IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

GOAMLP: Network Intrusion Detection with Multilayer Perceptron and Grasshopper Optimization Algorithm

SHADI MOGHANIAN[1] (Member, IEEE), FARSHID B. SARAVI[2] (Member, IEEE), GITI JAVIDI[3] (Member, IEEE), EHSAN O. SHEYBANI[3] (Senior Member, IEEE)

[1]Computer Science Department, Universidad Politécnica de Cataluña, Barcelona, Spain
[2]Research Consultant, CS-IT Hub, Florida, USA
[3]Muma College of Business, University of South Florida, Tampa, Florida, USA

Corresponding author: Ehsan O. Sheybani (e-mail: sheybani@usf.edu).

Abstract In this paper, an intrusion detection system is introduced that uses data mining and machine learning concepts to detect network intrusion patterns. In the proposed method, an artificial neural network (ANN) is used as a learning technique in intrusion detection. The metaheuristic algorithm with the swarm-based approach is used to reduce intrusion detection errors. In the proposed method, the Grasshopper Optimization Algorithm (GOA) is used for better and more accurate learning of ANNs to reduce intrusion detection error rate. The role of the GOAMLP algorithm is to minimize the intrusion detection error in the neural network by selecting useful parameters such as weight and bias. Our implementation in MATLAB software and using the KDD and UNSW datasets show that the proposed method detects abnormal, malicious traffic and attacks with high accuracy. The GOAMLP method outperforms and is more accurate than the existing state-of-the-art techniques such as RF, XGBoost, and embedded learning of ANN with BOA, HHO, and BWO algorithms in network intrusion detection.

Index Terms Network intrusion detection, data mining, machine learning, artificial neural network, multilayer perceptron, swarm-based algorithm

I. INTRODUCTION

Computer networks have grown significantly in recent years. A variety of them have been introduced and presented for different applications with different benefits. Some good examples include wireless sensor networks (WSNs) [1], vehicular ad hoc networks (VANETs) [2], and the internet of things (IoT) [3]. One of the standard features of these networks is that each device or node can share its information with other nodes over the internet [4].

The main problem with different computer networks is their security issues, which pose a significant challenge. In various computer networks, attackers and hackers have repeatedly sought gaps to infiltrate the network configuration, steal valuable network information, and disrupt normal function of the network. To this end, hackers or organized cyber-attacks are capable of disrupting entire countries' computer networks. Thus, network intrusion has widespread repercussions [5].

In software-based methods, the hacker executes malicious code on a network server and disrupts its services, sometimes denying service even without running any code on the server. This attack is achieved simply by repeating a massive number of requests seen as a distributed denial of service (DDoS) attack [6].

Hackers try to break into networks by exploiting network tools such as port scanning used by network nodes. In this case, the hacker exploits ports that were dropped due to incorrect configuration or simple access levels. An example of these services is SSH and telnet [7].

Network intrusion can be the result of hackers, (1) guessing weak passwords, (2) using password guessing software, or (3) attacking the network via brute force [8]. By leveraging social engineering techniques, hackers can also communicate with people in social networks, messengers, and e-mails to obtain important information from the system toward network intrusion [9].

Network intrusion entails unusual traffic and thus has a set of features that distinguish it from regular traffic. Existing patterns in unusual traffic can be detected by various data mining and machine learning methods that facilitate an Intrusion Detection System (IDS). Network intrusion has a devastating effect on the network and can even disable the entire network. Therefore, various efforts have been aimed at designing intrusion and firewall systems to surmount this security challenge. Most intrusion detection methods attempt to remove or filter unauthorized network traffic via intrusion pattern recognition [10].

In other words, intrusion detection methods try to discretize normal and unusual traffic. Recent studies have designed network intrusion detection systems (NIDS) [11], in one example introducing a new IDS based on K-means clustering and the

Firefly algorithm (FA). In this method, the Firefly Algorithm is used to improve the K-means clustering algorithm to detect the intrusion system. Another relevant study [12] has conducted extensive research on the NIDS of vehicular ad hoc networks (VANETs) and evaluated IDS based on their performance.

The use of fuzzy clustering to design an IDS in the cloud computing space has also been demonstrated [13]. Furthermore, a new method has applied machine learning and concepts of an evolutionary algorithm in intrusion detection [14]. In this proposed approach, the support vector machine (SVM) was chosen as the learning method in intrusion detection, and a Decision Tree (DT) was utilized to pick up features of the attacks. Moreover, Genetic Algorithm (GA) techniques were leveraged to improve feature selection. The findings confirmed that this method outperforms SVM in accurate intrusion identification.

IGA has also been used to discover effective rules for NIDS [15]. In the example cited, each set of rules was considered a variable-length chromosome. The implementation demonstrated that this method offers some quality rules for intrusion detection. Similar approaches using SVM and kernel principal component analysis have been taken with the GA mechanism [16].

IDS is also of paramount relevance to Wireless Sensor Networks (WSNs) [17]. IDS is vulnerable to various types of attacks such as flooding, root attack, port scanning, backdoor, etc. in the cloud computing space [18]. It is shown that ANN can be used as a data mining technique to identify the intrusion of the computer network. The GOA algorithm with a swarm-based approach can be used to further improve ANN accuracy in the IDS.

Some studies in IDS use a blacklist to detect network intrusion; however, these methods are not very useful because blacklists need to be updated continuously. On the other hand, data mining and machine learning methods are subject to significant error. We tried to minimize the error as much as possible. These methods enable considering this challenge as an optimization problem with a minimization approach that aims to reduce the detection error between normal and unusual traffic.

Among the recently proposed metaheuristic algorithms with the swarm-based approach is the Grasshopper Optimization Algorithm (GOA), which has been proven more accurate than the Particle Swarm Optimization Algorithm (PSO), Genetic Algorithm (GA), Bat Algorithm (BA), Firefly Algorithm (FA), Gravitational Search Algorithm (GSA), Flower Pollination Algorithm (FPA) and Differential Evolution (DE) [18].

Studies show that designing a NIDS requires an accurate classification process. There are several methods for classification, like the artificial neural network (ANN) method. ANN has different types; one of the simplest and most practical ones of which is the multilayer perceptron (MLP).

An essential advantage of an MLP compared to other neural networks (such as deep neural networks) is that MLP requires less time to learn. Reducing learning and analysis time is critical in an IDS.

In this section, we have reviewed the NIDS. The remainder of this paper is organized as follows. We introduce the classification of metaheuristic algorithms, GOA algorithm, and multilayer perceptron of ANN in Section II. In Section III, we formulate and present the proposed method. In Section IV, we provide the implementation and analysis of experiments. Finally, we offer conclusions for this paper in Section V.

II. RELATED WORK

A. METAHEURISTIC ALGORITHMS

Metaheuristic algorithms are a set of pseudo-random methods based on the organisms' behavior or physical phenomena in nature. These algorithms can be modeled on organisms' behavior or non-living physical or mathematical phenomena.

In metaheuristic algorithms, each solution is coded as a natural element. These solutions (i.e., factors) are sent to the optimal answer via modeling. Several metaheuristic algorithms are divided into different categories based on their respective algorithmic approaches [19]. In this classification, metaheuristic algorithms are divided into six groups (Evolutionary Algorithms, Swarm-based Algorithms, Physical-based Algorithms, Human-based Algorithms, Bio-inspired algorithms, Nature-inspired algorithms), as summarized in Fig.1.

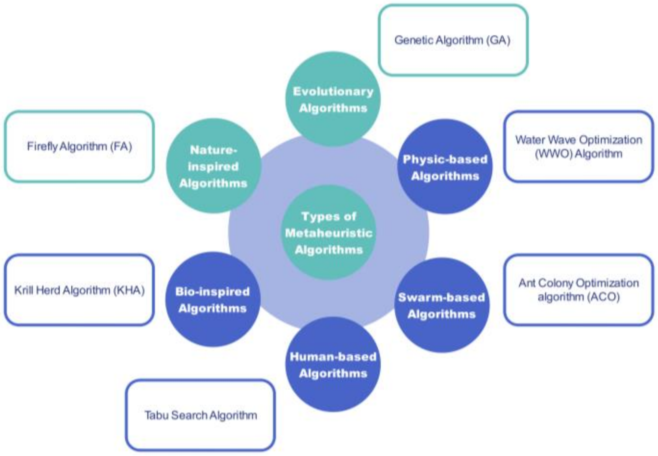

Figure 1. **Classification of metaheuristic algorithms**

Evolutionary algorithms use the principles of competence and survival. The more appropriate a solution is, the more survival is promoted. These algorithms can produce similar solutions to those of the Genetic Algorithm (GA). Physical-based Algorithms are comprised of a physical element or are based on the laws of physics. Swarm-based Algorithms take the approach of modeling organisms' behaviors as a group. Human-based Algorithms give way to models of human behavior that reflect political, cultural, or social dimensions to solve optimization problems. In Bio-inspired algorithms, the swarm behaviors of cells or microscopic organisms lacking higher intelligence are used. Ultimately, Nature-inspired algorithms are partial to the use of behaviors from organisms for which intelligence is weak or absent [20].

1) GRASSHOPPER OPTIMIZATION ALGORITHM (GOA)

Metaheuristic algorithms are typically classified into four categories based on Evolutionary Algorithms such as Genetic Algorithm (GA), Differential Evolution (DE), Physics-based algorithms, Gravitational Search Algorithm (GSA), Swarm-based algorithms, Particle Swarm Optimization (PSO), Ant Colony

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040740, IEEE Access

Optimization (ACO) Algorithm, Human-based algorithms, and Teaching-Learning-Based Optimization (TLBO) Algorithms [20].

The GOA is one of the population-based and swarm-based algorithms that has been used to model grasshopper invasion into farms and green plains. Grasshoppers are a group of insects that live together with a prescribed set of interactions. Thus, in this algorithm, each solution is considered as a single grasshopper. In each iteration, there exists a set of solutions in the form of the grasshopper population. The grasshopper swarm-based algorithm is modeled using three factors: tendency toward group, wind, and gravity [21].

In this algorithm, each solution (i.e., grasshopper) is updated according to the above three factors. Equation (1) models how each grasshopper is affected by the three aforementioned laws of tendency toward grasshopper aggregation, wind direction, and gravity [22]:

$$X_i = S_i + G_i + A_i \tag{1}$$

Here, S_i , G_i and A_i are defined as an attractive factor for the grasshopper population, effect of the Earth's gravity on the grasshopper's movement, and the wind factor in locating the grasshopper's activity, respectively. The GOA uses the function $s(d_{ij})$ to control the intended steps of each grasshopper to navigate the problem space based on the distance between two grasshoppers, shown in equation (2) [22]:

$$s(d_{ij}) = f e^{\frac{-r}{l}} - e^{-r} \tag{2}$$

In this equation, f and l are the absorption coefficient of the grasshopper population and the absorption factor in the group, respectively; r is the distance between two grasshoppers in the group or population. In Fig.2, the performance of the function s is shown in the grasshopper's movement toward the center of the group. This illustrates how a grasshopper can modulate its distance from a swarm of grasshoppers based on the output value of the function $s(d_{ij})$.

The grasshoppers' movement toward a swarm of grasshoppers, which is just an effective behavior, changes the problem solutions [22]:

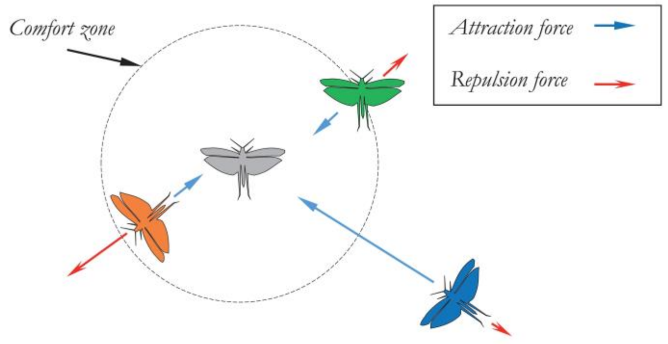

Figure 2. Disposal or absorption of grasshoppers toward a swarm of grasshoppers [22]

The GOA does not directly use the wind, gravity, and movement toward a swarm of grasshoppers to change the grasshoppers' position; rather, the optimal answer is the center of the grasshopper swarm's position. This gives way to the more practical equation (3) [17]:

$$X_i^d = c.\left[\sum_{j=1}^{N} c \frac{ub_d - lb_d}{2} s(|X_j - X_i|) . \frac{X_j - X_i}{d_{ij}}\right] + \hat{T} \tag{3}$$

In this equation, $\hat{T}$ is the most optimal position of a swarm of grasshoppers, ub_d and lb_d are the upper and lower range d of the objective function, respectively, and c is the convergence coefficient in the GOA - which is a decreasing value in terms of its iteration. The above equation is performed on each population member of the GOA in successive iterations to move the population members toward optimal solutions [23].

B. MULTILAYER PERCEPTRON (MLP) OF AN ANN

A neuron has three main parts: nucleus, dendrites, and axons. Each of these sections is considered in an ANN to solve complex problems. The network comprising the nervous system (i.e. neural network) is a complex problem-solving structure made up of billions of neurons. Algorithms emulating the behavior of this natural system have made it possible to solve challenging and complex problems. A single neuron performs weakly in decision making, but a set of neurons can be used to identify patterns and classifications. Typically, an ANN is comprised of different sections, depicted in Fig.3. In each section, several artificial neurons interact to create the output of an ANN:

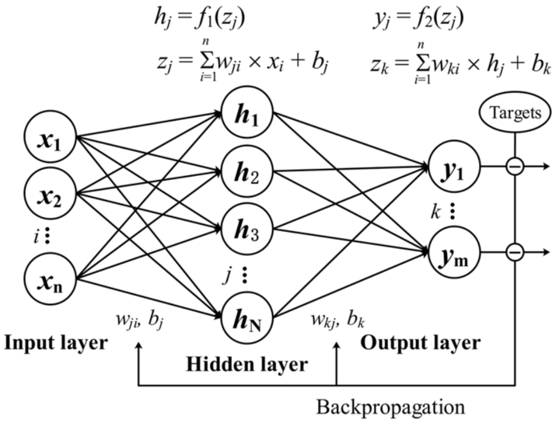

Figure 3. The structure of an ANN [24]

They present a new MLP prediction model (MLP-PM) to find the base station. E-ANDSF invokes MLP to select a base station and to predict the QoS [25].

C. INTRUSION DETECTION SYSTEM (IDS)

This section covers various studies carried out on the IDS. In Table I, we compare algorithms, methods, challenges, and advantages.

TABLE I
A COMPARISON OF INTRUSION DETECTION SYSTEMS

Authors	Methods	Challenges	Advantages
Cai et al. [26]	Introducing a method of IPS	It requires a lot of training	The most crucial advantage is to anticipate and prevent intrusion
Khraisat et al. [27]	They presented a review article of the IDS	Failure to investigate the IDS in new networks such as the IoT	Effective classification of the IDS
Elhag et al. [28]	Developed a multi-objective evolutionary fuzzy system for the IDS	The accuracy depends on the training, and the uncertainty of the system is high	This design allows the end-user to decide from a wide range of solutions which one is best suited for the current network features
Ben et al. [29]	They presented a method for the NIDS using swarm intelligence behavior and artificial neural network	Training time and detecting an intrusion is high	Their proposed approach is more accurate than the particle swarm optimization (PSO) and genetic algorithms (GA).
Kabir et al. [30]	Provided a statistical sampling and classification technique with the support vector machine (SVM) technique for the NIDS	The accuracy of the proposed method depends on the accurate sampling of the network traffic	The proposed method is more accurate and better performance than other learning methods such as SVM in the NIDS
Shah et al. [31]	Investigation of Snort and Suricata of the IDS for computer networks	It requires plugins to increase the accuracy	The hybrid system is more accurate than the SVM and the improved SVM with fuzzy logic
Hamed et al. [32]	Feature selection approach (RFA) based on the recursive method and graph-based probabilistic principles	Unable to detect most attacks and focus on the other attacks	False alarm rate in the proposed is less than other methods, and the accuracy is reported up to 92.9% in the NIDS
Kumar et al. [33]	An integrated system for detecting intrusion into the IoT using machine learning techniques	Implementation is based on synthetic data, not actual data	Their proposed method has less computational complexity than other similar methods
Ven et al. [34]	An Adaptive Hybrid IDS to combat DDoS attacks	Only able to detect DDoS attacks	The proposed method is a successful hybrid model for the IoT such as smart cities
Salim et al. [35]	A review of DDoS attacks against the IoT networks	Investigate a range of security challenges related to DDoS attacks	Discusses a comprehensive study of DDoS attacks from the IoT devices to the cloud environment
Shorman et al. [36]	Providing an OCSVM unsupervised intelligent learning system based on the Gray Wolf optimization algorithm for intrusion detection	An optimized SVM (OCSVM) parameters with the Gray Wolf algorithm takes a long time	The proposed method has less error detection for botnet than similar methods and detects them in less time
Sultana et al. [37]	Presented an SDN-based NIDS using machine learning (ML) techniques	The volume of data is more than what was analyzed in the IoT by traditional learning methods	Limited research on SDN-based NIDS
Jallad et al. [38]	Provided Big data processing platform for NIDS with deep learning (DL) mechanisms	A significant number of systems or clusters are needed to set up Apache Spark	Apache Spark platform can do deep learning in near real-time and detect the real-time intrusions
Nafessah and Casale [39]	An ANN-based learning technique in Apache Spark processing environment for the detection of malformations	Reduces processing speed in Apache Spark	Apache Spark platform is capable of processing Big Data traffic in a short time
Niksefat et al. [40]	Presented a review article on the NIDS and the challenges	NIDS has not been analyzed and is generally discussed	One of the essential benefits of this research is the classification of the IDS based on the nature of the data used in network traffic
Pozi et al. [41]	A model based on machine learning and evolutionary algorithm concepts is used	The proposed method or GPSVM has more time to detect intrusion than the SVM method	The proposed method (GPSVM) is more accurate than the SVM method
Anthi et al. [42]	Introduce an IDS for the IoT devices	Limitations in computing power, lack of encryption in data transmission, unsecured web interfaces, lack of authentication/authorization mechanisms, and their heterogeneity using uniform security mechanisms	The activity of all malicious or non-destructive IoT devices on the network is detectable, and attacks are automatically detected and logged by the internal hardware
Niak et al. [43]	Proposed E-TLBO-FLANN method using TLBO and neural network for the IDS	Using TLBO to obtain suitable parameters for the functional link neural net (FLANN) and weighted-average using elitism	The processing of duplicate parameters is avoided by using the mutation procedure. The proposed method is more effective in terms of the computational burden
Farivar et al. [44]	Artificial intelligence is used to diagnose the detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems	Linear dynamic systems are not investigated. The optimal parameter values are not unique, and the STD and mean vectors of Gaussian functions are not trained	The proposed approach uses nonlinear cyber-physical systems (CPS) and Radial Basis Function Neural Network (GRBFNN) as an intelligent estimator to reconstruct and compensate of cyber-attacks
Liang et al. [45]	An industrial NIDS based on a multi-feature data clustering model improves the detection rate and performance of detecting abnormal behavior	The operation and information need to be analyzed to diagnose illegal nodes or systems under threatening on IDSs	The proposed method has excellent superiority in terms of discovery time and rate compared to other algorithms

III. THE PROPOSED METHOD

This section presents the GOAMLP method for reducing the ANN error rate with the GOA algorithm and MLP. We begin by presenting the challenge, following which the framework of the proposed method is described. Also, equations and modeling the GOAMLP algorithm are discussed. Finally, a flowchart illustrating the proposed method for detecting network intrusion is presented.

For NIDS, machine learning methods and data mining like MLP of ANN are essential tools to analyze network traffic. To diagnose the small error rate of intrusion detection, weight and bias need to be well-evaluated and optimized. There exist several candidate strategies for optimization, including the GOA algorithm. GOA is an intelligent algorithm to solve np-hard problems. It gradually changes the nature of the search from global to local. On the other hand, there is no high complexity for this intelligence swarm algorithm. It is more accurate than conventional metaheuristic algorithms like genetic (GA) and particle swarm optimization (PSO) algorithms, thereby motivating the development of the proposed method.

ANN is an efficient method for classification in various areas. One of the essential applications of ANN is detecting network intrusion. In recent studies, metaheuristic algorithms have been used to improve and reduce classification errors in the multilayer perceptron of ANNs. In most of these studies, reducing the ANN's output error has been considered the main challenge. For example, the GOA algorithm has been used to minimize ANN error in breast cancer diagnosis [46]. Relative to this approach, we believe our proposed method to be superlative for multiple reasons outlined below:

- The formulation of the method is limited to a two-layer perceptron of the ANN. In contrast, the method proposed herein assumes that each grasshopper has several sections for the hidden weight layers.
- The article's authors have relied exclusively on a medical dataset with a limited number of samples and features. Our method is based on the KDD dataset with 42 features and many records for detecting network attacks.
- An extraordinary advantage of our method is the use of feature selection in the preprocessing phase. The GOA algorithm is leveraged to reduce the dimensionality of big data through feature selection. Thus, the dimension of the network traffic is reduced and considered as the input of an ANN. This dimensionality reduction in network traffic means that ANN "learning" is performed only on essential features. Then, the GOA algorithm is responsible for assigning the network weight and bias.
- Additionally, the aforementioned article is a case study of a disease; whereas, our case study offers the terrific advantage of being the network. Moreover, our method offers feature selection and dimensionality reduction to facilitate learning that runs on substantial features.

In our method, the blacklist concept is used in the ANN to interrogate the source traffic addresses in the list first. If the attack pattern is not found in the list, then a learning mechanism is carried out by the ANN to reduce detection time.

A. PROPOSED METHOD PHASES

To reduce NID error rate by ANN, weight and bias parameter selections are performed with comprehensive precision. NIDS neural network output error is associated with optimum weight and bias. To decrease the MLP of ANN output error in NIDS, the introduced method will utilize the following process:

- Weight and bias of artificial neural networks are coded in the form of a vector. This vector is a grasshopper (i.e. a member of the GOA population).
- Each grasshopper (i.e. weight and bias vector) is analyzed with a cost function or average error rate of the NIDS in each iteration. GOA updates these vectors in each itearation.
- In the last iteration, the most optimum weight and bias vector are used for learning in the context of using ANN toward optimizing the output rate of the model.
- ANN is coded in the form of a member of the grasshopper optimization population.
- The primary population is initiated from random solutions in a grasshopper population that an ANN considers as a grasshopper.
- The population is analyzed with average NID error.
- Grasshopper algorithm model is applied in neural networks, and weights and biases are updated.
- Most optimum ANN is selected in each iteration.
- The best ANN with optimal weights and biases is extracted.

We will next discuss choosing the amount of weight and bias of ANN for decreasing the NID error. In Fig.4 are proposed steps for the NIDS with selection of optimum ANN weight and bias explained. It can be inferred that its mechanisms are akin to the following:

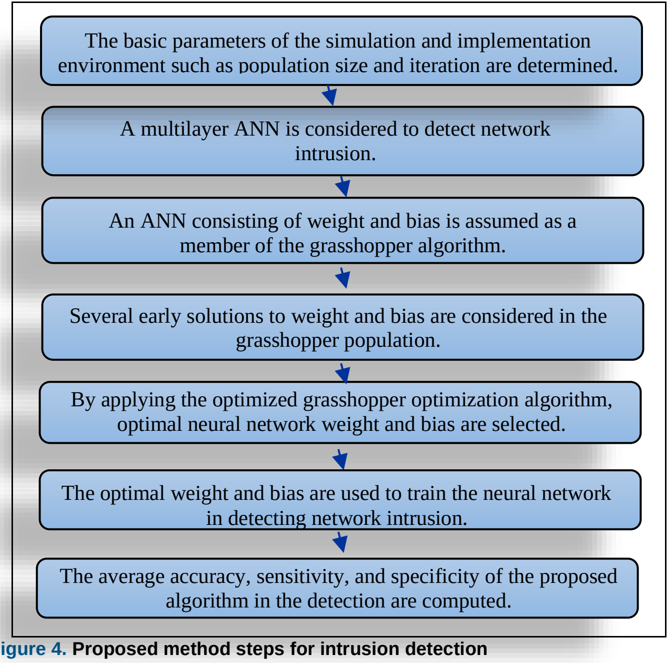
The basic parameters of the simulation and implementation environment such as population size and iteration are determined.
A multilayer ANN is considered to detect network intrusion.
An ANN consisting of weight and bias is assumed as a member of the grasshopper algorithm.
Several early solutions to weight and bias are considered in the grasshopper population.
By applying the optimized grasshopper optimization algorithm, optimal neural network weight and bias are selected.
The optimal weight and bias are used to train the neural network in detecting network intrusion.
The average accuracy, sensitivity, and specificity of the proposed algorithm in the detection are computed.

Figure 4. Proposed method steps for intrusion detection

B. PROPOSED METHOD FRAMEWORK

In Fig.5, the decreasing mechanism of multilayer ANN error with GOA is shown. This approach contrasts with standard framework by reducing ANN error in last phase learning by using GOA to determine optimum weight and bias in a neural network. The below schematic summarizes how we first utilize dimension reduction by feature selection on input traffic, following which ANN weight and bias are updated for error reduction by GOA in both normal and abnormal detection contexts.

According to the above diagram, the following steps can be taken toward NIDS using MLP of ANN and GOA:

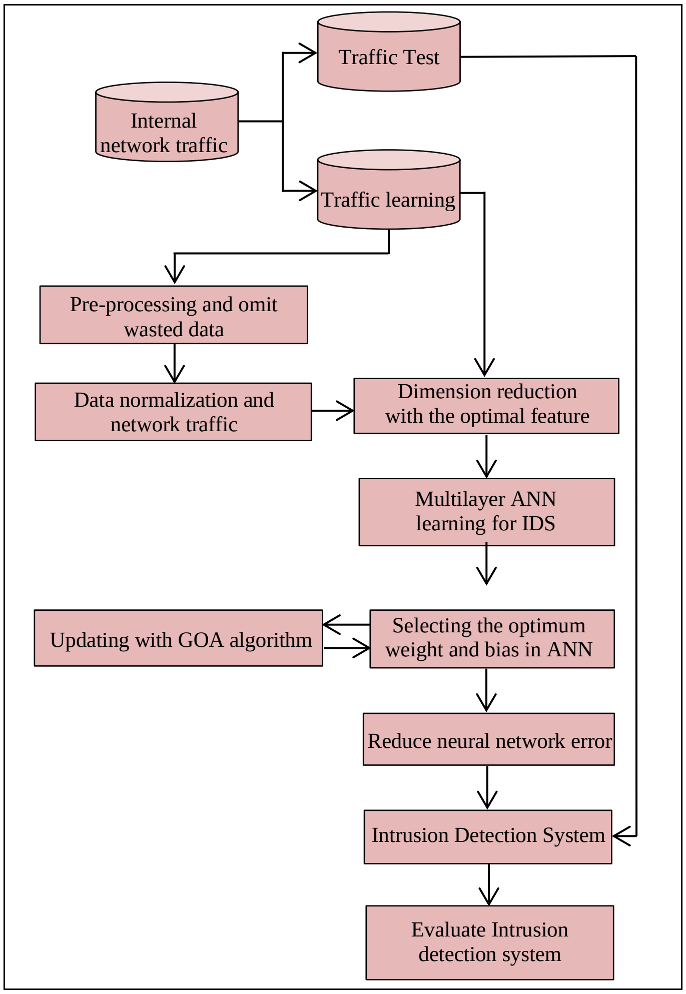
Traffic Test
Internal network traffic
Traffic learning
Pre-processing and omit wasted data
Data normalization and network traffic
Dimension reduction with the optimal feature
Multilayer ANN learning for IDS
Updating with GOA algorithm
Selecting the optimum weight and bias in ANN
Reduce neural network error
Intrusion Detection System
Evaluate Intrusion detection system

Figure 5. Improvement of ANN precise by GOA optimization algorithm for the NIDS

The preprocessing phase of the method in this paper contains two sections. First, the data is normalized and reduced in dimensionality by feature selection. In the learning phase, the network traffic must be preprocessed and normalized. Normalization is a process in which each traffic attribute's value is placed in a specific interval, such as$[a, b]$. The amplitude changes of all qualities and characteristics are in this interval. Equation (4) is used for the preprocessing of the normalization type:

$$f' = \frac{f-min}{max-min}(b-a)+a \tag{4}$$

In this equation, f', f, min, max, a, and b are the normalized value, the abnormalized value of a feature in traffic, the lower and upper normal range for traffic, and its features. In the proposed method where the range is considered in the interval [0,1], normalization is shown in equation (5):

$$f' = \frac{f-min}{max-min} \tag{5}$$

Traffic normalization and its features increase ML, data, and traffic accuracy into normal and abnormal classifications. After normalizing the network traffic, dimension reduction and feature selection can be applied. In the proposed method, GOA is used for feature selection. A feature vector of network traffic is considered a grasshopper. Each feature vector has a zero, and one component shows no feature selection and feature selection, respectively. In the present method, a feature vector of network traffic for intrusion detection is defined by equation (6), which is considered a grasshopper, i.e. a member of the GOA:

$$GOA_i = [F_i^1, F_i^2, F_i^3, \dots, F_i^D] \tag{6}$$

G_i is a grasshopper or a feature vector, F_i^j is the $j\text{-}th$ component of the feature vector, and D is the number of potential properties in a feature vector. In this equation, each component is equal to 0 or 1, indicating lack or existence of feature selection, respectively. In this equation, GOA_i is a feature vector in network traffic analysis. Several feature vectors are created randomly as several grasshoppers. The feature vector is used to map network traffic to reduce the input of the classification technique in repetition. This technique has a definite error that can be used to scale up the feature vector. A feature vector is appraised based on the number of selected attributes and the average error of detecting normal from abnormal traffic. For this case, the objective function in equation (7) is suggested [47]:

$$Cost = \alpha.E + \beta.\frac{f}{F} \tag{7}$$

In this equation, each feature vector has a lower value for $Cost$, which is picked out as the optimal feature vector. E is the error of detection between normal and abnormal traffic, f is the number of selected features, and F is all selected features. Meanwhile, α is a random number between 0 and 1, and β is equal to $1-\alpha$. Minimizing this objective function makes the selected feature vector optimal. It reduces the minimum error and number of features to minimize the firewall warning's dimensions and error.

By implementing the GOAMLP algorithm on feature vectors, the optimal feature vector can be calculated to reduce dimensions. It also reduced the feature space and trained the neural network inputs to reduce properties. The GOAMLP algorithm can improve the weight and bias to minimize the intrusion detection error.

C. SEVERAL STEPS OF THE PROPOSED METHOD

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3040740, IEEE Access

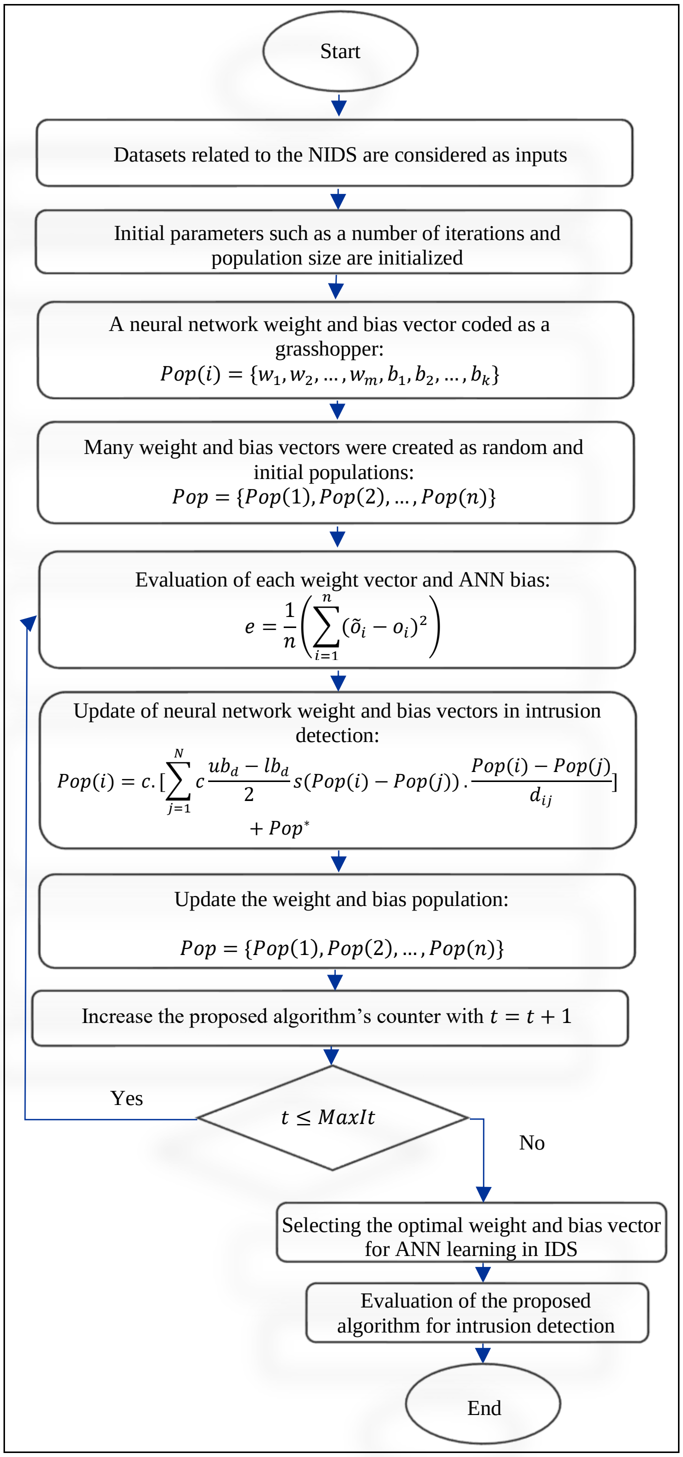
Start
Datasets related to the NIDS are considered as inputs
Initial parameters such as a number of iterations and population size are initialized
A neural network weight and bias vector coded as a grasshopper:
$Pop(i) = \{w_1, w_2, \ldots, w_m, b_1, b_2, \ldots, b_k\}$
Many weight and bias vectors were created as random and initial populations:
$Pop = \{Pop(1), Pop(2), \ldots, Pop(n)\}$
Evaluation of each weight vector and ANN bias:
$e = \frac{1}{n}\left(\sum_{i=1}^{n}(\tilde{o}_i - o_i)^2\right)$
Update of neural network weight and bias vectors in intrusion detection:
$Pop(i) = c.[\sum_{j=1}^{N} c\frac{ub_d - lb_d}{2}s(Pop(i) - Pop(j)).\frac{Pop(i) - Pop(j)}{d_{ij}}] + Pop^*$
Update the weight and bias population:
$Pop = \{Pop(1), Pop(2), \ldots, Pop(n)\}$
Increase the proposed algorithm's counter with $t = t + 1$
$t \leq MaxIt$
Yes
No
Selecting the optimal weight and bias vector for ANN learning in IDS
Evaluation of the proposed algorithm for intrusion detection
End

Figure 6. **Flowchart of the proposed method for the NIDS**

The flowchart in Fig.6 depicts the steps of the proposed method for the NIDS.

According to the proposed flowchart, it is first necessary to encode a multilayer ANN into a grasshopper. Equation (8) is a set of weights and biases considered as an array:

$$Pop(i) = \{w_1, w_2, \ldots, w_m, b_1, b_2, \ldots, b_k\} \tag{8}$$

In this equation, $\{w_1, w_2, \ldots, w_m\}$ and $\{b_1, b_2, \ldots, b_k\}$ represent the weights and biases, respectively, of a multilayer ANN coded as a grasshopper herein. In the proposed method, several solutions - or grasshoppers - that are weight and bias vectors, are created randomly as the initial population based on equation (9):

$$Pop = \{Pop(1), Pop(2), \ldots, Pop(n)\} \tag{9}$$

Here, Pop is the population of the GOA or an equivalent set of ANNs, and n is considered as the population number or initial population of the GOA. Each population member of the GOA (or equivalent of each ANN) needs to be estimated to define its appropriateness. It uses the average classification error in detecting intrusions for n data, according to equation (10):

$$e = \frac{1}{n}(\textstyle\sum_{i=1}^{n}(\tilde{o}_i - o_i)^2) \tag{10}$$

Here, o_i and $\tilde{o}_i$ are considered the primary and predicted class of i-th network traffic, respectively. In each iteration, according to equation (11) of the GOA algorithm, it performs on weight and bias vectors or grasshoppers' population. By using this equation, weights and biases update in each iteration:

$$Pop(i) = c.\left[\textstyle\sum_{j=1}^{N} c\frac{ub_d - lb_d}{2}s\big(Pop(i) - Pop(j)\big).\frac{Pop(i) - Pop(j)}{d_{ij}}\right] + Pop^* \tag{11}$$

In this equation, Pop^* is the weight and bias vector, and $Pop(i)$ and $Pop(j)$ are two weight and bias vectors wherein i and j represent the grasshopper population.

Meanwhile, s considers the flight step of the grasshopper. The above equation ensures that every weight and bias applied to the network is updated. By repetition of the algorithm, weight and bias are repeatedly modified to reduce intrusion detection error rate by ANNs.

By selecting the optimal ANN (i.e. fit grasshopper population), one can obtain a multilayer ANN optimized by its weights and bias; thus, reduced error rate among the optimum generated neural networks may be possible toward network intrusion detection. In algorithm 1, the pseudo-code of the GOAMLP algorithm is shown.

Algorithm 1: Pseudo-code of GOAMLP Algorithm

```
Input data, number of maxiter and Population, etc
Coding each grasshopper population: $Pop(i) = \{w_1, w_2, \ldots, w_m, b_1, b_2, \ldots, b_k\}$
Initialize grasshopper population: $Pop =\ll Pop_1, Pop_2, \ldots, Pop_N \gg$
Initialize $C_{Min} = 0.00004, C_{Max} = 5, f = 0.5, l = 1.5$
$BestSol = []$;
Neural network training with any weight and bias
Calculate the fitness of each grasshopper using $Ee = \frac{1}{n}(\sum_{i=1}^{n}(\tilde{o}_i - o_i)^2)$
Calculate the best weight and bias vector or Pop*
While $(it < Maxiter)$
        $c = C_{Max} - it * ((C_{Max} - C_{Min})/MaxIt)$
    For $i = 1:N$
        $sum = 0;$
        $d1 = (VarMax - VarMin)./2;$
        $c1 = c.* d1;$
            For $j = 1:N$
                If i~=j
                    $Dij = norm(pop(i).Position - pop(j).Position)$
                    $t = (pop(j).Position - pop(i).Position)./Dij$
                    $r = abs(pop(i).Position - pop(j).Position)$
                    $r = 1 + rem(r, 3)$
                    $S = f * exp(-r/l) - exp(-r)$
                End
            End
            $pop(i).Position = c.[\sum_{j=1}^{N} c \frac{ub_d - lb_d}{2} s(Pop(i) - Pop(j)).\frac{Pop(i)-Pop(j)}{d_{ij}}] + Pop^*$
            Neural network training with any weight and bias of Grasshopper
            **Calculate** the fitness of each weight and bias vector by cost Function and $set\ pop(i).Cost$
            **Update** Best Weight and Bias Vector or $Pop^*$
                    **If** $pop(i).Cost < BestSol.Cost$
                      $BestSol = pop(i)$
                    **End**
      **End**
**End while**
**Return** best weight and bias vector
Neural network training using the optimal weight and bias vector
```

IV. RESULTS AND DISCUSSION

A. DATASET

By modeling the proposed method in the NIDS at this level, we implement and evaluate the algorithm with relevant datasets such as KDD in the MATLAB programming environment. In this research, the KDD dataset is used to assess and analyze the proposed method in the NIDS.

The KDD dataset [48] comprises of data transmitted to the US Air Force network. It can be used to evaluate the accuracy and efficiency of NIDS algorithms. This dataset has three essential features:

- It is a compilation of normal and suspicious traffic, thus serving as an appropriate dataset for intrusion detection.
- This dataset is longitudinal and thus is not limited to instantaneous network traffic.
- The network traffic in this dataset is collected from a single host and entire network nodes.

The KDD dataset has 41 features, of which 34 are considered numerical, and the rest are non-numerical. There are four kinds of attacks used for classification in this dataset: U2R, R2L, DOS, and Prob. The dataset contains 42 different attributes each of which describes a traffic attribute such as the type of protocol used for traffic, the type of service requested by users, status of the flag, the number of sent or received bytes, segmentation errors, etc. [49].

Implementation of the algorithm for our proposed NIDS method is benefitted by the availability of an appropriate dataset (i.e. KDD) [48] and programming environment (i.e. MATLAB). In this implementation, average intrusion detection error based on GOA repetition is an imperative index and criterion for evaluating the optimum neural network weight and bias.

B. DATASET CHARACTERISTICS

As mentioned before, the KDD dataset has many records. Only a percentage of them are used to measure the efficiency of the NIDS algorithms [50]. This dataset has 42 different properties, each of which describes a traffic feature. These features are summarized in Table II.

TABLE II
KDD DATASET ATTRIBUTES [50]

No	Attribute name	No	Attribute name	No	Attribute name
1	Duration	15	Su_attempted	29	Srv_serror_rate
2	Protocol_type	16	Num_root	30	Srv_rerror_rate
3	Service	17	Num_file_creations	31	Srv_diff_host_rate
4	Src_bytes	18	Num_shells	32	Dst_host_count
5	Dst_bytes	19	Num_access_files	33	Dst_host_srv_count
6	Flag	20	Num_outbound_cmds	34	Dst_host_same_srv_rate
7	Land	21	Is_hot_login	35	Dst_host_diff_srv_rate
8	Wrong_fragment	22	Is_guest_login	36	Dst_host_same_src_port_rate
9	Urgent	23	Count	37	Dst_host_srv_diff_host_rate
10	Hot	24	Serror_rate	38	Dst_host_serror_rate
11	Num_failed_logins	25	Rerror_rate	39	Dst_host_srv_serror_rate
12	Logged_in	26	Same_srv_rate	40	Dst_host_rerror_rate
13	Num_compromised	27	Diff_srv_rate	41	Dst_host_srv_rerror_rate
14	Root_shell	28	Srv_count	42	class

C. EVALUATION MEASURES

In the proposed method, we use accuracy, sensitivity, and specificity rate for intrusion detection. True positive (TP), False positive (FP), True negative (TN), and False negative (FN) are needed for these calculations. To this end, equations (12-14) are used to determine classification error and intrusion detection [51]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

$$Specificity = \frac{TN}{TN+FP} \tag{13}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{14}$$

In addition to these measures, average error-index, and squared error of the normal to abnormal traffic detection error can be applied, as shown in equation (15) and (16), respectively:

$$mse = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{15}$$

$$rmse = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{16}$$

D. IMPLEMENTATION PARAMETERS

In Table III, we highlight several parameters that are critical to the simulation of the proposed method. According to GOA, each of these parameters are initially assigned a value. The implementations are presented according to these values:

TABLE III
PARAMETERS OF OPTIMIZATION ALGORITHM

Parameters	Value
Population size of grasshoppers	10
Repeat size of the proposed algorithm	30
C_{max} in the grasshopper Algorithm	1
C_{min} in the grasshopper Algorithm	0.004
F index in the grasshopper algorithm	0.5
L index in the grasshopper algorithm	1.5
Number of hidden layers of neural network	2
Number of hidden layer neurons	5

E. ANALYSIS

In Fig.7, an example of the output of our method for the NIDS in MATLAB is shown. In this example, the output has a replication size of 20, and the initial population is considered with sizes of 5, 10, 15, and 20. NIDS error rate is shown in this context. According to the output, it is evident that the error reduction process is descending, which elucidates a reason for selecting weights and biases by way of algorithm iteration. The examination based on the mean error-index displays that this process varies from one experiment to another due to the pseudo-random behavior of GOA and metaheuristic algorithms.

This approach seeks to make an ANN more efficient and intelligent for detecting intrusion. For this purpose, the proficiency of swarm-based grasshoppers is combined with the facility of ANN-based learning to cut down intrusion detection error in the network.

GOA reduces spam detection error by repetition rendering optimal choice of weight and bias for ANN. Output analysis of the proposed method shows that despite the initial population size, error reduction of the NIDS through algorithm repetition is a meaningful reduction process. Thus, we can conclude the following:

- Increasing grasshopper population size will decrease the NIDS.
- Error reduction in the form of repetition facilitates timely convergence of weights and biases to an optimum amount.

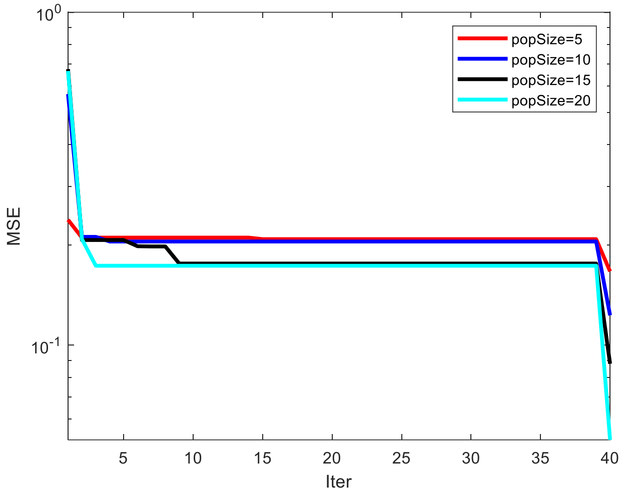

Figure 7. MSE Error reduction in the NIDS with an initial population of 5, 10, 15, 20, and 40 iterations

Figure 7 shows the intrusion detection error rate in terms of the algorithm iteration in different populations. The analysis shows that by increasing the weight and bias vectors' population size, the chances of finding the optimal weight and bias vector increases in ANN. Moreover, increasing the population reduces the intrusion detection error in the proposed method.

The results of 30 different experiments with initial population variables of 5, 10, 15, 20, and 30 iterations are illustrated in Fig.8; this demonstrates that population size in reducing the NIDS is excellent.

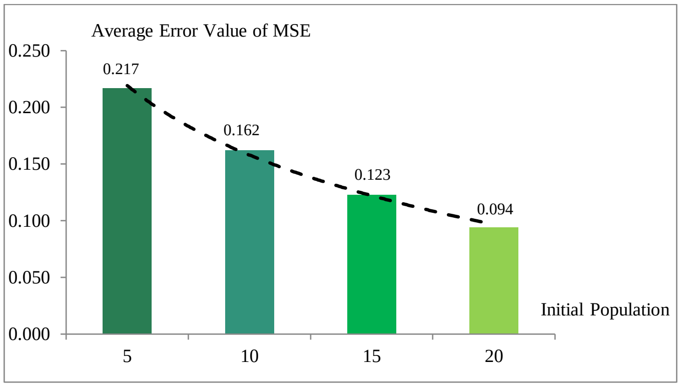

Figure 8. Error reduction in the NIDS with increasing population size

The experiments and subsequent analysis exhibit that with an increase in population size from 5 to 20. The MSE error of ID reduces from 0.217 to 0.094 - an error reduction of about 2.31 fold. This reduction in error is underscored by the heightened probability of identifying optimal weight and bias as population size increases. When the proposed method is implemented for 30 different experiments with a population size of 15 and 20 iterations, mean sensitivity, specificity, and accuracy index for detecting network intrusion are 93.17%, 89.25%, and 95.41%, respectively [52]. According to the diagram in Fig.9, the three indices mentioned earlier are consistent with better performance relative to SVM techniques, decision tree (DT) combined with PSO algorithm, and SVM in conjunction with particle algorithm [53].

A simple way (approach 1) to compute accuracy and other ANN parameters is to consider the GOA's population equal to 1 and the number of iterations equal to 0. This status is performed only in the ANN. An alternative method is to implement the ANN only in the MATLAB environment (approach 2). We have chosen the second approach to calculate accuracy, specificity, and sensitivity indexes associated with the ANN. Evaluations show that our method (GOAMLP) outperforms the basic method (MLP in ANN) in accuracy, specificity, and sensitivity of intrusion detection by 4.77%, 6.89%, 7.49%, respectively.

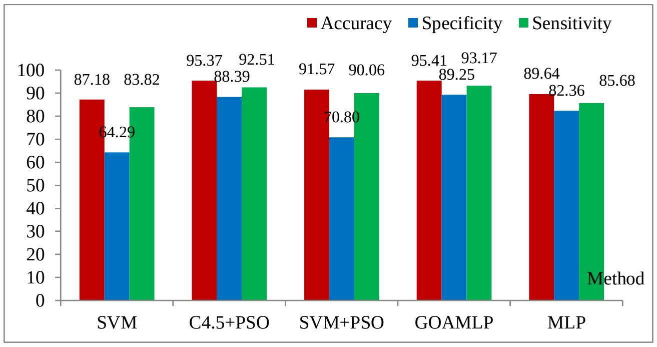

Figure 9. Comparison of Proposed Method in Intrusion Detection with Research [52]

In Fig.10, we compare the accuracy of our method with Binary Particle Swarm Optimization (BPSO), Binary Bat Algorithm with Lévy Flights (BBAL), Binary Firefly Algorithm (BFA) and Naïve Bayesian Classifier (NBC) [53], [54]:

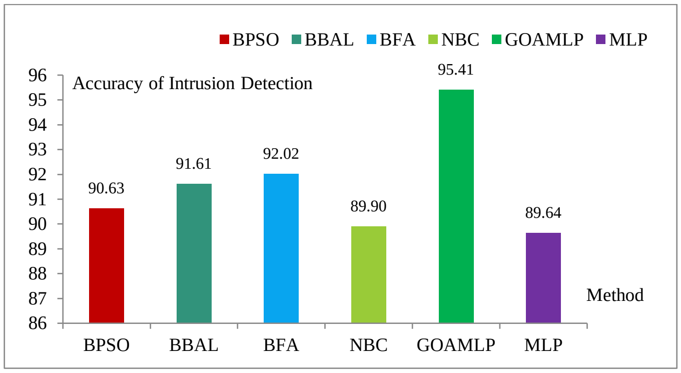

Figure 10. Comparison of Proposed Method in Intrusion Detection with Research [53], [54]

This evaluation shows that the accuracy, sensitivity, and specificity index of our method surpasses SVM and DT techniques in combination with PSO for accurate detection of intrusion. Moreover, it is more accurate in detecting network intrusion than BPSO, BBAL, BFA, and NBC.

Tables IV and V, show a comparison between embedded learning methods of ANN and metaheuristic algorithms such as Butterfly optimization (BOA) (2019), Harris hawks optimization (HHO) (2019), and Black widow optimization (BOW) (2020) algorithms based on KDD and UNSW-NB15 datasets in IDS. In this comparison, two embedded learning methods such as random forest (RF) and XGBoost are also utilized:

TABLE IV
COMPARISON BETWEEN GOAMLP WITH EMBEDDED LEARNING METHODS BASED ON THE KDD DATASET

Method	Accuracy	Specificity	Sensitivity
BOA	%93.82	%93.29	%93.54
HHO	%94.27	%94.00	%94.02
BWO	%94.66	%94.26	%94.33
Random Forest	%94.23	%93.92	%94.19
XGBoost	%95.15	%94.82	%94.88
Proposed Method	%95.41	%93.17	%89.25

TABLE V
COMPARISON BETWEEN GOAMLP WITH EMBEDDED LEARNING METHODS BASED ON THE UNSW DATASET

Method	Accuracy	Specificity	Sensitivity
BOA	%97.83	%97.44	%97.58
HHO	%98.02	%97.77	%97.85
BWO	%98.19	%98.07	%98.12
Random Forest	%97.82	%97.59	%97.63
XGBoost	%98.33	%97.66	%98.08
Proposed Method	%98.88	%98.09	%98.14

Experimental analysis shows that the proposed method has more accuracy, sensitivity, and specificity than embedded learning methods based on KDD and UNSW datasets. Among the embedded methods, after the proposed method, XGBoost is in second place in terms of accuracy in detecting network intrusion. The proposed method also has the highest specificity, while BWO is in the second place among metaheuristic methods.

A significant criterion in IDS is that the method has appropriate operating speed for analyzing network traffic. A noteworthy advantage of our proposed method is that the IDS was initially trained using small network traffic. We brought forth feature selection using the GOAMLP algorithm and ANN. The substantial features were then extracted and placed as a filter on the network traffic to reduce traffic volume. In our implementations, the number of features was reduced from 41 input features to 17 features. The traffic volume was thus reduced by a factor of ~2.41. Therefore, the proposed method ultimately requires less time than ANN because it trains fewer features. Due to our use of GOA for optimal selection of weights and biases, some computational load is created. Still, in turn, this makes the proposed method nearly 2.41 times faster than the ANN.

V. CONCLUSION AND FUTURE WORKS

In this paper, a multilayer ANN and the GOA algorithm are proposed to design an IDS algorithm that reduces the amount of ANN output error in intrusion detection. Our implementation results show that three indicators of the proposed method are more accurate than state-of-the-art, such as RF, XGBoost, and embedded learning of ANN with BOA, HHO, and BWO algorithms in the KDD and UNSW databases. GOAMLP also outperforms the MLP and other IDS methods. In future research, due to the progressively growing IoT and increasing security challenges, we intend to provide a new IDS for the IoT infrastructure. One of our future works provides an improved and binary version of metaheuristic algorithms, such as the GOA algorithm, to reduce traffic mass using feature selection (FS). Besides, in future research, deep learning neural networks will be used along with FS to detect intrusion.

REFERENCES

[1] Y. Ai, M. Cheffena, T. Ohtsuki, and H. Zhuang, "Secrecy Performance Analysis of Wireless Sensor Networks," IEEE Sensors Lett., vol. 3, no. 5, pp. 1–4, May 2019, doi: 10.1109/LSENS.2019.2909323.

[2] H. Zhou, H. Wang, X. Chen, X. Li, and S. Xu, "Data Offloading Techniques Through Vehicular Ad Hoc Networks: A Survey," IEEE Access, vol. 6, pp. 65250–65259, 2018, doi: 10.1109/ACCESS.2018.2878552.

[3] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things," IEEE Internet Things J., vol. 6, no. 4, pp. 6822–6834, Aug. 2019, doi: 10.1109/JIOT.2019.2912022.

[4] K. Kang, I. Lee, K. Liu, M. K. Yoon, and K. J. Park, "Guest Editorial Special Issue on RRCPS: Reliable and Resilient Cyber-Physical Systems," IEEE Internet Things J., vol. 6, no. 4, pp. 6271–6275, Aug. 2019, doi: 10.1109/JIOT.2019.2926610.

[5] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An Intelligent Anomaly Based Intrusion Detection System for IoT Edge Devices," IEEE Internet Things J., pp. 1–1, 2020, doi: 10.1109/JIOT.2020.2970501.

[6] B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A. Mohammed, and W. M. Abduallah, "Comprehensive Review of Artificial Intelligence and Statistical Approaches in Distributed Denial of Service Attack and Defense Methods," IEEE Access, vol. 7, pp. 51691–51713, 2019, doi: 10.1109/ACCESS.2019.2908998.

[7] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, no. Itnec, pp. 1602–1606, doi: 10.1109/ITNEC.2019.8729425.

[8] S. Nag, S. Banerjee, and S. Sen, "A New Three Party Authenticated Key Agreement Protocol which is Defiant towards Password Guessing Attack," in 2019 International Conference on Automation, Computational and Technology Management, ICACTM 2019, 2019, pp. 13–18, doi: 10.1109/ICACTM.2019.8776744.

[9] I. Dutt, S. Borah, and I. K. Maitra, "Multiple Immune-based Approaches for Network Traffic Analysis," Procedia Comput. Sci., vol. 167, no. 2019, pp.

2111–2123, 2020, doi: 10.1016/j.procs.2020.03.259.

[10] A. Kazim et al., “Memory Forensics: Recovering Chat Messages and Encryption Master Key,” 2019 10th Int. Conf. Inf. Commun. Syst., pp. 58–64, Jun. 2019, doi: 10.1109/IACS.2019.8809179.

[11] A. Kaur, S. K. Pal, and A. P. Singh, “Hybridization of K-Means and Firefly Algorithm for intrusion detection system,” Int. J. Syst. Assur. Eng. Manag., vol. 9, no. 4, pp. 901–910, Aug. 2018, doi: 10.1007/s13198-017-0683-8.

[12] W. Wu et al., “A survey of intrusion detection for in-vehicle networks,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 919–933, Mar. 2020, doi: 10.1109/TITS.2019.2908074.

[13] M. Chen, N. Wang, H. Zhou, and Y. Chen, “FCM technique for efficient intrusion detection system for wireless networks in cloud environment,” Comput. Electr. Eng., vol. 71, pp. 978–987, Oct. 2018, doi: 10.1016/j.compeleceng.2017.10.011.

[14] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, “Building an Effective Intrusion Detection System by Using Hybrid Data Optimization Based on Machine Learning Algorithms,” Secur. Commun. Networks, vol. 2019, pp. 1–11, Jun. 2019, doi: 10.1155/2019/7130868.

[15] B. Badhon, M. M. J. Kabir, S. Xu, and M. Kabir, “A survey on association rule mining based on evolutionary algorithms,” Int. J. Comput. Appl., vol. 0, no. 0, pp. 1–11, May 2019, doi: 10.1080/1206212X.2019.1612993.

[16] P. Tao, Z. Z. Sun, and Z. Z. Sun, “An Improved Intrusion Detection Algorithm Based on GA and SVM,” IEEE Access, vol. 6, pp. 13624–13631, 2018, doi: 10.1109/ACCESS.2018.2810198.

[17] M. M. Patel and P. K. Patel, “Intrusion Detection System Based on Trust Value in Wireless Sensor Networks,” 2019 3rd Int. Conf. Electron. Commun. Aerosp. Technol., pp. 618–620, Jun. 2019, doi: 10.1109/ICECA.2019.8822081.

[18] W. Wang, L. Ren, L. Chen, and Y. Ding, “Intrusion detection and security calculation in industrial cloud storage based on an improved dynamic immune algorithm,” Inf. Sci. (Ny)., vol. 501, pp. 543–557, Oct. 2019, doi: 10.1016/j.ins.2018.06.072.

[19] T. W. Liao and G. Li, “Metaheuristic-based inverse design of materials – A survey,” J. Mater., vol. 6, no. 2, pp. 414–430, Jun. 2020, doi: 10.1016/j.jmat.2020.02.011.

[20] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications,” Futur. Gener. Comput. Syst., vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[21] S. Saremi, S. Mirjalili, S. Mirjalili, and J. Song Dong, “Grasshopper Optimization Algorithm: Theory, Literature Review, and Application in Hand Posture Estimation,” in Studies in Computational Intelligence, vol. 811, Cham: Springer International Publishing, 2020, pp. 107–122.

[22] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper Optimisation Algorithm: Theory and application,” Adv. Eng. Softw., vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.

[23] X. Yue, H. Zhang, and H. Yu, “A Hybrid Grasshopper Optimization Algorithm with Invasive Weed for Global Optimization,” IEEE Access, vol. 8, pp. 5928–5960, 2020, doi: 10.1109/ACCESS.2019.2963679.

[24] C. Chen, W. He, H. Zhou, Y. Xue, and M. Zhu, “A comparative study among machine learning and numerical models for simulating groundwater dynamics in the Heihe River Basin, northwestern China,” Sci. Rep., vol. 10, no. 1, pp. 1–13, Dec. 2020, doi: 10.1038/s41598-020-60698-9.

[25] F.-Y. Leu, K.-L. Tsai, and S.-Y. Lin, “E-ANDSF-Based Base Station Selection Scheme by Using MLP in Untrusted Environments,” IEEE Trans. Ind. Informatics, vol. 15, no. 10, pp. 5708–5714, Oct. 2019, doi: 10.1109/TII.2019.2916335.

[26] C. Cai, S. Mei, and W. Zhong, “Configuration of intrusion prevention systems based on a legal user: the case for using intrusion prevention systems instead of intrusion detection systems,” Inf. Technol. Manag., vol. 20, no. 2, pp. 55–71, Jun. 2019, doi: 10.1007/s10799-018-0291-6.

[27] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[28] S. Elhag, A. Fernández, A. Altalhi, S. Alshomrani, and F. Herrera, “A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems,” Soft Comput., vol. 23, no. 4, pp. 1321–1336, Feb. 2019, doi: 10.1007/s00500-017-2856-4.

[29] I. Benmessahel, K. Xie, M. Chellal, and T. Semong, “A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization,” Evol. Intell., vol. 12, no. 2, pp. 131–146, Jun. 2019, doi: 10.1007/s12065-019-00199-5.

[30] E. Kabir, J. Hu, H. Wang, and G. Zhuo, “A novel statistical technique for intrusion detection systems,” Futur. Gener. Comput. Syst., vol. 79, pp. 303–318, Feb. 2018, doi: 10.1016/j.future.2017.01.029.

[31] S. A. R. Shah and B. Issac, “Performance comparison of intrusion detection systems and application of machine learning to Snort system,” Futur. Gener. Comput. Syst., vol. 80, pp. 157–170, Mar. 2018, doi: 10.1016/j.future.2017.10.016.

[32] T. Hamed, R. Dara, and S. C. Kremer, “Network intrusion detection system based on recursive feature addition and bigram technique,” Comput. Secur., vol. 73, pp. 137–155, Mar. 2018, doi: 10.1016/j.cose.2017.10.011.

[33] V. Kumar, A. K. Das, and D. Sinha, “UIDS: a unified intrusion detection system for IoT environment,” Evol. Intell., no. 0123456789, pp. 1–13, Sep. 2019, doi: 10.1007/s12065-019-00291-w.

[34] S. Venkatraman and B. Surendiran, “Adaptive hybrid intrusion detection system for crowd sourced multimedia internet of things systems,” Multimed. Tools Appl., vol. 79, no. 5–6, pp. 3993–4010, Feb. 2020, doi: 10.1007/s11042-019-7495-6.

[35] M. M. Salim, S. Rathore, and J. H. Park, “Distributed denial of service attacks and its defenses in IoT: a survey,” J. Supercomput., vol. 76, no. 7, pp. 5320–5363, Jul. 2020, doi: 10.1007/s11227-019-02945-z.

[36] A. Al Shorman, H. Faris, and I. Aljarah, “Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection,” J. Ambient Intell. Humaniz. Comput., vol. 11, no. 7, pp. 2809–2825, Jul. 2020, doi: 10.1007/s12652-019-01387-y.

[37] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, “Survey on SDN based network intrusion detection system using machine learning approaches,” Peer-to-Peer Netw. Appl., vol. 12, no. 2, pp. 493–501, Mar. 2019, doi: 10.1007/s12083-017-0630-0.

[38] K. Al Jallad, M. Aljnidi, and M. S. Desouki, “Big data analysis and distributed deep learning for next-generation intrusion detection system optimization,” J. Big Data, vol. 6, no. 1, p. 88, Dec. 2019, doi: 10.1186/s40537-019-0248-6.

[39] A. Alnafessah and G. Casale, “Artificial neural networks based techniques for anomaly detection in Apache Spark,” Cluster Comput., vol. 23, no. 2, pp. 1345–1360, Jun. 2019, doi: 10.1007/s10586-019-02998-y.

[40] S. Niksefat, P. Kaghazgaran, and B. Sadeghiyan, “Privacy issues in intrusion detection systems: A taxonomy, survey and future directions,” Comput. Sci. Rev., vol. 25, pp. 69–78, Aug. 2017, doi: 10.1016/j.cosrev.2017.07.001.

[41] M. S. Mohd Pozi et al., “Improving Anomalous Rare Attack Detection Rate for Intrusion Detection System Using Support Vector Machine and Genetic Programming,” Neural Process. Lett., vol. 44, no. 2, pp. 279–290, Oct. 2016, doi: 10.1007/s11063-015-9457-y.

[42] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, “A Supervised Intrusion Detection System for Smart Home IoT Devices,” IEEE Internet Things J., vol. 6, no. 5, pp. 9042–9053, Oct. 2019, doi: 10.1109/JIOT.2019.2926365.

[43] B. Naik, M. S. Obaidat, J. Nayak, D. Pelusi, P. Vijayakumar, and S. H. Islam, “Intelligent Secure Ecosystem Based on Metaheuristic and Functional Link Neural Network for Edge of Things,” IEEE Trans. Ind. Informatics, vol. 16, no. 3, pp. 1947–1956, Mar. 2020, doi: 10.1109/TII.2019.2920831.

[44] F. Farivar, M. S. Haghighi, A. Jolfaei, and M. Alazab, “Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT,” IEEE Trans. Ind. Informatics, vol. 16, no. 4, pp. 2716–2725, Apr. 2020, doi: 10.1109/TII.2019.2956474.

[45] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, “An Industrial Network Intrusion Detection Algorithm Based on Multifeature Data Clustering Optimization Model,” IEEE Trans. Ind. Informatics, vol. 16, no. 3, pp. 2063–2071, Mar. 2020, doi: 10.1109/TII.2019.2946791.

[46] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, “An efficient hybrid multilayer perceptron neural network with grasshopper optimization,” Soft Comput., vol. 23, no. 17, pp. 7941–7958, 2019.

[47] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A.-Z. Ala’M, and S. Mirjalili, “Binary grasshopper optimisation algorithm approaches for feature selection problems,” Expert Syst. Appl., vol. 117, pp. 267–286, 2019.

[48] K. Siddique et al., “KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research,” Computer (Long.

Beach. Calif)., vol. 52, no. 2, pp. 41–51, Feb. 2019, doi: 10.1109/MC.2018.2888764.

[49] S. El-Sappagh, A. S. Mohammed, and T. A. AlSheshtawy, “Classification Procedures for Intrusion Detection Based on Kdd Cup 99 Data Set,” Int. J. Netw. Secur. Its Appl., vol. 11, no. 03, pp. 21–29, May 2019, doi: 10.5121/ijnsa.2019.11302.

[50] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, “Intrusion detection model using machine learning algorithm on Big Data environment,” J. Big Data, vol. 5, no. 1, p. 34, Dec. 2018, doi: 10.1186/s40537-018-0145-4.

[51] S. Otoum, B. Kantarci, and H. Mouftah, “Empowering Reinforcement Learning on Big Sensed Data for Intrusion Detection,” in IEEE International Conference on Communications, 2019, vol. 2019-May, pp. 1–7, doi: 10.1109/ICC.2019.8761575.

[52] G. V. V. Nadiammai and M. Hemalatha, “Effective approach toward Intrusion Detection System using data mining techniques,” Egypt. Informatics J., vol. 15, no. 1, pp. 37–50, Mar. 2014, doi: 10.1016/j.eij.2013.10.003.

[53] S. Anwar et al., “From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions,” Algorithms, vol. 10, no. 2, p. 39, Mar. 2017, doi: 10.3390/a10020039.

[54] R. F. Najeeb and B. N. Dhannoon, “A feature selection approach using binary Firefly Algorithm for network intrusion detection system,” ARPN J. Eng. Appl. Sci., vol. 13, no. 6, pp. 2347–2352, 2018.

Shadi Moghanian received her BS from the University of Omran & Tosseh and MS from Azad University of science and research. She worked in the industry for several years, including the food industry as a software engineer. Miss Moghanian has more than 10 years of experience in research and industry. She joined Tobacco Company (IT) in Fall 2016. Her research interests include machine learning, artificial intelligence, neural networks, and data mining. In recent years, she has been studying evolutionary and heuristic algorithms. She has been a Microsoft Windows server specialist and administrator. Her current research focuses on business intelligence development and detection of phishing pages.

Farshid Bagheri Saravi has earned an MS in Electrical and Computer Engineering with a concentration in telecommunications. He is currently a Research Consultant and Network Administrator in CS-IT Hub. He has published extensively in refereed journals and conferences. He is also a reviewer of several Q1 journals such as IEEE Internet of Things (IoT), IEEE Access, IEEE Systems (ISJ), and Elsevier Expert Systems with Applications (ESWA) Journals. His current research interests include Wireless Communications (WBAN, WSN, 6G), Metaheuristic algorithms (Swarm-based), IoT, Blockchain Business (3D Printing, Healthcare, Society, Cryptocurrency), Artificial Intelligence (Machine Learning, Deep Learning), Cloud Services and Big Data. He is certified in MCP, MCTS, MCSA, and CCNA routing and switching. He is an IEEE member.

Giti Javidi earned her BS from University of Central Oklahoma and MS and PhD from University of South Florida. Her main research area includes Human Computer Interaction, Human-Centric Cybersecurity, Data Mining, Visualization and STEM Education. She has been involved in teaching Computer Science, Information Technology, and Cybersecurity. Professor Javidi’s extensive list of publications and research grants include projects with NASA, NSF, MSIP, Google, NCWIT and other industry partners. Currently she serves as a Professor of Information Assurance and Cybersecurity Management at the University of South Florida. She is an IEEE member.

Ehsan Sheybani has earned his BS, MS, and PhD in Electrical Engineering from UF, FSU, and USF respectively. His main research area has been applications of communication, signal processing, and data analysis. He has been involved in teaching, practicing, researching, and consulting applications of DSP in technology, system analysis, and data sciences for the past 20 years. He has a long list of publications and research grants including projects with NASA, NSF, NIH, DoD, DEd, and industry partners. Currently he serves as a faculty at the University of South Florida. He is a senior member of IEEE.