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

SHADI MOGHANIAN1, (Member, IEEE), FARSHID BAGHERI SARAVI2, (Member, IEEE), GITI JAVIDI3, (Member, IEEE), AND EHSAN O. SHEYBANI3, (Senior Member, IEEE)

1Computer Science Department, Universidad Politécnica de Cataluña, 08034 Barcelona, Spain
2CS-IT Hub, Bradenton, FL 34212-3609, USA
3Muma College of Business, University of South Florida, Tampa, FL 33620-5500, USA
Corresponding author: Ehsan O. Sheybani (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.

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
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 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 farmlands 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]:

\[ X_i = S_i + G_i + A_i \]  

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 \( s(\text{d}_ij) \) 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]:

\[ s(\text{d}_ij) = f e^{-\frac{r}{\text{d}_ij}} - e^{-r} \]  

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 \( s(\text{d}_ij) \).

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

\[ \hat{X}_i = c \left[ \sum_{j=1}^{N} \frac{ub_d - lb_d}{2} \left| \frac{X_j - X_i}{d_{ij}} \right| \right] + \hat{T} \]  

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:

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 1, we compare algorithms, methods, challenges, and advantages.

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
TABLE 1. 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 (FRA) 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 |
| Shooshtari 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 |

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 modeling the GOAMLP algorithm are discussed. Finally, a flowchart illustrating the proposed method for detecting network intrusion is presented.
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 iteration.
- 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:

### 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:

The preprocessing phase of the method in this paper contains two sections. First, the data is normalized and reduced

![FIGURE 3. The structure of an ANN [24.]](image-url)
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:

\[
f' = \frac{f - \min}{\max - \min} (b-a) + a
\]  

(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):

\[
f' = \frac{f - \min}{\max - \min}
\]  

(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:

\[
G_{OA} = [F_1, F_2, F_3, \ldots, F_D]
\]  

(6)

\(G_i\) is a grasshopper or a feature vector, \(F_j\) is the \(j\) – 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, \(G_{OA}\) 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]:

\[
Cost = \alpha \cdot E + \beta \cdot \frac{f}{F}
\]  

(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, \(\alpha\) is a random number between 0 and 1, and \(\beta\) 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...
S. Moghanian et al.: GOAMLP: Network Intrusion Detection

FIGURE 6. Flowchart of the proposed method for the NIDS.

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

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:

\[ Pop(i) = \{w_1, w_2, \ldots, w_m, b_1, b_2, \ldots, b_k\} \]  

(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):

\[ Pop = \{Pop(1), Pop(2), \ldots, Pop(n)\} \]  

(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):

\[ e = \frac{1}{n} \left( \sum_{i=1}^{n} (\tilde{o}_i - o_i)^2 \right) \]  

(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:

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

(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.

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.
Considered numerical, and the rest are non-numerical. There are 34 features, of which 34 are considered numerical, and the rest are non-numerical. There are four types 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 2.

**TABLE 2. KDD dataset attributes [50].**

| No | Attribute name | No | Attribute name | No | Attribute name |
|----|----------------|----|----------------|----|----------------|
| 1  | Duration       | 15 | Be_ignored     | 29 | Serv.recon      |
| 2  | Protocol_type  | 16 | Num_root       | 30 | Serv. perform   |
| 3  | Service        | 17 | Num_file_changes| 31 | Serv. reorg     |
| 4  | Src_bytes     | 18 | Num_shells     | 32 | Disc. cache     |
| 5  | Dist_bytes    | 19 | Num_access_files| 33 | Disc. host count|
| 6  | Flag           | 20 | Num_outbound   | 34 | Disc. host count|
| 7  | Land           | 21 | Is_host_login  | 35 | Disc. host count|
| 8  | Wrong_fragment | 22 | Is_guest_login | 36 | Disc. host count|
| 9  | Urgent         | 23 | Count          | 37 | Disc. host count|
| 10 | Hot            | 24 | Serv. rate     | 38 | Disc. host count|
| 11 | Num_failed_logins | 25 | Retcode rate   | 39 | Disc. host count|
| 12 | Logged_in      | 26 | Same_srv_rate  | 40 | Disc. host count|
| 13 | Num_compromised| 27 | Diff_srv_rate  | 41 | Disc. host count|
| 14 | Root_shell     | 28 | Serv. count    | 42 | class           |

The KDD dataset has 41 features, of which 34 are considered numerical, and the rest are non-numerical. There are four types 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 2.

**TABLE 2. KDD dataset attributes [50].**

| No | Attribute name | No | Attribute name | No | Attribute name |
|----|----------------|----|----------------|----|----------------|
| 1  | Duration       | 15 | Be_ignored     | 29 | Serv.recon      |
| 2  | Protocol_type  | 16 | Num_root       | 30 | Serv. perform   |
| 3  | Service        | 17 | Num_file_changes| 31 | Serv. reorg     |
| 4  | Src_bytes     | 18 | Num_shells     | 32 | Disc. cache     |
| 5  | Dist_bytes    | 19 | Num_access_files| 33 | Disc. host count|
| 6  | Flag           | 20 | Num_outbound   | 34 | Disc. host count|
| 7  | Land           | 21 | Is_host_login  | 35 | Disc. host count|
| 8  | Wrong_fragment | 22 | Is_guest_login | 36 | Disc. host count|
| 9  | Urgent         | 23 | Count          | 37 | Disc. host count|
| 10 | Hot            | 24 | Serv. rate     | 38 | Disc. host count|
| 11 | Num_failed_logins | 25 | Retcode rate   | 39 | Disc. host count|
| 12 | Logged_in      | 26 | Same_srv_rate  | 40 | Disc. host count|
| 13 | Num_compromised| 27 | Diff_srv_rate  | 41 | Disc. host count|
| 14 | Root_shell     | 28 | Serv. count    | 42 | class           |

The KDD dataset has 41 features, of which 34 are considered numerical, and the rest are non-numerical. There are four types 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 2.

**TABLE 2. KDD dataset attributes [50].**

| No | Attribute name | No | Attribute name | No | Attribute name |
|----|----------------|----|----------------|----|----------------|
| 1  | Duration       | 15 | Be_ignored     | 29 | Serv.recon      |
| 2  | Protocol_type  | 16 | Num_root       | 30 | Serv. perform   |
| 3  | Service        | 17 | Num_file_changes| 31 | Serv. reorg     |
| 4  | Src_bytes     | 18 | Num_shells     | 32 | Disc. cache     |
| 5  | Dist_bytes    | 19 | Num_access_files| 33 | Disc. host count|
| 6  | Flag           | 20 | Num_outbound   | 34 | Disc. host count|
| 7  | Land           | 21 | Is_host_login  | 35 | Disc. host count|
| 8  | Wrong_fragment | 22 | Is_guest_login | 36 | Disc. host count|
| 9  | Urgent         | 23 | Count          | 37 | Disc. host count|
| 10 | Hot            | 24 | Serv. rate     | 38 | Disc. host count|
| 11 | Num_failed_logins | 25 | Retcode rate   | 39 | Disc. host count|
| 12 | Logged_in      | 26 | Same_srv_rate  | 40 | Disc. host count|
| 13 | Num_compromised| 27 | Diff_srv_rate  | 41 | Disc. host count|
| 14 | Root_shell     | 28 | Serv. count    | 42 | class           |
TABLE 3. Parameters of optimization algorithm.

| Parameters                                         | Value |
|----------------------------------------------------|-------|
| Population size of grasshoppers                    | 10    |
| Repeat size of the proposed algorithm              | 30    |
| $C_{\text{max}}$ in the grasshopper Algorithm      | 1     |
| $C_{\text{min}}$ 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     |

D. IMPLEMENTATION PARAMETERS

In Table 3, 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:

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 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.

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) outper-
forms the basic method (MLP in ANN) in accuracy, specificity, and sensitivity of intrusion detection by 4.77%, 6.89%, and 7.49%, respectively.

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]: 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 4 and 5, 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 (BWO) (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:

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, May 2019, Art. no. 7500304, doi: 10.1109/LSENS.2019.2909523.
[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. 65290–65299, 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 RCRPS: 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., vol. 7, no. 8, pp. 6882–6897, Aug. 2020, doi: 10.1109/JIOT.2020.2970501.

[6] B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A. Mohammed, and M. M. Patel and P. K. Patel, “Intrusion detection system based on trust posture estimation,” in Optimization algorithm: Theory, literature review, and application in hand, vol. 97, pp. 849–872, Aug. 2019, Future Gener. Comput. Syst. doi: 10.1016/j.future.2017.10.016.

[VOLUME 8, 2020]

B. Badhon, M. M. J. Kabir, S. Xu, and M. Kabir, “A survey on association studies in computational intelligence,” vol. 8, no. 2, pp. 51691–51713, 2019, doi: 10.1109/ACCESS.2019.2908074.

[8] H. Wang and B. Wu, “SDN-based hybrid honeypot for attack capture,” in Proc. IEEE 3rd Int. Conf. Netw., Electron. Autom. Control Conf. (ITNec), Mar. 2019, pp. 1602–1606, doi: 10.1109/ITNec.2019.8729425.

[9] S. Nag, S. Banerjee, and S. Sen, “A new three party authenticated key agreement protocol which is defiant towards password guessing attack,” in Proc. Int. Conf. Autom., Comput. Technol. Manage. (ICACTM), Apr. 2019, pp. 13–18, doi: 10.1109/ICACTM.2019.8776744.

[10] I. Dutt, S. Borah, and I. K. Maitra, “Multiple immune-based approaches for network traffic analysis,” Procedia Comput. Sci., vol. 167, pp. 2111–2123, Jan. 2020, doi: 10.1016/j.procs.2020.03.259.

[11] A. Kazim, F. Almæenei, S. A. Ali, F. Iqbal, and K. Al-Hassaei, “Memory forensics: Recovering chat messages and encryption master key,” in Proc. 10th Int. Conf. Inf. Commun. Syst. (ICICS), Jun. 2019, pp. 58–64, doi:10.1109/ICACS.2019.8890719.

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

[13] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, “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.

[14] 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.

[15] 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. Netw., vol. 2019, pp. 1–11, Jun. 2019, doi:10.1155/2019/7130868.

[16] 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., pp. 1–11, May 2019, doi:10.120621X.2019.1612993.

[17] P. Tao, Z. Sun, and 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.

[18] M. M. Patel and P. K. Patel, “Intrusion detection system based on trust value in wireless sensor networks,” in Proc. 3rd Int. Conf. Electron. Commun. Aeron. Technol., Jun. 2019, pp. 618–620, doi:10.1109/ICEEA.2019.8822081.

[19] 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., vol. 501, pp. 543–557, Oct. 2019, doi:10.1016/j.ins.2018.06.072.

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

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

[22] 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, Switzerland: Springer, 2020, pp. 107–122.
SHADI MOGHANIAN (Member, IEEE) received the B.S. degree from the University College of Oman and Tosseh, and the M.S. degree from the Azad University of Science and Research. She is currently pursuing the Ph.D. degree with Universidad Politècnica de Catalunia, Barcelona, Spain. She was a Microsoft Windows Server Specialist and an Administrator. She worked in industry for several years, including the Food Industry as a Software Engineer. She has more than ten years of experience in research and industry. She joined Tobacco Company (IT), in Fall 2016. She has been studying evolutionary and heuristic algorithms. Her current research interests include machine learning, artificial intelligence, neural networks, data mining, business intelligence development, and detection of phishing pages.

FARSHID BAGHERI SARAVI (Member, IEEE) received the M.S. degree in electrical and computer engineering with a concentration in telecommunications. He is currently a Research Consultant and a Network Administrator with CS-IT Hub. He has published extensively in refereed journals and conferences. He has certified in MCP, MCTS, MCSA, and CCNA routing and switching. His current research interests include wireless communications (WBAN, WSN, and 6G), metaheuristic algorithms (swarm-based), the IoT, blockchain business (3D printing, healthcare, society, and cryptocurrency), artificial intelligence (machine learning and deep learning), cloud services, and big data. He serves as a Reviewer for several Q1 journals, such as the IEEE INTERNET OF THINGS (IoT), IEEE ACCESS, the IEEE SYSTEMS JOURNAL (ISJ), and Expert Systems With Applications (ESWA) (Elsevier) journal.

GITI JAVIDI (Member, IEEE) received the B.S. degree from the University of Central Oklahoma and the M.S. and Ph.D. degrees from the University of South Florida. She was involved in teaching computer science, information technology, and cybersecurity. She is currently a Professor of information assurance and cybersecurity management with the University of South Florida. She has extensive list of publications and research grants, including projects with NASA, NSF, MSIP, Google, NCWIT, and other industry partners. Her main research interests include human–computer interaction, human-centric cybersecurity, data mining, visualization, and STEM education.

EHSAN O. SHEYBANI (Senior Member, IEEE) received the B.S. degree in electrical engineering from UF, the M.S. degree in electrical engineering from USF, and the Ph.D. degree in electrical engineering from USF. He was involved in teaching, practicing, researching, and consulting applications of DSP in technology, system analysis, and data sciences for the past 20 years. He is currently a Faculty Member with the University of South Florida. He has a long list of publications and research grants, including projects with NASA, NSF, NIH, DoD, D.Ed., and industry partners. His main research interests include applications of communication, signal processing, and data analysis.