A Novel Anomaly-based Intrusion Detection System using Whale Optimization Algorithm WOA-Based Intrusion Detection System

Aliakbar Tajari Siahmarzkooh*
Department of Computer Sciences, Faculty of Sciences
Golestan Unversity
Gorgan, Iran
a.tajari@gu.ac.ir

Mohammad Alimardani
Department of Computer Sciences, Faculty of Sciences
Golestan Unversity
Gorgan, Iran
mohammad.alimardani.gu.ac@gmail.com

Received: 2021/09/18 Revised: 2021/11/29 Accepted: 2021/12/24

***Abstract*— The Internet has become an important part of many people's daily activities. Therefore, numerous attacks threaten Internet users. IDS is a network intrusion detection tool used to quickly identify and categorize intrusions, attacks, or security issues in network-level and host-level infrastructure. Although much research has been done to improve IDS performance, many key issues remain. IDSs need to be able to more accurately detect different types of intrusions with fewer false alarms and other challenges. In this paper, we attempt to improve the performance of IDS using Whale Optimization Algorithm (WOA). The results are compared with other algorithms. NSL-KDD dataset is used to evaluate and compare the results. K-means clustering was chosen for pre-processing after a comparison between some of the existing classifier algorithms. The proposed method has proven to be a competitive method in terms of detection rate and false alarm rate base on a comparison with some of the other existing methods.**

Keywords*— *Intrusion Detection; Whale Optimization Algorithm; NSL-KDD Dataset; K-Means Clustering.

1. INTRODUCTION

The Internet has become an important part of many people's daily activities. According to world internet usage statistics, the number of internet users has reached over 4.66 billion in 2021 [1]. Demands for a reliable security system have grown as the number of internet users increases and more critical data are being shared over the internet. Various security systems such as firewalls, antivirus, and Intrusion Detection Systems (IDS) are being used to ensure the safety of data against cyber-attacks. IDS is a network intrusion detection tool used to quickly identify and categorize intrusions, attacks, or security issues in network-level and host-level infrastructure.

Intrusion is an unwanted or malicious action that is dangerous for sensor nodes. IDS acts as an alarm or network perception when an attack occurs and prevents intruders from damaging the system by launching a notification before launching an attack. A typical IDS consists of sensors, an analysis engine and a reporting system. The sensors are located in different locations of the network or host, the main purpose of which is to collect information. The collected information is then being sent to the analysis engine, which has the ability to analyze the collected information and identify intrusions. When an intrusion is detected by the analysis engine, the system generates a notification report and sends it to the network administrator [2].

1-1. IDS Classification

An intrusion detection system is categorized based on where the IDS sensors are located: network or host. In HIDS, anti-threat programs such as firewalls, anti-virus software, and spyware detection programs are installed on any network computer that has two-way access to an off-net environment such as the Internet [3]. In NIDS, anti-threat software is only installed in specific locations, such as servers that interface between the external environment and the network portion to protect the network.

IDSs are used to detect attacks in the early stages. IDS monitors traffic and network behavior to detect suspicious activity and warns of such activities. Based on these strategies, IDS uses two main methods, signature-based and anomaly-based [4]. Signature-based IDS detects attacks by adapting network behavior to predefined patterns, so it is not effective against new malware attacks that have an unknown pattern, while anomaly-based IDS uses machine learning to create a reliable activity model and compares this model with input data. Because the anomaly-based method uses machine learning algorithms, training phase can be done according to programs and hardware configurations (See Fig.1).

1-2. Challenges of Intrusion Detection Systems

Although much research has been done to improve IDS performance, many key issues remain. IDSs need to be able to more accurately detect different types of intrusions with fewer false alarms and other challenges.

Evasion attacks are a type of attack that can occur in hostile settings when the system is operating. For example, spammers and hackers often try to prevent detection by obscuring the contents of unsolicited emails and malware code [5]. In evasion settings, malicious samples are modified during testing to prevent detection, which means to be classified as legal. No effect on training data is possible. The strength of IDS against various evasion methods still needs further research. For example, signature-based IDS in regular expressions can detect deviations caused by simple mutations such as space character

10.22133/ijwr.2022.305467.1106 20.1001.1.26454335.2021.4.2.2.0

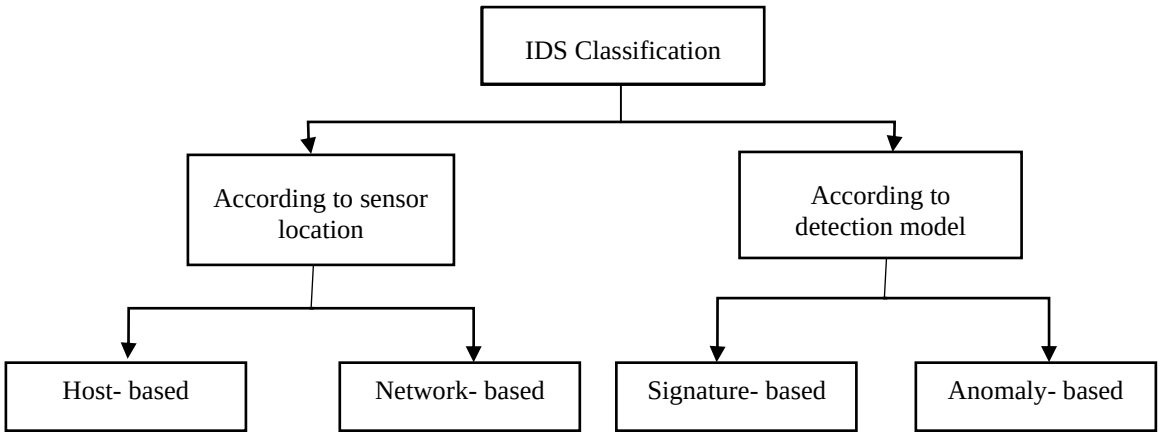

Fig. 1. IDS Classification [3]

manipulation, but it is still useless against several encryption techniques.

Distributed Denial of Service (DDoS) attack, in this type of attack, attackers use multiple sources that are infected with malware to flood the target network with overwhelming traffic. DDoS is a simple, effective, and powerful technique provided by insecure devices. Today, this is one of the most worrying areas in cybersecurity because it is very difficult to prevent [6].

2. RELATED WORKS

IDS is an important part of data security. IDS must maximize detection speed by reducing storage and computing. Various optimization techniques have been proposed by researchers to achieve high accuracy in IDS. In this section, some of the previous work and methods are discussed. We will talk about a wide range of related tasks that motivate us to devise our approach.

Sadiq et al. [7] developed an IDS based on a hybrid heuristic optimization algorithm that deals with the attraction between scattered particles in the search space. It extracts the most relevant features that can help identify network attacks accurately. These features are obtained by labeled index values that represent the information obtained from the classifier training course to be used as a basis for the developed IDS. It integrates the proposed hybrid magnetic optimization algorithm-particle swarm optimization (MOA-PSO) technique in order to improve the accuracy of artificial neural network (ANN) classifier. THE updated KDD CUP data set is formed and used during the training and testing phases.

A learning model developed for a fast-learning network (FLN) based on particle swarm optimization (PSO) has been proposed by Mohammad Hassan Ali et al. and named PSO-FLN [8]. This model is applied to the intrusion detection problem and is validated based on the famous KDD99 dataset. The proposed PSO-Based optimized fast learning network (FLN) is trained based on selecting weights using particle swarm optimization. The developed model has been compared against some meta-heuristic algorithms for training ELM, and FLN classifier. He concluded that the developed PSO-FLN has outperformed other learning approaches in the testing accuracy of the learning.

Due to the dynamic nature of the malware and the changing nature of the constantly attacking methods, the existing malware databases need to be regularly updated and evaluated. Vinayakumar et al. [9] proposed a Deep Neural Network (DNN), a kind of deep learning model, to create a flexible and effective IDS for detecting and classifying unpredictable cyber-attacks. The proposed DNN model, which was tested in KDD Cup 99, is used in other datasets such as NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS 2017 to conduct the benchmark. The proposed method is a hybrid intrusion detection alert system using a scalable framework on commodity hardware server which has the capability to analyze the network and host-level activities.

Hajisalem et al. [10] proposed a hybrid classification method based on Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms. The Fuzzy C-Means Clustering (FCM) and Correlation-based Feature Selection (CFS) techniques are applied to divide the training dataset and remove the irrelevant features, respectively. The simulation is done on NSL-KDD and UNSW-NB15 datasets. The results show that the proposed method outperforms in terms of performance metrics and can achieve 99% detection rate and 0.01% false positive rate.

An intrusion detection system based on a combination of a multilayer perceptron network and an artificial bee colony (ABC), and fuzzy clustering algorithms is proposed by Hajimirzaei et al. [11]. The CloudSim simulator and the NSL-KDD dataset were used to validate the proposed method. Training speed is improved by breaking the data set into uniform subsets. The fuzzy clustering method is used to create different training subsets. MLP is used to identify normal and abnormal packets in network traffic. NSL-KDD 99 data set features play a key role in creating the MLP structure. The ABC algorithm is used to train MLP by optimizing the weight values of the links and biases. The proposed method consists of three steps, which are training, validation, and testing.

Mazini et al. [12] proposed a hybrid method for an anomaly-based network-based IDS (A-NIDS) using an artificial bee colony (ABC) and AdaBoost algorithms to achieve a high detection rate (DR) and low false-positive rate (FPR). The ABC algorithm is used to select features and the AdaBoost algorithm is used to evaluate and classify features. The simulation results in the NSL-KDD and ISCXIDS2012 databases shows that this method differs from other IDSs with the same data set. This is shown in different scenarios based on different attacks.

In [13], Chung et al. used an intelligent dynamic swarm-based rough set (IDS-RS) for feature selection and simplified swarm optimization for intrusion data classification to create a hybrid intrusion detection system. This method is proposed to select the appropriate features that can represent the pattern of network traffic. To improve SSO classification performance, a weighted local search (WLS) method has been introduced that is included in SSO. The goal of this local search method is to discover a better solution from the neighborhood of the current solution produced by SSO. The performance of the proposed hybrid system is compared using the KDD Cup99 dataset with standard particle swarm optimization (PSO) and two other benchmark classifiers. Based on the test results, the proposed hybrid system can achieve higher classification accuracy with 93.3% than others.

Guo et al. [14] has proposed a hybrid method for achieving high detection rates with low false-positive rates in [16]. This approach is a two-level hybrid solution consisting of two components used to detect anomalies and one component to detect misuse. In the first step, an anomaly detection method with low computational complexity is developed and used to construct the detection component. The k-nearest neighbor algorithm is important in constructing the two components of the detection for step 2. In this method, all detection components are well coordinated. Step 1 detection component is involved in the construction of two detection components of step 2, which reduces the false positives and negatives produced by the step 1 detection component. The experiments were performed on the KDD99 dataset and the Kyoto University benchmark.

Singaravelan et al. [15] proposes an Internal Detection and Defense System (IIDDS) at the Call System (SC) level using data mining and forensic techniques in [17]. The main purpose of the proposed method is to identify internal intruders by increasing accuracy and reducing response time. User profiles are maintained using the Hollinger distance and compared with the actual dataset. A hash function is applied to incoming messages, and they are summarized in the sketch data set. Experimental results show that this method has better response time, intrusion accuracy, and alert accuracy.

An Intrusion Weighted Particle-based Cuckoo Search Optimization (IWP-CSO) and Hierarchical Neuron Architecture based Neural Network (HNA-NN) techniques are proposed by Shitharth et al. [16]. The main purpose of this method is to identify and classify intrusions in a Supervisory Control and Data Acquisition (SCADA) network based on the optimization. Initially, the input network data set is given as input, where attributes are arranged and clusters are initialized. Then, the features are optimized to select the best features using the proposed IWP-CSO algorithm. Finally, intrusions are classified using the proposed HNA-AA algorithm.

Liang et al. [17] used a multi-feature data clustering optimization model to propose an industrial network intrusion detection algorithm, where the weighted distances and security coefficients of data are classified based on the priority threshold of data attribute feature for each node in the network. The proposed algorithm can effectively improve the real-time detection and performance of abnormal behavior detection for multi-feature data in industrial networks. The purpose of this method is to quickly select a node with a high-security coefficient as the center of the cluster and to match the multi-feature data around the center into a cluster. The results show that the proposed algorithm performs well in terms of rate and time of detection compared to other algorithms. In the industrial network, the abnormal data detection accuracy reaches 97.8%, and the FP detection accuracy decreases by 8.8%.

Benmessahel et al. [18] has used the locust swarm optimization (LSO) algorithm to deal with the feed-forward neural network (FNN) training problems. FNN is integrated with LSO (FNN-LSO) to create an advanced detection system and improve IDS performance. This method is applied to a series of experiments to evaluate the ability and performance of the proposed approach. Experimental studies have been performed using NSL-KDD and UNSW-NB15 to test the performance of the proposed method. Particle swarm optimizer, PSO-based trainer, and genetic algorithm GA-based trainer were implemented to verify the results. The results show that the training algorithm has performed well in terms of speed convergence and reliability due to the reduction of the probability of being trapped in local minima. Also, the proposed model improves the detection rate.

In [19], Tummalapalli et al. proposed an intrusion detection framework for the cloud environment with clustering and two-level classifiers. In the first step, for the nodes clustering in the cloud, a Bayesian fuzzy clustering is used. In the next step, a two-level gravitational group search-based support vector neural network (GG-SVNN) classifier identifies intrusion in clusters. Intrusion information provided by the Level 1 classifier is organized to create compact data and given to the level 2 classifier. The Level 2 classifier eventually identifies all nodes affected by the attackers. The simulation is performed using the KDD Cup dataset. From the simulation results, it seems that the proposed GG-SVNN classifier had a good overall performance with an accuracy of 92.41% and a false alarm rate of 4.75%.

Tree pruning is a machine learning method used to reduce the size of the decision tree (DT) to reduce the complexity of the classifier and improve its prediction accuracy. Malik et al., in [20], tried to prune a DT using a particle swarm optimization (PSO) algorithm and apply it to the network intrusion detection problem. The proposed approach is a hybrid method in which PSO is used for node pruning, and pruned DT is used to classify network intrusions. The proposed approach uses single and multi-objective PSO algorithms. These experiments were performed on the popular KDD99Cup dataset. The results of the proposed method are compared with other classifiers, and it is observed that the proposed method has a good performance in terms of intrusion detection rate, false-positive velocity, accuracy and precision.

In [21], a hybrid approach to intrusion detection is provided by Samriya et al. to improve the overall security of the cloud-based computing environment. It also helps you manage a variety of security obstacles in the cloud, such as fake identity detection, data leaks, and phishing attacks, etc. to maintain security in the cloud. The proposed method uses this hybrid approach to overcome the iterative classification and selection process of the fuzzy clustering approach by automatically updating the fitness value. The SMO optimization approach leads to dimensionality, and the reduced data set is sent to a neural network. The proposed method reduces the computation time and increases accuracy.

3. PROPOSED APPROACH

In recent decades, several methods have been used to improve the performance of IDS systems. Also, some hybrid techniques have been proposed to address the deficiencies of each of these methods. In this section, we propose a new method to increase the accuracy of the intrusion detection system based on the whale optimization algorithm.

3-1. Preprocessing

Data preprocessing is a data mining method used to convert raw data into a useful and efficient format. In this paper, we attempt to analyze some of the preprocessing algorithms to be able to examine the advantages and disadvantages of these methods compared to each other and choose a suitable algorithm to complete the proposed method in this paper. The algorithms used for this purpose are AdaBoost, Extreme gradient-boosted trees, Random Forests and C4.5 Decision Tree.

To evaluate these algorithms, we used the detection rate, precision, and F1 score, which have been described in section 4.2, also the Receiver Operating Characteristic (ROC) is used to show how much model is capable of distinguishing between classes, and balanced accuracy [22]. The results are in section (4.1).

The ROC curve is the relationship between the true positive rate on the y-axis and the false positive rate on the x-axis. In essence, the higher the AUC (Area Under the Curve), the better the model is at distinguishing between different classes.

Balanced accuracy is a measure that one can use when evaluating a binary classifier. This is especially useful when classes are unbalanced, meaning that one of the two classes appears more than the other. This is shown in (1).

$$\text{Balanced Accuracy} = (TP+TN) / 2 \quad (1)$$

3-2. Whale Optimization Algorithm (WOA)

Whales are considered to be the largest mammals in the world. Whales are more commonly thought of as predators. The interesting thing about whales is that they are very intelligent and their brain is capable of experiencing a range of emotions. Humpback whales have a special hunting method. This foraging behavior is called the bubble-net feeding method. Humpback whales prefer to hunt school of krill or small fishes near the surface. It has been observed that this foraging is done by creating distinctive bubbles along a circle or '9'-shaped path.

The whale optimization algorithm (WOA) is a nature-inspired meta-heuristic algorithm, which mimics the social behavior of humpback whales (See Fig.2.). The algorithm is inspired by the bubble-net hunting strategy [23].

Researchers found two bubble-related maneuvers called "upward-spirals" and "double-loops." In the first maneuver, humpback whales dive about 12 meters down and then begin to form bubbles around the prey and swim to the top in a spiral shape. The next maneuver consists of three different stages: coral loop, lob tail, and capture loop.

Initialize the whale population X_i (i =1,2,…,n)
Calculate the fitness of each search agent
X^*= the best search agent
while *(t<maximum number of iterations)*
  for *each search agent*
  Update a, A, C, l, and p
    if1 *(p,0.5)*
      if2 *(|A|<1)*
        Update the position of the current search agent by the Eq. (6)
      else if2 *(|A|>=1)*
        Select a random search agent (X_{rand})
        Update the position of the search agent by the Eq. (13)
      end if2
    else if1 *(p>=0.5)*
      Update the position of the current search agent by the Eq. (10)
    end if1
  end for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each search agent
  Update X^* if there is a better solution
  t =t+1
end while
return X^*

Fig. 2. Pseudo-code of the WOA algorithm

Step 1: Encircling Prey

Humpback whales can detect the position of prey and encircle them. Because the optimal design position in the search space is not already known, the WO algorithm assumes that the best current candidate solution is target prey or close to optimal. After defining the best search agent, other search agents try to update their positions towards the best search agent [23]. This behavior is shown by the (2) and (3):

$$\vec{D} = |\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)| \quad (2)$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \quad (3)$$

where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, X^* is the position vector of the best solution obtained so far, X^* is the position vector, | | is the absolute value, and .is an element-by-element multiplication. It is worth mentioning here that X^* should be updated in each iteration if there is a better solution. The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows in (4) and (5):

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad (4)$$

$$\vec{C} = 2.\vec{r} \quad (5)$$

where $\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and $\vec{r}$ is a random vector in [0,1].

Step 2: Bubble-Net Attacking Method (Exploitation Phase)

This phase consists of two approaches, Shrinking encircling mechanism and Spiral updating position.

Shrinking encircling mechanism: This behavior is achieved by decreasing the value of $\vec{a}$ in (4). Note that the fluctuation range of $\vec{A}$ is also decreased by $\vec{a}$. In other words, $\vec{A}$ is a random value in the interval [−a, a] where a is decreased from 2 to 0 over the course of iterations. Setting random values for $\vec{A}$ in [−1,1], the new position of a search agent can be defined anywhere in between the original position of the agent and the position of the current best agent. (0 ≤A ≤1).

Spiral updating position: this approach first calculates the distance between the whale located at (X, Y) and prey located at (X^*, Y^*). A spiral equation is then created between the position of whale and prey to mimic the helix-shaped movement of humpback whales as follows in (6):

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) \quad (6)$$

where $\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right|$ and indicates the distance of the i-th whale to the prey (best solution obtained so far), b is a constant for defining the shape of the logarithmic spiral, l is a random number in [−1,1], and. is an element-by-element multiplication.

Note that humpback whales swim around their prey at the same time and are in a spiral path. We assume that there is a 50% chance of choosing between the shrinking encircling mechanism or the spiral model to update the position of the whales during optimization so that we can model this simultaneous behavior. The mathematical model is shown in (7):

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\vec{D} & if\ p < 0.5 \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \quad (7)$$

where p is a random number in [0,1].

Step 3: Search for Prey (Exploration Phase) In addition to the bubble-net method, the humpback whales search for prey randomly. The mathematical model of the search is as follows.

The same method based on the change of vector A can be used to search for prey (exploration). In fact, humpback whales search randomly depending on each other's position. So, we use random values greater than -1 or less than 1 for A to force the search agent to distance itself from the reference whale. Unlike the exploitation phase, we update the position of a search agent in the exploration phase based on a randomly selected search agent instead of the best search agent available so far. This mechanism and A> 1 emphasize exploration and allow the WO algorithm to search globally. The mathematical model is shown in (8) and (9):

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}} - \vec{A}.\vec{D}\right| \quad (8)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \quad (9)$$

where $\overrightarrow{X_{rand}}$ rand is a random position vector (a random whale) chosen from the current population.

4. RESULTS AND DISCUSSION

4-1. Preprocessing Algorithms Evaluation Results

In this section, we describe the results of testing our selected algorithms.

Table I shows the results of the simulation for each algorithm based on accuracy, balanced accuracy, detection rate, precision, F1 score, and ROC-AUC. In terms of accuracy, k-means clustering and AdaBoost have similar results, however, in balanced accuracy, these two algorithms cannot match the performance of others. For precision metric, AdaBoost has performed slightly better than random forests and decision trees but it has a considerably better percentage than all other algorithms.

AdaBoost also has a better number in F1 score and is the second algorithm in the ROU curve after the extreme gradient-boosted tree. However, in this case, for preprocessing we choose the algorithm with the highest accuracy, so base on Table I, k-means clustering is the algorithm for preprocessing.

4-2. The Proposed Method Results

In general, applying a suitable dataset plays an important role in validating the intrusion detection systems. There are numerous datasets such as KDD Cup 99, NSL-KDD, UNSW-NB15, ADFA, Kyoto 2006 +, and ISCX 2012 that can be applied in IDS applications. KDD Cup 99 has created in UCI Machine Learning Repository that is the most widely applied dataset for evaluation of network security solutions. NSL-KDD dataset is the modified version of the KDD Cup 99 dataset in which redundant, duplicated, noisy, and irrelevant data records have removed to solve some of the inherent problems of this dataset.

NSL-KDD (National security lab–knowledge discovery and data mining) is an advanced form of KDD99 to overcome its limitations and address its shortcomings. This is a publicly available dataset. Initially, duplicate records are removed from the training and test sets. Second, several records of the original KDD99 have been selected to achieve reliable results from classification systems. Third, the problem of unbalanced probability distribution was eliminated.

4-3. Performance Metrics

This section represents a performance analysis of the proposed method. Intrusion detection efficiency is measured by the detection rate and false alarm rate. Because the detection rate and FAR are the basic parameters that are considered to identify attacks for IDS.

In evaluation to estimate the various statistical measures, the ground truth value is required. The ground truth is composed of a set of connection records labeled either Normal or Attack in the case of binary classification. The following terms are used for determining the quality of the classification models:

TABLE I. RESULTS OF ALGORITHMS BASED ON DIFFERENT ATTACK TYPES

Attack Name / Approach	DOS	Infiltration	DDOS	Botnet	Web attack	All Types
	Accuracy					
AdaBoost	94.43	85.54	93.02	58.93	79.36	83.26
Random forests	92.17	89.58	83.84	56.94	86.03	81.71
Decision tree	88.26	79.47	74.95	75.39	95.28	82.67
Extreme gradient-boosted trees	78.57	83.05	83.92	84.38	69.59	79.90
k-means clustering	88.53	81.63	94.29	72.98	88.43	85.17
	F1 score					
AdaBoost	95.13	85.93	89.32	92.15	84.38	89.38
Random forests	92.86	85.04	88.62	80.39	83.94	85.78
Decision tree	94.03	78.93	84.99	86.77	84.95	85.93
Extreme gradient-boosted trees	88.35	68.04	89.35	85.39	91.25	84.48
k-means clustering	85.83	85.14	86.27	85.36	90.84	86.69
	ROC					
AdaBoost	90.25	87.49	90.44	89.94	90.23	89.67
Random forests	91.25	88.93	89.44	78.33	91.54	87.90
Decision tree	93.25	82.19	85.36	85.32	94.32	88.09
Extreme gradient-boosted trees	95.24	89.38	86.27	83.53	95.43	89.97
k-means clustering	90.18	90.54	84.66	88.87	90.22	88.89

a) True Positive (TP) - the number of records correctly classified to the Normal class.

b) True Negative (TN) - the number of records correctly classified to the Attack class.

c) False Positive (FP) - The number of records that a record is mistaken for an attack.

d) False Negative (FN) - The number of records that a record is mistaken for normal activity, when in fact it is an attack.

Based on the above terms, the following common evaluation criteria are considered in (10) to (16).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (10)$$

$$Detection\ rate\ (DR) = \frac{TP}{TP + FN} \quad (11)$$

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$F1 - Measure = \frac{2}{\left(\frac{1}{DR}\right) + \left(\frac{1}{Precision}\right)} \quad (14)$$

$$False\ Negative\ Rate\ (FNR) = \frac{FN}{FN + TP} \quad (15)$$

$$False\ Alarm\ Rate\ (FAR) = \frac{FNR + FPR}{2} \quad (16)$$

Accuracy: This number estimates the ratio of correctly identified records to the entire test dataset. If the accuracy is higher, the machine learning model is better (Accuracy $\in$ [0,1]).

Precision: It estimates the ratio of the number of correctly detected attack records to the total number of available attack records. If the accuracy is higher, the machine learning model is better (Precision $\in$ [0,1]).

An F1 score is a measure of the accuracy of a test. It is calculated from the precision and recall of the test, where an F1 score's best value is at 1 and its worst score is at 0. Precision and recall have a relatively equal share in F1 score. If the F1-Score is higher, the machine learning model is better (F1 Score $\in$ [0,1]).

True Positive Rate (TPR): Also called Recall. This ratio estimates Attack records correctly classified to the total number of Attack records. The higher TPR means the machine learning model (TPR $\in$ [0,1]).

False Positive Rate (FPR): It estimates the ratio of the Normal records flagged as Attacks to the total number of Normal records. The lower FPR means the machine learning model (FPR $\in$ [0,1]).

It should be noted that in intrusion detection problems, detection rate (DR) and false alarm rate (FAR) can be called sensitivity and specificity, respectively. The mentioned performance criteria can be calculated with equations (15) and (16). The performance of the proposed intrusion detection system is confirmed in terms of DR, FAR, and DA.

Table II shows the NSL-KDD dataset's traffic distribution. The best percentages of each normal or attack data for training and testing phases are listed in the table. These values are achieved from our simulation.

Table III, also shows the best values of accuracy, detection rate, false positive rate, false negative rate, precision and F1 measure. The results show the proper values of all the

TABLE II. NSL-KDD DATASET'S TRAFFIC DISTRIBUTION

Traffic	*Training*	*Test*
Normal	67%	33%
DOS	69%	31%
Probe	71%	29%
DDOS	63%	37%
Infiltration	65%	35%
Botnet	66%	34%
Web attack	73%	27%

TABLE III. RESULT OF THE PROPOSED METHOD BASED ON DATASET'S TRAFFIC DISTRIBUTION

Attack	*ACC*	*DR*	*FPR*	*FNR*	*Precision*	*F1 measure*
Normal	98.24	95.64	0.03	0.01	99.32	96.04
DOS	99.65	99.83	0.12	0.05	96.04	95.44
Probe	97.53	99.64	0.05	0.07	98.66	97.36
DDOS	98.05	98.27	0.05	0.11	97.37	98.62
Botnet	94.23	97.38	0.15	0.16	99.35	97.55
Web attack	97.72	98.04	0.16	0.02	98.72	98.13

mentioned parameters, so, we can say that our proposed method has a good idea for intrusion detection.

Table IV shows the Detection rate and False alarm rate of the proposed model in each attack class. The proposed model was compared with existing techniques such as BPNN, GA-ANFIS, and PSO-ANFIS. The proposed method shows promising results in terms of both DR and FAR compared to these algorithms.

Intrusion detection performance is measured with the DR and FAR parameters. Because the DR and FAR are more important parameters that are considered for IDS to detect attacks. From the performance of the proposed model, the detection rate, and false alarm compared to other techniques, as shown in the Fig.3, are satisfactory.

5. CONCLUSION

In this research, some issues related to intrusion detection systems are presented, and various techniques for solving problems are discussed. The WOA-based intrusion detection system is a system that has been proposed to detect attacks in networks. The proposed model was used to solve intrusion detection problems, and the model is validated using the NSL-KDD dataset. The proposed model was compared with other techniques such as BPNN, GA-ANFIS, and PSOANFIS. Intrusion detection results based on the NSL-KDD data set were better and more efficient compared to those models because the detection rate was 99.84%, and the FAR result was 0.02%. Future work would be to increase the detection rate and

TABLE IV. COMPARISON OF DIFFERENT METHODS

Method	*Detection rate*	*False alarm rate*
WOA	99.84	0.02
BPNN	98.55	0.14
GA-ANIFS	96.84	0.09
DDOS	98.05	98.27
PSO-ANFIS	95.93	0.18

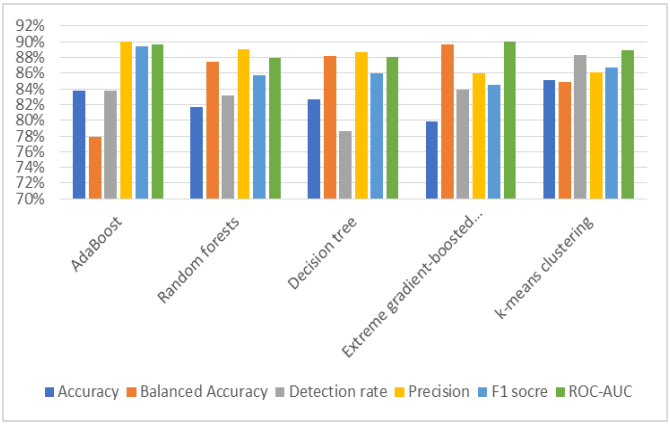

Fig. 3. Graphical plot of overall performance of algorithm with merged attack types

reduce false alarms with a new classifier with another optimization method for detecting attacks.

REFERENCES

[1] World Internet Usage Statistics News and World Population Stats [Online]. https://www.internetworldstats.com/stats.htm [accessed: 18/09/2021].

[2] J. Peng, K. K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *Journal of Network and Computer Applications*, vol. 72, pp.14-27, 2016.

[3] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: A fast filter for the large -scale detection of malicious Web pages categories and subject descriptors," *In Proceedings of the 20th International World Wide Web Conference*, 2017, pp.197-206.

[4] A. Shrivastava, M. Baghel, and H. Gupta, "A Review of Intrusion Detection Technique by Soft Computing and Data Mining Approach," *International Journal of Advanced Computer Research*, vol. 3, pp. 224-228, 2013.

[5] S. Rajabi, S. Jamali, and J. Javadian, "An Intrusion Detection System in Computer Networks using the Firefly Algorithm and the Fast Learning Network," *International Journal of Web Research*, vol. 3, pp. 50-56, 2020.

[6] S. Gamage, and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, p. 102767, 2020.

[7] A. S. Sadiq, B. Alkazemi, S. Mirjalili, N. Ahmed, and S. Khan et al., "An Efficient IDS Using Hybrid Magnetic Swarm Optimization in WANETs," *IEEE Access*, vol. 6, pp. 29041-29053, 2018.

[8] M. H. Ali, B. A. D. Al Mohammed, A. Ismail and M. F. Zolkipli, "A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization," *IEEE Access*, vol. 6, pp. 20255-20261, 2018.

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525-41550, 2019.

[10] V. Hajisalem, and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[11] B. Hajimirzaei, and N. J. Navimipour, "Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm," *ICT Express*, vol. 5, pp. 56–59, 2019.

[12] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *Journal of King Saud University - Computer and Information Sciences*, vol 31, pp. 541–553, 2019.

[13] Y. Y. Chung, and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (SSO)," *Applied Soft Computing*, vol. 12, pp. 3014–3022, 2012.

[14] C. Guo, Y. Ping, N. Liu, and S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[15] S. Singaravelan, R. Arun, D. Arunshunmugam, S. Joy, and D. Murugan, "Inner interruption discovery and defense system by using data mining," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, pp. 592–598, 2020.

[16] S. Shithrth, and W. Prince, "An enhanced optimization based algorithm for intrusion detection in SCADA network," *Computers & Security*, vol. 70, pp. 16–26, 2017.

[17] W. Liang, K. Li, J. Long, X. Kui, and A. Y. Zomaya, "An Industrial Network Intrusion Detection Algorithm Based on Multifeature Data Clustering Optimization Model," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 2063-2071, March 2020.

[18] I. Benmessahel, K. Xie, M. Chellal, and T. Semong, "A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization," *Evolutionary Intelligence*, vol. 12, pp. 131–146, 2019.

[19] S. R. K. Tummalapalli, and A. S. N. Chakravarthy, "Intrusion detection system for cloud forensics using bayesian fuzzy clustering and optimization based SVNN," *Evolutionary Intelligence*, vol. 14, pp. 699-709, 2021.

[20] A. J. Malik, and F. A. Khan, "A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection," *Cluster Computing*, vol. 21, pp. 667–680, 2018.

[21] J. K. Samriya, and N. Kumar, "A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing," Materials Today: Proceedings, In Press, 2020.

[22] B. N. Neethu, S. Jayanthy, and J. A. Kovilpillai, "Greenhouse Monitoring and Controlling using Modified K Means Clustering Algorithm," In *Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, 2019, pp. 456-462.

[23] S. Mirjalili, and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.

Aliakbar Tajari Siahmarzkooh received the B.Sc. degree in Computer Engineering from Ferdowsi University of Iran in 2009, and the M.Sc. and Ph.D. degree in Computer Science from University of Tabriz, Iran in 2012 and 2017, respectively. He has been working with the Department of Computer Sciences, Golestan University, since 2017, where he is now an assistant professor. His current research interests include network security, data mining and artificial intelligence.

Mohammad Alimardani received the B.Sc. degree in Computer Sciences from Golestan University of Iran in 2021. He has been working with the Department of Computer Sciences, Golestan University on data mining approaches since 2019. His current research interests include network, data mining and cloud computing.