computed using a two-second time window

Description	type	The name of the feature
The number of calls to the same host in the current connection in the last two minutes	continuous	count
Note: The following features refer to these same-host connections.	continuous	

% of connections that have ``SYN'' errors	continuous	serror_rate
% of connections that have ``REJ'' errors	continuous	rerror_rate
% of connections to the same service	continuous	same_srv_rate
% of connections to different services.	continuous	diff_srv_rate
number of connections to the same service as the current connection in the past two seconds *Note: The following features refer to these same-service connections.*	continuous	srv_count
% of connections that have ``SYN'' errors	continuous	srv_serror_rate
% of connections that have ``REJ'' errors	continuous	srv_rerror_rate
% of connections to different hosts	continuous	srv_diff_host_rate

Table 2: Content features of a suggested connection

Description	type	The name of the feature
Number of hot indicators.	continuous	hot
Number of failed login attempts.	continuous	num_failed_logins
If successfully logged in, the value is 0; otherwise 1.	discrete	logged_in
Number of precarious conditions.	continuous	num_compromised
If root shell obtained, 0; otherwise 1.	discrete	root_shell
If the command (su root) is used, 0; otherwise 1.	discrete	su_attempted
Number of root accesses.	continuous	num_root
Number of file creation operations.	continuous	num_file_creations
Number of file control accesses.	continuous	num_shells
Number of external commands in an ftp session.	continuous	num_access_files
If the login belongs to the "hot" list, 0; otherwise 1.	continuous	num_outbound_cmds
1 if the login is a "guest" login; 0 otherwise	discrete	is_hot_login
If the login is a guest, 0; otherwise 1	discrete	is_guest_login

Table 3: Basic features of a unique TCP connection

Description	type	The name of the feature
Duration of connection (in seconds)	continuous	duration
Protocol type such as (TCP, UDP) etc.	discrete	protocol_type
Network service at the destination, for example (http, telnet) etc.	discrete	service
Number of bytes of data from source to destination	continuous	src_bytes
Number of bytes of data from destination to source	continuous	dst_bytes
Normal or error state of a connection	discrete	flag
If the connection is from a similar host, then port 0; otherwise port 1	discrete	land
Number of incorrect fragments	continuous	wrong_fragment
Number of urgent packets	conti	urgent

Table 4: Weighting features

Weighted value	Weight	Normal value	Initial value	Feature
0.21	0.3	0.71	0.48	Feature1
0.44	0.8	0.55	0.53	Feature1
--	--	--	--	--
0.63	0.7	0.9	0.33	Feature41

4.2 MLP Neural Network

A neural network is used to detect intrusion through a learning process to predict intrusion. This method identifies relationships and patterns between data using processors called neurons. The neural network approach provides a map between the input space (input layer) and the desired space (output layer). The input layer receives the data and sends it

to the hidden layers. Information is processed in the hidden layers and made available to the output layer. Each network learns by receiving training samples. Learning is done through a training process. Neural network learning occurs when the difference between the predicted values and the actual information is within an acceptable range. A trained neural network is used for prediction with a new set of data. Key features of artificial neural networks include pattern learning ability, high processing speed, ability to generalize knowledge after learning, flexibility against unexpected errors, and minimal disruption in case of failure.

4.3 Creating an Intrusion Detection Model Network by Gray Wolf Algorithm:

4.3.1 Mathematical Models of Gray Wolf Algorithm:

In this section, mathematical models related to the social hierarchy, tracking, encircling, and attacking prey in the Gray Wolf Algorithm are presented.

Social Hierarchy:

When designing the Gray Wolf Algorithm, to mathematically model the social hierarchy of wolves, we consider the best solution as alpha. Therefore, among the best solutions, we label the second and third as beta and delta. The remaining candidate solutions are considered omega. In the Gray Wolf Algorithm, the hunting (optimization) process is guided by alpha, beta, and delta. Omega wolves follow these three groups.

Encircling the Prey:

As mentioned above, gray wolves encircle the prey during hunting. To mathematically model this encircling behavior, equations (6) and (7) are presented:

$$\overrightarrow{D} = \left|\overrightarrow{C_1} . \overrightarrow{X_p}(t) - \overrightarrow{X}(t)\right| \quad (6)$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} . \overrightarrow{D} \quad (7)$$

In equations (6) and (7), t represents the current iteration. $\vec{A}$ And $\vec{C}$ represent coefficient vectors, $\overrightarrow{X_p}$ represents the vector of prey position, and $\overrightarrow{X}$ represents the vector of a gray wolf's position. Vectors $\vec{A}$ and $\vec{C}$ are calculated using equations (8) and (9):

$$\vec{A} = 2\vec{A} . \overrightarrow{r_1} - \vec{a} \quad (8)$$

$$\vec{C} = 2. \overrightarrow{r_2} \quad (9)$$

In equations (8) and (9), $\vec{a}$ linearly decreases from 2 to 0 during the iteration. . $\overrightarrow{r_1}$ and . $\overrightarrow{r_2}$ are random vectors in the range [1, 0]. To see the results of the equations, equations (1) and (2) along with a two-dimensional location vector and some probable neighbors in figure (2 a) are shown. As seen in the figure, a gray wolf located at (X, Y) can change its position based on the prey's position (X*, Y*). Different locations around the best agent can be obtained based on its current position and adjusting and changing the values of vectors A and C. For example, the location (X*-X, Y*) can be calculated by assigning $\vec{C} = (1 .1)$. $\vec{A} = (1 . 0)$. The updated probable locations related to a gray wolf in three-dimensional space in figure (2-b) are shown. It should be noted that random vectors r1 and r2 allow wolves to access any position between the points shown in (figure 2). Therefore, a gray wolf can change its position within the space that encompasses the prey randomly and by using equations (10) and (11). This concept can be extended to a multidimensional search space. In this case, gray wolves move around the best solution obtained in dimensions larger than the cube dimensions.

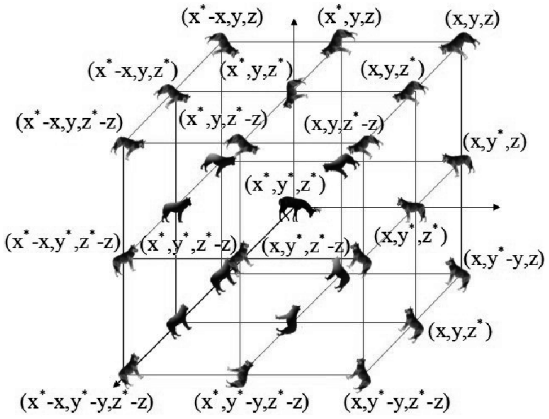

Figure 2: Two-dimensional and three-dimensional spatial vectors and their next probable positions

Hunting:

Gray wolves have the ability to detect the location of prey and trap it. The hunting process is usually guided by the alpha. Beta and delta wolves may also participate in the hunt at times. Unfortunately, in an abstract search space, there is no idea about the optimal position (prey). In order to mathematically simulate the hunting behavior of gray wolves, we assume that alpha (the best existing solution), beta, and delta have better knowledge about the potential location of the prey. Therefore, we save three of the best solutions obtained and force other search agents (including omega wolves) to update their position based on the position of the best search agents. This is done by equations (10), (11), and (12).

$$\overrightarrow{D_a} = \left|\overrightarrow{C_1} . \overrightarrow{X_a} - \overrightarrow{X}\right| \overrightarrow{D_\beta} = \left|\overrightarrow{C_2} . \overrightarrow{X_\beta} - \overrightarrow{X}\right| \overrightarrow{D_\delta} = \left| \overrightarrow{C_\varepsilon} . \overrightarrow{X_\delta} - \overrightarrow{X} \right| \quad (10)$$

$$\overrightarrow{X_a} = \overrightarrow{X_a} . \overrightarrow{A_1} . (D_a) \; \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} . \left(\overrightarrow{D_\beta}\right) . \overrightarrow{X_\varepsilon} = \overrightarrow{X_\delta} - \overrightarrow{A_\varepsilon} \left(\overrightarrow{D_\delta}\right) \quad (11)$$

$$\overrightarrow{X}\left(\vec{t} + 1\right) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{\varepsilon} \quad (12)$$

Figure 3 shows how to update the position of a search agent in a two-dimensional search space based on the positions of alpha, beta, and delta. It can be observed that the final position obtained is random and located inside a circle defined based on the positions of alpha, beta, and delta. In other words, alpha, beta, and delta estimate the hunting position, and the other wolves update their positions randomly in the hunting area.

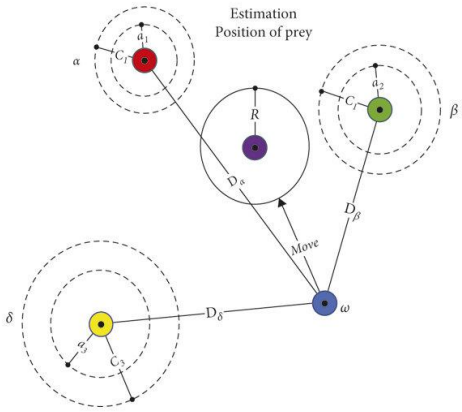

Figure 3: Position update in Gray Wolf Algorithm

4.3.2 Attack on prey (exploitation)

As mentioned above, gray wolves attack and end the hunt when it stops. To mathematically model the approach to the prey, we decrease the value of $\overrightarrow{a}$. It should be noted that the range of oscillation $\overrightarrow{A}$ is also reduced by $\overrightarrow{a}$. In other words, $\overrightarrow{A}$ is a random value in the interval [-2a, 2a], while a decreases from 2 to 0 during iterations. When random values of $\overrightarrow{A}$ are within [-1, 1], the next position of a search agent can be in any position between its current position and the prey

position. The value |A| < 1 forces wolves to attack towards the prey. With the operators presented so far, the Gray Wolf Algorithm allows its search agents to update their positions based on alpha, beta, and delta positions and attack the prey. However, the Gray Wolf Algorithm may be prone to trapping in local minima, so there is a need for operators to prevent getting stuck in local minima. Although the proposed trapping mechanism somewhat resembles the identification mechanism, the Gray Wolf Algorithm requires more operators to create the exploration property.

4.3.3 Search for prey (identification)

Gray wolves mainly engage in the search process based on the alpha, beta, and delta positions. They distance themselves from each other for hunting and approach each other for attacking, cooperating in the process. To model this divergence mathematically, vector $\overrightarrow{A}$ with a random value greater than 1 or less than -1 is used to compel the search agent to diverge and distance from the prey. This pattern demonstrates the identification process and allows the Gray Wolf Algorithm to perform the search operation globally. The value of |A|>1 forces the wolves to diverge from the prey and find a more suitable hunt. Another component of the Gray Wolf Algorithm that affects the identification process is the value of C. As seen in equation (10), vector C takes random values in the range [2 and 0]. This component provides random weights for hunting to intensify (C>1) or weaken (C<1) the impact of prey position on distance in equation (6). This component also aids the Gray Wolf Algorithm in showing more randomness during optimization, leading to better exploration and avoiding local minima traps. It is worth mentioning that C does not decrease linearly with A. However, random values are needed from C to execute the identification process not only in the initial iteration but also in the final iteration. This component is particularly useful in preventing local minima, especially in the final iteration. Vector C can also be considered as an obstacle effect that prevents wolves from approaching the prey in nature at an appropriate speed. In general, natural obstacles appear in the wolves' hunting path, impeding their approach to the prey at a suitable speed. This precisely illustrates the effect of vector C. Depending on a wolf's position, vector C can give a random weight to the prey to make it harder or easier for the wolves to reach it. In summary, the Gray Wolf Algorithm begins the search process by creating a random population of gray wolves (candidate solutions). Throughout the iteration period, alpha, beta, and delta wolves estimate the probable hunting positions. Each candidate solution updates its distance from the prey. The parameter a decreases from 2 to zero to enhance the identification and attack process. When $|\overrightarrow{A}|$>1, the candidate solutions diverge, and when $|\overrightarrow{A}|$<1, they converge towards the prey. The pseudo-code of the Gray Wolf Algorithm is presented in Figure 4.

Initialize the grey wolf population Xi (i = 1, 2,..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
Xα=the best search agent
Xβ=the second best search agent
Xδ=the third best search agent
while (t < Max number of iterations)
for each search agent
Update the position of the current
search agent by equation (7)
end for
Update a, A, and C
Calculate the fitness of all search agents
Update Xα, Xβ, and Xδ
t=t+1
end while
return Xα

Figure 4: Pseudocode of Gray Wolf Algorithm

4.4 Building the final model using Gray Wolf Algorithm and Neural Network

In this study, a multilayer perceptron (MLP) is used to predict network intrusion. The network consists of an input layer, one or more hidden layers, and an output layer. The dataset related to network intrusion is sent for processing to the input

layer. Training this network usually involves the Backpropagation (BP) algorithm. During the training of the MLP network using the Backpropagation (BP) learning algorithm, computations are first performed from the network input to its output, and then the calculated error values are propagated to the previous layers. Initially, the calculation of the output layer to layer is performed, and the output of each layer will be the input to the next layer. In the backpropagation phase, the output layers are adjusted as each neuron in the output layer has a desired value, and the weights can be adjusted using these values and regularization rules. In this study, the value and role of each feature are precisely determined using a neural network, and intrusion is identified. The proposed stages of training the neural network algorithm are as follows:

Stage one: Assign random weight matrices to each input value.

Stage two: Select appropriate input and output vectors.

Stage three: Calculate the output of neurons in each layer and consequently calculate the output of neurons in the output layer.

Stage four: Regularize the weights to the previous layers using the backpropagation error method.

Stage five: Evaluate the trained network's performance.

The overall structure of the neural network used is shown in Figure 5. Figure 6 illustrates the flowchart of the proposed approach. In this study, exploratory algorithms are used to adjust the dataset weights in network learning. The cost function for exploratory algorithms is defined as follows:

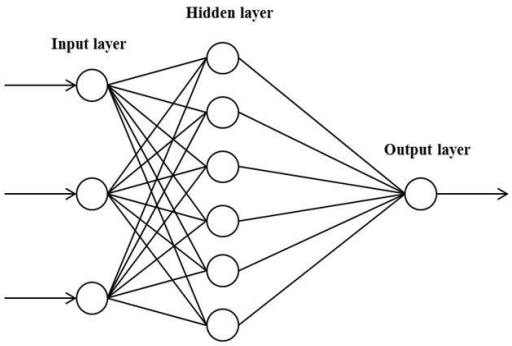

Figure 5: Structure of Artificial Neural Network

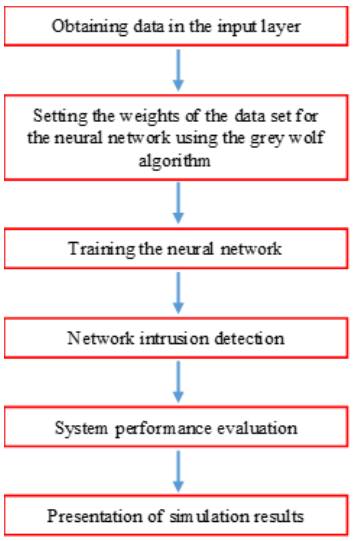

Figure 6: Proposed Approach Flowchart

4.5 Proposed Method for Improving Gray Wolf Algorithm

The most important factor that controls the efficiency and accuracy of an optimization algorithm is the trade-off between exploration and exploitation. Exploration refers to the algorithm's ability to search different regions of the search space to find suitable optima. On the other hand, exploitation is the ability to focus the search within a specific range to find the desired solution precisely. A good optimization algorithm balances these two competing objectives. In each iteration or in a complementary version, efforts are made to improve the performance of the method by controlling these two parameters. Experience shows that in the initial iterations, high exploration power is needed, gradually shifting towards more prominent exploitation power. This means that in the initial iterations, the algorithm explores diverse spaces and in later iterations, it searches the discovered areas more accurately.
To increase the efficiency and accuracy of the Grey Wolf Algorithm in reaching optimal values, at each stage of the algorithm's execution, the positions of a set of wolves change. This is done by averaging over a specific set of wolves. In other words, the exploration metric is well-adjusted to increase the optimization accuracy. In each stage of the algorithm's execution, solutions far from the target (wolves far from the target) are brought closer to the target through averaging to improve the algorithm's accuracy.

The number of wolves or agents is considered to be 25. The Grey Wolf Algorithm starts by creating a random population of grey wolves (candidate solutions). After initializing the parameters A, a, c randomly, the fit of each agent is calculated, and then agents are assigned to one of the alpha, beta, or delta categories based on their fitness. After determining the category of agents, in each iteration until reaching the final stage, the position of each agent is updated. Before updating the parameters A, a, c, the algorithm goes through a phase of calculating the average. In the averaging phase, the probability of selecting the averaged wolf belonging to which category is determined. The probability of selecting wolves farther from the target is higher. The probabilities of selecting wolves are adjusted as follows.

$$P(X_a) = 0.1$$

$$P(X_\beta) = 0.2$$

$$P(X_\delta) = 0.3$$

$$P(X_\omega) = 0.4$$

In the final stage, the average fitness factor is calculated and if its value is higher than the selected gray wolf's fitness, the new position of the selected wolf is replaced with the average position. Then, parameters A, a, c are updated, fitness factors are calculated, they are categorized into alpha, beta, and delta groups, and finally the next iteration of the algorithm begins. The performance average of the wolves is controlled. Initially, in the execution of the gray wolf algorithm, we have very little pressure for averaging. During the algorithm execution, the pressure for averaging increases. Therefore, different values of average pressure during algorithm execution are achievable. In other words, the algorithm starts the averaging operation with very low pressure (almost zero) at the beginning and gradually increases the average pressure. To prevent the algorithm from getting stuck in a local optimum at the beginning of algorithm execution, exploration operations increase while maintaining diversity and changing the categorization of the wolves. Figure 7 illustrates the flowchart of the proposed method.

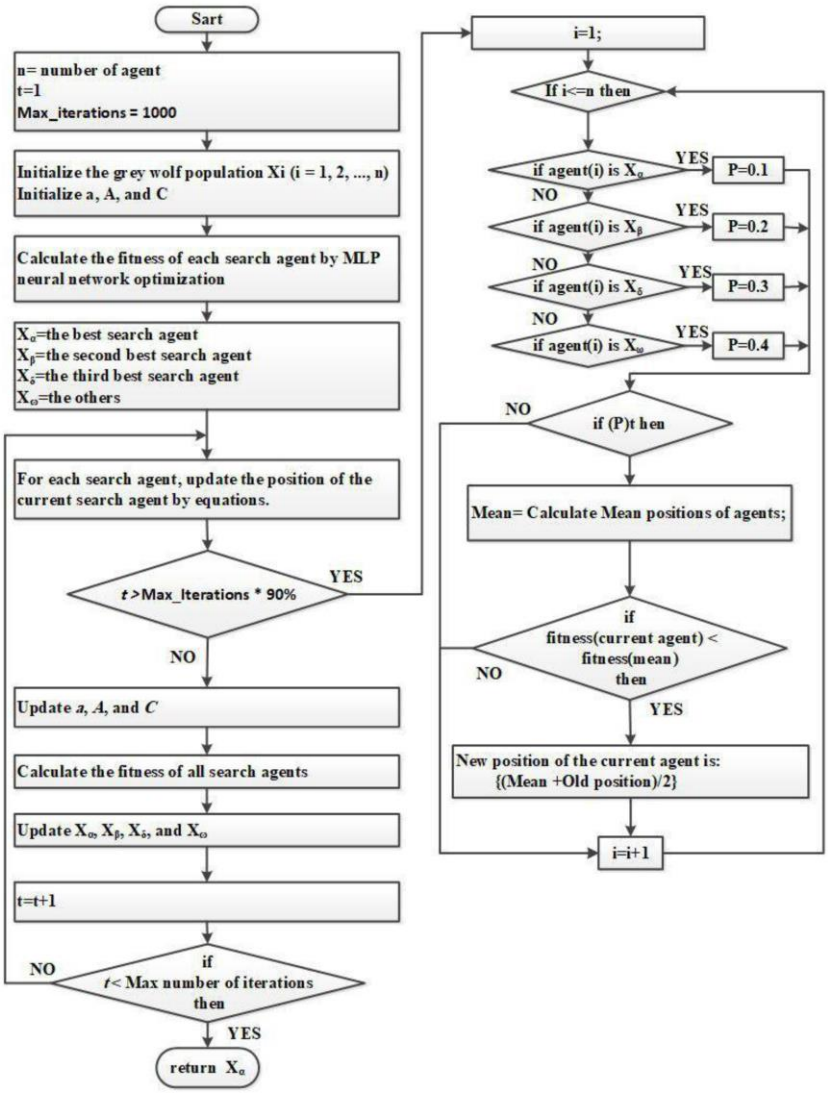

Figure 7: Proposed Method Flowchart

5. Results of the proposed method

5.1 Evaluation and Comparison of the Proposed Method Based on Optimization Algorithms

In the proposed method, to determine the value and role of each feature in intrusion detection, a random weight between the range [0,1] is assigned to each feature, indicating the importance of the feature.

The available data from the database (1999 KDD Cup) has been described, simulated, and analyzed using software (Matlab R2016b). The weighted feature values are inputted into a neural network (MLP) and optimized by evolutionary algorithms. Genetic algorithm and particle swarm optimization algorithms based on the objective function are used for predicting intrusion and improving the performance of the neural network (MLP). The vertical axis in the graph indicates the prediction error percentage, while the horizontal axis shows the number of algorithm iterations. In the initial iterations, a noticeable error reduction is observed since the initial population is random. However, in subsequent iterations, the error reduction decreases, and ultimately, the proposed method achieves a better error reduction at the end of the simulation. In Figure 8, a comparison of error values for predicting intrusion in the network using evolutionary feature weight optimization in 500 iterations is shown. The results in Figure 4-1 demonstrate the superiority of the algorithm (GWO) compared to algorithms (GBC, GA, PSO).

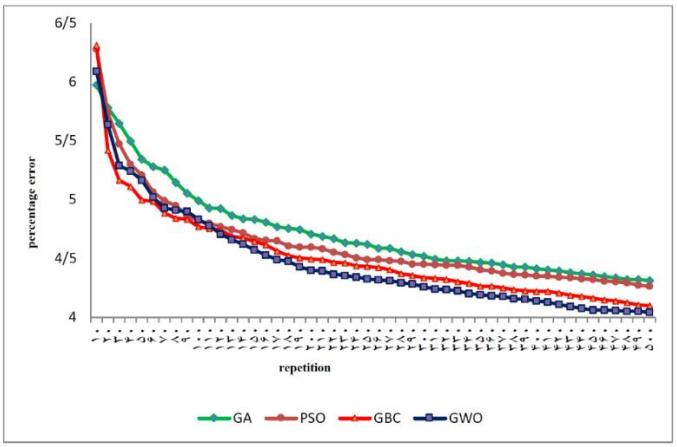

Figure 8: Error rate for network intrusion prediction in the proposed model

5.2 Evaluation and comparison of the proposed method based on machine learning algorithms

The proposed model has been compared with Bayesian methods, decision trees, and support vector machines. The relationship between actual classes and predicted classes can be calculated using a confusion matrix. The parameters required for the confusion matrix are mentioned in Figure 9.

To compare the proposed model with other methods, metrics such as Accuracy, Sensitivity, Specificity, Precision, and F-Measure are used based on Figure 2 equations.

$$\text{Accuracy} = (TP + TN) / All \quad (13)$$

$$\text{Sensitivity} = TP / (TP + FN) \quad (14)$$

$$\text{Specificity} = TN / (FP + TN) \quad (15)$$

$$\text{Precision} = TP / (TP + FP) \quad (16)$$

$$\text{Recall} = TP / (TP + FN) \quad (17)$$

$$\text{F Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

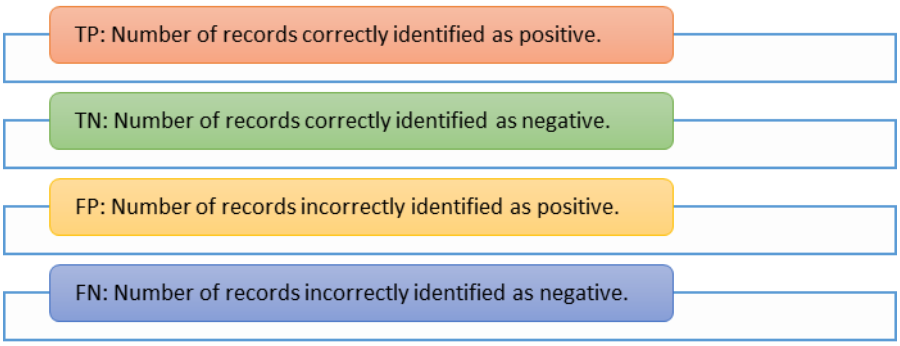

Figure 9: Parameters required for the communication between actual classes and predicted classes.

Table 9 shows the results chart of different detection methods with the metric of Accuracy. As seen, the proposed model has a higher accuracy compared to other methods. Additionally, a comparison of intrusion prediction results with criteria (sensitivity, specificity, accuracy, and F-measure) is presented in Tables 5 to 9. The comparison results demonstrate the superior performance of the proposed model. The values in the performance table indicate better performance of the proposed approach.

Table 5: Chart of results with Accuracy criteria

Bees	decision tree	Support vector machine	proposed model
0.93	0.94	0.92	0.96

Table 6: **Comparison of results with Sensitivity metric**

MLP & GWO	Bayes	SVM	Decision Tree	
0.98	0.96	0.97	0.96	Network intrusion
0.81	0.71	0.56	0.73	No network intrusion

In Table 6, a comparison of the proposed approach with Bayesian algorithms, neural networks, and decision trees for predicting network intrusion is shown. Evaluation metrics in the table indicate the superiority of classification in the proposed method with weighted features compared to other methods in a similar weighted feature scenario.

Table 7: **Comparison of results with Specificity metric**

MLP & GWO	Bayes	SVM	Decision Tree	
0.81	0.71	0.56	0.73	Network intrusion
0.99	0.96	0.97	0.96	No network intrusion

The evaluation of network penetration prediction based on feature criteria in Table 7 superiority of classification in the proposed method compared to Bayesian methods, neural networks, and decision trees. Comparing the results in the improved performance table shows classification features with weights compared to features modeled with similar weights.

Table 8: **Comparison of results with Precision metric**

MLP & GWO	Bayes	SVM	Decision Tree	
0.95	0.93	0.89	0.93	Network intrusion
0.92	0.82	0.92	0.83	No network intrusion

An accurate evaluation criterion for comparing the performance of classification with weighted features versus weighted modeling features for predicting network intrusion is shown in Table 8. The proposed approach aims to achieve better results in intrusion detection compared to Bayesian methods, neural networks, and decision trees.

Table 9: **Comparison of results with F-Measure metric**

MLP & GWO	Bayes	SVM	Decision Tree	
0.95	0.94	0.94	0.94	Network intrusion
0.85	0.76	0.70	0.77	No network intrusion

Based on the F-Measure metric, the proposed approach for predicting the presence or absence of network intrusion performs better compared to other methods. The comparison of results based on this metric indicates the superiority of the proposed approach over other compared methods. This evaluation demonstrates the performance of classification with weighted features compared to modeling features with similar weights.

6. Conclusion

Searching databases related to networks to gather knowledge and information for prediction, diagnosis, and decision-making is one of the applications of data mining in the industry. Innovative algorithms like the Gray Wolf Optimizer can be used to optimize data mining techniques. Accurately predicting and detecting network intrusions using artificial intelligence and machine learning increases the chances of successful treatment. In this study, the GWO algorithm was used for predicting and detecting network intrusions, optimizing the results of the neural network (MLP), and introducing a new model. Simulation results show that the proposed model performs better with a prediction accuracy of 96.0 compared to Bayesian methods, support vector machines, and decision trees. Predicting network intrusions using evolutionary methods for the neural network (MLP) has improved in this research. The suitable accuracy of the proposed method compared to other classification algorithms used in this study indicates the efficiency of the proposed method. When combined with GWO and MLP, the accuracy of this algorithm significantly increases compared to other classification algorithms. The high accuracy in detecting network intrusions demonstrates the superiority of the proposed method in improving neural network results (MLP). The complexity and time-consuming nature of execution are weaknesses of this method. The findings of the proposed method include: - Improving the performance of MLP neural networks using the GWO algorithm - The flexible structure of the proposed method in optimizing issues - Improving the number of layers and neurons in each layer of the MLP neural network to increase prediction accuracy - Optimizing feature weights in predicting network intrusions - Increasing prediction accuracy of network intrusions using the proposed method - Identifying and reacting quickly to attacks - Identifying different types of attacks and malicious attempts to breach - Enhancing network security and services - Completing other security components in the network

References

[1] Emary .E, Zawbaa .H.M., Zawbaa .A.E.(2016) Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 . 371–381, https://doi. org/10.1016/j.neucom.2015.06.083.

[2] Ghazal T M.(2022) Data Fusion-based machine learning architecture for intrusion detection. Computers, Materials & Continua, 70(2): 3399- 3413.

[3] Gligor .V. D.,(2006) “A note on the denial-of-service problem.” in IEEE Symposium on Security and Privacy, p. 139-149.

[4] Kumar, A., & Lee, S. (2024). Comparative analysis of ensemble learning methods for heart disease prediction. Journal of Medical Informatics, 35(2), 145-160.

[5] Li, J., Zhang, L., & Wang, Y. (2024). Hybrid convolutional neural networks and traditional machine learning algorithms for heart disease diagnosis. *Journal of Artificial Intelligence in Medicine*, 12(3), 113-125.

[6] Huang Y, Pullen JM,(2016) “Countering denial-of-service attacks using Congestion triggered packet sampling and filtering”, Presented at 10th International Conference on Computer Communications and Networks.

[7] Islam N, Farhin F, Sultana I, et al.(2021) Towards machine learning based intrusion detection in IoT networks. Comput. Mater. Contin, , 69(2): 1801-1821.

[8] Pourbahrami, S., Balafar, M. A., Khanli, L. M., & Kakarash, Z. A. (2020). A survey of neighborhood construction algorithms for clustering and classifying data points. Computer Science Review, 38, 100315.

[9] Kareem .S.S, Mostafa .R.R, Hashim .F.A., El-Bakry., (2022) An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection, Sensors 22 (4) , https://doi.org/10.3390/s22041396.

[10] Khan .S, Traore .I.,(2015) Queue-based analysis of DoS attacks, in: Proceeding of the 2015 IEEE Workshop on Information Assurance and Security, United States Mulitary Academy, West Point, NY, pp. 266–273.

[11] Kumar A,. (2020). “Intrusion Detection using Feature Selection and Machine Learning Techniques." Expert Systems with Applications 47: 106-119.

[12] Mohammed .H, Rashid .T.,(2023) FOX: a FOX-inspired optimization algorithm, Appl. Intell. 53 (1) 1030–1050, https://doi.org/10.1007/s10489-022- 03533-0.

[13] Kakarash, Z. A., Mardukhia, F., & Moradi, P. (2023). Multi-label feature selection using density-based graph clustering and ant colony optimization. Journal of Computational Design and Engineering, 10(1), 122-138.

[14] Mothukuri V, Khare P, Parizi R M, et al(2021). Federated-learning-based anomaly detection for iot security attacks. IEEE Internet of Things Journal, 9(4): 2545-2554

[15] Safa .H, Chouman .M, Artail .H, Karam .M.,(2014) "A collaborative defense mechanism against SYN flooding attacks in IP networks", Journal of Network and Computer Applications, Volume 31, Pages 509-534.

[16] SaiSindhuTheja .R , Shyam .G.K.,(2021) An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment, Appl. Soft Comput. 100 (Mar. 2021) 106997, https://doi.org/10.1016/j.asoc.2020.106997.

[17] Liu, H., Wang, Q., & Zhou, F. (2024). Personalized machine learning models for heart disease diagnosis: The role of genetics and lifestyle factors. Journal of Medical Genetics and Informatics, 31(1), 25-36.

[18] McDonald, J., Cooper, D., & Stevens, R. (2024). Generative adversarial networks for augmenting heart disease datasets. International Journal of Machine Learning in Medicine, 40(7), 1789-1802.
[19] Patel, V., Gupta, R., & Singh, K. (2023). Addressing imbalanced datasets in heart disease prediction using oversampling techniques. Healthcare AI Journal, 29(4), 210-222.
[20] Kakarash, Z. A., Ezat, H. S., Omar, S. A., & Ahmed, N. F. (2022). Time series forecasting based on support vector machine using particle swarm optimization. International Journal of Computing, 21(1), 76-88.
[21] Waheed N, He X, Ikram M, et al.(2020) Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures. ACM Computing Surveys (CSUR), 53(6): 1-37.
[22] Kakarash, Z. A., Karim, S. H. T., Ahmed, N. F., & Omar, G. A. (2021). New topology control base on ant colony algorithm in optimization of wireless sensor network. Passer Journal of Basic and Applied Sciences, 3(2), 123-129.
[23] Yang Wang , Chuang Lin , Quan-Lin Li, Yuguang Fang ,(2007)"A queueing analysis for the denial of service (DoS) attacks in computer network", Computer Networks 51 3564–3573.
[24] Kakarash, Z. A., Karim, S. H. T., & Mohammadi, M. (2020). Fall detection using neural network based on internet of things streaming data. UHD Journal of Science and Technology, 4(2), 91-98.
[25] Smith, T., Miller, H., & Thomas, P. (2023). Decision trees for heart disease prediction: A comparative study. International Journal of Health Informatics, 41(1), 56-67.
[26] Tan, J., & Lee, H. (2023). Unsupervised learning for feature extraction in heart disease diagnosis. Journal of Data Mining in Healthcare, 11(2), 84-95.
[27] Zhang, Y., Zhao, J., & Liu, B. (2022). Deep learning for heart disease diagnosis: Accuracy and interpretability challenges. Journal of Machine Learning in Healthcare, 18(5), 82-94.