Wireless Networks (2022) 28:721–744
https://doi.org/10.1007/s11276-021-02866-x

ORIGINAL PAPER

An enhanced Grey Wolf Optimizer based Particle Swarm Optimizer for intrusion detection system in wireless sensor networks

Mohammed Otair[1] · Osama Talab Ibrahim[1] · Laith Abualigah[1,2] · Maryam Altalhi[3] · Putra Sumari[2]

Accepted: 10 December 2021 / Published online: 27 January 2022

Abstract
The intrusion detection system is a method for detection against attacks, making it one of the essential defense layers. Researchers are trying to find new algorithms to inspect all inbound and outbound activities and identify suspicious patterns that may show an attempted system attack. The proposed technique for detecting intrusions uses the Grey Wolf Optimization (GWO) to solve feature selection problems and hybridizing it with Particle Swarm Optimization (PSO) to utilize the best value to update the information of each grey wolf position. This technique preserves the individual's best position information by the PSO algorithm, which prevents the GWO algorithm from falling into a local optimum. The NSL KDD dataset is used to verify the performance of the proposed technique. The classification is done using the k-means and SVM algorithms to measure the performance in terms of accuracy, detection rate, false alarm rate, number of features, and execution time. The results have shown that the proposed technique attained the necessary improvement of the GWO algorithm when using K-means or SVM algorithms.

Keywords Intrusion detection system (IDS) · Wireless sensor networks (WSN) · Grey wolf optimization (GWO) · Particle swarm optimization (PSO) · K-means

1 Introduction

The Internet has lately been one of the most advanced and powerful communication tools around the world. It is the main key factor that creates a motivating environment for innovation and creativity to build an innovative natural environment. To make such an environment requires consistent development in wireless technologies and installing sensors into various objects. This refers to the ability to allow each item to work independently while connected to the Internet. This concept is known as The Internet of Things (IoT) by Kevin Ashton [1]. IoT is the physical object network that allows specific objects to gather and share data with computers, instruments, cars, buildings, and other items embedded with processors, circuitry, applications, sensors, and network connectivity [2–4].

The Wireless Sensor Networks (WSN) provide the data's connectivity, captured by employing sensors and IoT devices, to record, monitor, and control various environmental conditions, such as water quality, temperature, air quality [5]. The WSN contains many sensors known as nodes, and each node has two tasks: data originator and

✉ Laith Abualigah
Aligah.2020@gmail.com

Mohammed Otair
Otair@aau.edu.jo

Osama Talab Ibrahim
Ojaloudi@gmail.com

Maryam Altalhi
marem.m@tu.edu.sa

Putra Sumari
Putras@usm.my

[1] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

[2] School of Computer Sciences, Universiti Sains Malaysia, 11800 George, Pulau Pinang, Malaysia

[3] Department of Management Information System, College of Business Administration, Taif University, P.O. BOX 11099, Taif 21944, Saudi Arabia

data router. Each node contains four components: sensing, processing, transceiver, and a power source. However, these components are usually limited since each sensor has limited storage, low power, and limited processing capabilities. The WSN contains a sink node, which is also called the Base Station. The sink node collects all the data from the other sensor nodes, acting as the gateway between the sensor nodes and the data processing center [6]; Fig. 1 illustrates the components mentioned previously.

WSN is a great technology; it has some drawbacks, such as inadequate protection and performance problems, including memory insufficiency and sensor battery power, making sensor networks vulnerable to attacks [7]. Thus, the traditional security mechanisms are not enough to detect the WSNs intrusions. Several issues may threaten the security of WSNs, including data confidentiality, data authenticity, and data integrity. An intrusion detection system (IDS) is considered one of the critical methods for defending against hackers. It is a hot field of research, and researchers are trying to find new algorithms for inspecting all inbound and outbound activities to identify suspicious patterns. Intrusion detection monitors the events occurring in a computer system or network and analyzes them for signs of intrusions [8].

Detecting intrusion depends on understanding how the cyber-attack works. In most cases, such abnormal activity consumes network resources intended for specific uses and always affects the network's security and data. There are many types of cyber-attacks, such as device compromise, service disruption, data exfiltration, wrong data injection, and advanced, persistent threat to gain extended access to a device [9]. Also, IDS depends on other methods that see anomaly traffic (unusual traffic activities) using computer algorithms. Therefore, intrusion detection methodologies are classified into three major categories: Signature-based Detection (SD), Anomaly-based Detection (AD), and Stateful Protocol Analysis (SPA) [10].

Once the IDS has identified the dangerous or suspicious traffic, it blocks this activity by itself or by sending alerts to the intrusion prevention system (IPS) to secure the action or prevent intrusions. The IDS studies are classification

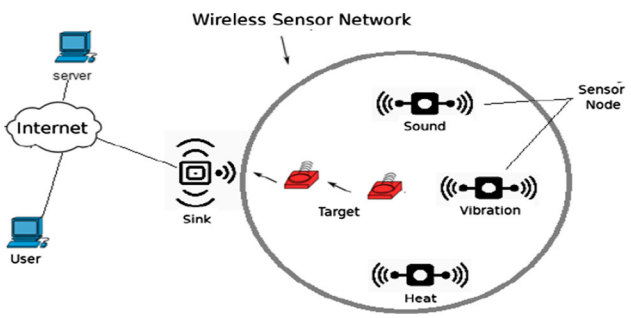


Fig. 1 Wireless sensor networks (WSN) [6]

tasks that separate the expected behavior of networks from attacks [11]. Thus, we need to use machine learning and data mining algorithms to accomplish this task since the attackers do not have a unique pattern and continuously use various tools and methods. Many techniques of machine learning have been used for intrusion detections like SVM and K-means [12, 13]. These techniques classify network connection data into two classes, regular or attacks, based on the connection's features. Before the classification step, it is essential to use optimization algorithms for feature selection. The feature selection process aims to pick up relevant components and exclude others that are irrelevant or redundant to increase the classification process's accuracy. Many models of optimization algorithms are inspired by nature, such as Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO), Arithmetic Optimization Algorithm (AOA) [14]. These algorithms' general aim is to find the highest quality of solutions and the best convergence performance [15–17]. They must also have the exploration and exploitation features; the exploration technique covers all search space areas while the exploitation technique is used to find the optimal solutions within this region.

In wireless sensor networks, intrusion detection systems face many problems such as low detection accuracy, high false alarm alerts, and long processing time. These problems are caused by a vast amount of intrusion, wireless traffic collected by sensors, besides the fact that attackers do not have a unique pattern; they continuously use various tools and methods. In this paper, the protection system against intrusions is built based on selecting significant features that assist the classification process because of its effects, such as increased detection quality and accuracy and reduced execution time. The method of choosing features is mainly based on the GWO and PSO algorithms because it has great power in this area and can determine the relevant features that have to do with it. Still, just as everything has strengths, there are also weaknesses such as the pace of gradual convergence, low accuracy level, and so on. The hybridization with PSO is adopted to achieve the best next position to update each Grey Wolf location information and avoid the GWO algorithm from falling into a local optimum. Because of its capacity in finding the global optimum, convergence level, and simplicity. This paper focuses on extracting the optimal subset of features. The feature selection step aims to reduce the data dimensions by excluding the irrelevant or redundant features to enhance the accuracy and reduce the execution time, leading to an increase in the detection rate and a decrease in the false alarm rate. This was achieved using the proposed hybrid Gray Wolf Optimizer with Particle Swarm Optimizer, where it worked to reduce and choose features from a significant improvement in accuracy, detection rate,

percentage of false alarms, and the speed of the entire process. The proposed method tackles the conventional Gray Wolf Optimizer's weaknesses by adding the search operators of the Particle Swarm Optimizer. This paper has achieved the goal of finding solutions to the challenges mentioned before. Finally, this work offers an excellent base for IDS' future study in many fields. The main contributions of this paper are listed as follows.

1 A new optimization method is proposed for solving the intrusion detection problem in wireless sensor networks.
2 The proposed method is based on using a hybrid search strategy utilizing the main operators of Gray Wolf Optimizer and Particle Swarm Optimizer.
3 The proposed method is validated on benchmark data sets used in domain of intrusion detection systems.
4 The results of the proposed method proved its ability to solve the intrusion detection problems compared with other methods.

The rest of the paper is organized as follows. Section 2 presents an overview of WSN, challenges, threats and attacks, WSNs protection, machine learning-based, and ML-based on the feature. In addition to discussing related studies. Section 3 presents the details of the proposed technique. In this chapter, the details of each stage of the proposed methods will be given. Section 4 provides details about the experimental results of the proposed technique, and it provides a comparison between the results of the proposed approach against some existing methods. Finally, Sect. 5 concludes this research.

2 Literature review

As mentioned earlier, WSNs consist of sensor nodes distributed in different places and are interconnected in a wireless network to collect information. The distributed nature and free wireless medium make WSNs vulnerable to security attacks at various levels. Self-organizing nature, low-battery power supply, limited bandwidth support, and dependency on other nodes are characteristics of sensor networks that expose them to many security attacks at all OSI model layers [18]. Sensor network security is a critical point in WSN. Confidentiality and privacy are necessary for sensitive information, for example, security data or military information. This network must have the capability of resisting separate attacks. One of the most severe challenges is how to protect the WSN since the wireless medium makes it easier for an attacker to spy on the traffic and cripple communication. A group of security issues and threats may face the WSNs [19, 20]; the following section summarizes several matters.

There are many types of IDs with different configurations that serve the same purpose of notifying the system or security administrator [21]. It provides reports about abnormal activities, and some IDS respond by preventing the threat or any attempts to attack. Most IDS define the threat using two commonly used methods, Signature-based Detection (SD) and Anomaly-based Detection (AD). It compares any packet received with this database to identify malicious behavior [22], while the second technique depends on behavioral models. They are based on processing types present in Fig. 2 like statistical-based, computer immunology, user intention identification, and machine learning-based [23].

Machine learning techniques emerged as the best solution to detect malicious patterns by teaching these patterns to the machine model, such as single classifiers, hybrid classifiers, and ensemble classifiers. Many classifier algorithms are used in the ML algorithm to classify data like K-Nearest Neighbor, Self-Organizing Maps, Decision Trees, Random Forest (RF), Naïve Bayes, Artificial Neural Networks (ANN), and Support Vector Machines (SVM) [24, 25].

An improved intrusion detection system is proposed in [19] using the NSL KDD dataset to measure the proposed methods. The developed method was introduced to pick features from the dataset. This was accomplished by increasing the number of wolves of the original algorithm. Two wolves were added to the original algorithm to become five wolves, and then on another experiment, four were added, so the number of wolves became seven. Then, classification was done using the SVM algorithm, and comparators were established to determine the efficiency by increasing the accuracy, detection rate, and decreasing the execution time, features a number, and the false alarm rate. The results showed that seven Wolves have the best results. For the intrusion detection method, Chahal & Kaur [26] suggested a hybrid solution to focus on classification,

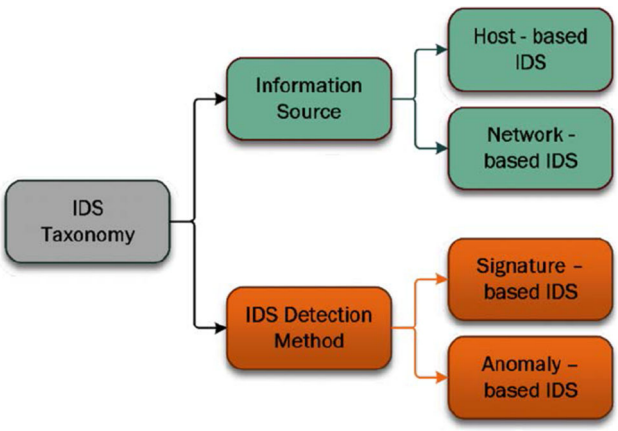


Fig. 2 Anomaly-based Detection (AD) [23]

using the Adaptive-SVM algorithm and clustering, using the K-means algorithm. The proposed system solves these problems (high false-positive rate and low false-negative rate) and generates a better accuracy rate. The NSL-KDD dataset was used to evaluate the performance of this study.

To detect the intrusions and compare their results, Malviya & Jain [27] studied two decision tree classifiers (J48, Id3). The researchers used the attribute selection filter to implement the feature selection step in this study. KDDCUP 99 and a basic k-means algorithm were used for data analysis. The results showed that J48 has better classification accuracy with a high True Positive Rate (TPR) and low False Positive Rate (FPR) compared to ID3 decision tree classifiers. Shukla & Vashishtha [28] proposed a new hybrid intrusion detection system based on Data Mining Technique; the suggested method is combining three different data mining techniques to improve execution efficiency in Intrusion Detection System (IDS). The first stage clustered related data instances based on their behaviors by using clustering as a pre-classification component. The second stage grouped the resulting clusters into attack groups using the Apriori algorithm as a final classification task. The last step, the classification, is applied by using a Decision Tree. KDD'99 is used to calculate IDS efficiency. In terms of precision and performance, the proposed IDS performed better since the Proposed system can classify them into four categories: Probe, Denial of Service (DoS), U2R (User to Root), and R2L (Remote to Local).

An intrusion detection system is suggested in [29] using the MapReduce methodology, based on a parallel particle swarm optimization clustering algorithm. The PSO was used for the clustering task because it prevents sensitivity problems of initial cluster centroids and premature convergence. The results showed that the detection rate was better by keeping the false alarm very low, and the IDS was better at detection speed. KDD99 was used to evaluate the proposed system. The Intrusion Detection System (IDS) is presented using k-means to construct a higher-efficiency and lower-false alarm IDS using the NSL-KDD dataset [30]. The k-means clustering findings have shown that a higher performance rating is obtained when the correct number of clusters is implemented. Increasing or reducing the cluster relative to the number of data types would affect the model's efficiency. Therefore, defining the number of groups affects the findings dramatically. In the beginning, one must know how many sets are required to attain accurate results. Based on the various types of data, 22 groups were used in this model. In a complex network, it would be difficult to identify the number of clusters since there is no "ground data" to act as the basis for determining the number of groups.

Li and Xu in [31] suggested a K-means clustering algorithm and optimization of particle swarm (PSO-KM). Anomaly Intrusion Detection System Experiments on KDD CUP 99 datasets. They revealed the proposed solution's effectiveness, its high detection rate, and low false detection rate. The PSO-KM algorithm combines the particle swarm optimization algorithm with the traditional K-means clustering algorithm: it has the best overall optimization potential. The results illustrate that the PSO-KM algorithm is an effective method when dealing with large datasets. Experimental results show that the detection rate of PSO-KM is improved to detect both known and unknown attacks. It enhances the implementation value of the K-means clustering algorithm in intrusion detection. The proposed technique achieved good results in K-means or SVM in terms of detection rate, accuracy, false alarm rate, number of features, and the whole process time. Table 1 illustrates the differences between the relevant studies mentioned in this section with the proposed technique. We concluded that the current methods are usually used optimization methods to solve the IDS problems. It is clear that the need for a new effect method is essential to solve the IDS problems as mentioned above.

3 The proposed method

The mission of IDS is essential to detect malicious activities. It shows how machine learning techniques are powerful in processing a massive amount of wireless intrusion traffic to classify abnormal and regular traffic. It was also previously mentioned how the features selection improves the classifier algorithms in terms of the processing time and the accuracy of the detection rate by reducing the number of features. In this paper, the features selection step was done by using the optimization algorithms. The definition of an optimization algorithm is how built a computer program that can generate models procedurally with the ability to change several parameters to minimize or maximize an objective. Furthermore, the main challenge is identifying the parameters that formulate the computer's problem and design a robust optimization algorithm to find the best design. The main components of an optimization problem, shown in Fig. 3: set of decision variables, an objective function, bounds on the decision variables, and constraints.

As shown in Fig. 4, PSO and GWO are used in the proposed technique, and they belong to the same family of population-based algorithms, called swarm-based algorithms [33]. To achieve the main objective of this paper, a quantitative research methodology is used. The dataset was imported from the NSL-KDD dataset that is available online (https://www.unb.ca/cic/datasets/nsl.html). Features

Table 1 Result of related studies

References	False alarm rate	Accuracy	Detection rate	No. of features	execution time
[19]	Low	High	High	Low	Low
[26]	Low	High	High	NA	NA
[27]	Low	High	High	Low	NA
[28]	Low	High	High	NA	Low
[29]	low	NA	High	NA	Low
[30]	Low	NA	High	NA	NA
[31]	low	NA	High	NA	NA
Proposed GWO-PSO-K-means	low	High	High	Low	Low
Proposed GWO-PSO-SVM	low	High	High	Low	Low

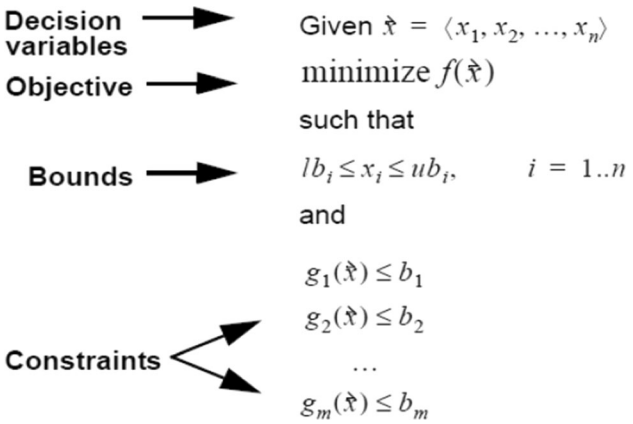


Fig. 3 Component of an optimization problem [32]

extraction step using the proposed technique was based on the hybridized algorithm. Finally, the classification process was applied based on K-means and SVM to divide the features (variables of a dataset) into separate groups: regular or attack.

3.1 NSL-KDD dataset

The NSL KDD dataset is an updated version of the KDD cup99 data set, which is suggested to solve the previous version's problems. It has several advantages over the original KDD data set [34]; a sufficient number of records will be available in the train and test datasets. There are no duplicate records in the test sets, and it does not include redundant records in the train set. There are 41 types of features in each record, and those are considered either attack type or regular type. Each feature is categorized into three attribute value types. (Nominal, Binary, and Numeric). Figure 5 shows the 41 features of the NSL-KDD dataset.

These types of attack classes are categorized into four parts; DoS- Denial of service, Probing- Surveillance and other probing attacks, U2R- Unauthorized access to local superuser, and R2L- Unauthorized access from a remote machine. Table 2 shows the types of attacks as per the above categorization [35]. Table 3 shows the distribution of the ordinary and attack records in the various NSL-KDD datasets [35].

3.2 Preparing dataset

This section presents a data preparation and preprocessing framework for producing qualitative data for experimental analysis. The experimental study was conducted here on the intrusion detection data. The following two subsections illustrate the main phases of the dataset's preprocessing (Data Transformation and normalization).

3.2.1 Data transformation

All nominal attributes are converted to a numeric value in the data transformation stage. For example: To convert the original values to numerical values such as tcp = 1, udp = 2, and icmp = 3, the protocol type attribute is given as an integer number. As shown in Fig. 6, the same transformation technique is adopted to convert nominal values [19].

3.2.2 Normalization

Values are scaled in the data normalization process using Eq. (1) since the NSL KDD dataset attributes are not distributed uniformly.

$$X' = (Original\ value - Min\_Value)/(Maxvalue - Min\_Value) \quad (1)$$

where X' is the normalized value [36]. Figure 7 shows the data set before the normalization phase, and Fig. 8 shows the dataset after the normalization phase.

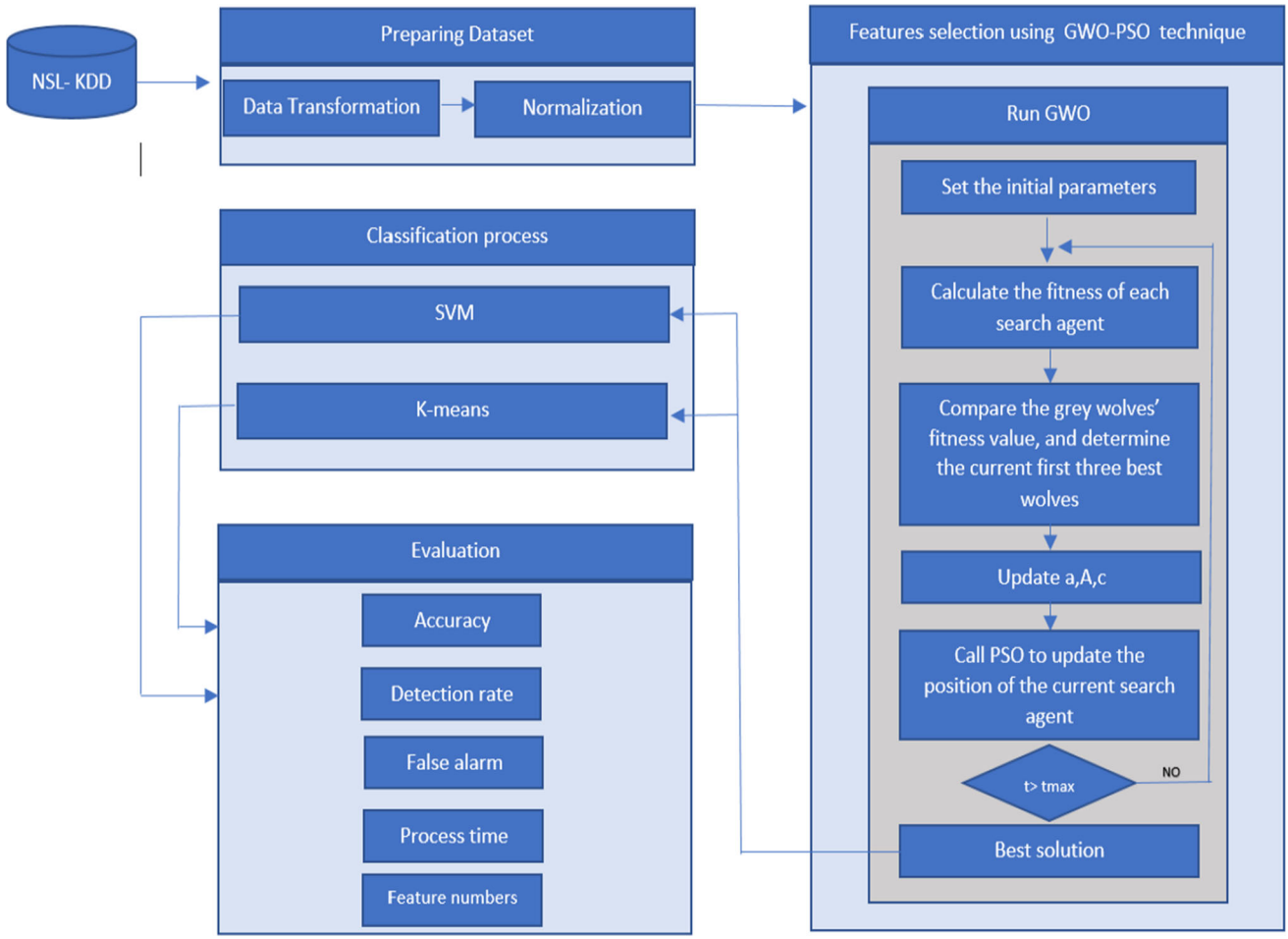


Fig. 4 The Proposed IDS method

3.3 The proposed feature selection method

In this process, the feature selection algorithm is built based on hybridizing the Gray Wolf Optimizer (GWO) with Particle Swarm Optimizer (PSO).

3.3.1 Grey wolf optimization (GWO)

This algorithm was introduced by Mirjalili in [37] and inspired by the nature of wolves. It mirrors the behavior and the hunting strategies of the grey wolves and works on leadership hierarchy hunting strategies. Alpha (α) leads the group, while Beta (β) is the second group and assists the Alpha group. The next level of the hierarchy contains Delta (δ) and Omega (ω) wolves. Delta wolves follow the upper level of the hierarchy and control the Omega wolves. Figure 9 illustrates the hierarchy of a grey wolf.

The main phases of grey wolf hunting are as follows [37]:

- Tracking, chasing and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attack towards the prey.

Figure 10 presents the hunting behavior of grey wolves: (A) chasing, approaching, and tracking prey (B–D) pursuing, harassing, and encircling (E) stationary situation and attack.

The exploration search is done by the upper three levels of wolves and aims to find the best position. The following Eqs. (2–12) describe the grey wolf surrounding the prey [37].

$$\vec{D} = \left| \vec{C}.\vec{X}_P(t) - \vec{X}(t) \right| \tag{2}$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A}.\vec{D} \tag{3}$$

where, (t) Is the number of the current iteration, $\vec{x}_p$ is the position vector of the prey, $\vec{x}$ is the position vector of a grey wolf, and $\vec{A}$ And $\vec{C}$ are coefficient vectors and they are calculated by:

Fig. 5 Features of NSL-KDD dataset

No.	Feature Name	No.	Feature Name
1	Duration	22	is_guest_login
2	protocol type	23	**count**
3	service	24	srv_count
4	**flag**	25	**serror_rate**
5	src_bytes	26	**srv_serror_rate**
6	**dst_bytes**	27	**rerror_rate**
7	land	28	srv_rerror_rate
8	wrong_fragment	29	**same_srv_rate**
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	**dst_host_srv_count**
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	**dst_host_serror_rate**
18	num_shells	39	**dst_host_srv_serror_rate**
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login	42	

Table 2 Types of attacks

DoS	Back, Land, Neptune, Pod, Smurf, teardrop, Mailbomb, Processtable, Udpstorm, Apache2, Worm
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Xlock, xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named

Table 3 Distribution of the normal and attack records

Dataset type	Total no. of					
	Records	Normal	DoS	Probe	U2R	R2L
KDD Train +	125,973	67,343	459 27	11,656	52	995
		53.46%	36.46%	9.25%	0.04%	0.79%
KDD Test +	22,544	9711	7458	2421	200	2754
		43.08%	33.08%	10.74%	0.89%	12.22%

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r}_1 - \overrightarrow{a} \tag{4}$$

$$\overrightarrow{C} = 2 \cdot \overrightarrow{r}_2 \tag{5}$$

where, $\overrightarrow{a}$ is the exploration rate (linearly decreased from 2 to 0 over the course of iterations), and $\overrightarrow{r}_1$ and $\overrightarrow{r}_2$ are random vectors in [0, 1].

Figure 11 illustrates the possible areas in which the wolf moves and updates its positions according to the position of the prey. At each iteration, values of $\overrightarrow{A}$ and $\overrightarrow{C}$ update the position of grey wolf, in the same figure, the 3-dimensional position update of the grey wolf can be seen. Using $\overrightarrow{r}_1$ and $\overrightarrow{r}_2$, the grey wolf can update its position to any random position by Eqs. 2 and 3. When |A| becomes less than 1, the

Attribute	Attribute Value With Their Numeric Value
Protocol type	tcp=1,udp=2,icmp=3
Service value	private=1 ftp_data=2 eco_i=3 telnet=4 http=5 smtp=6 ftp=7 ldap=8 pop_3=9 courier=10 discard=11 ecr_i=12 imap4=13 domain_u=14 mtp=15 systat=16 iso_tsap=17 other=18 csnet_ns= 19 finger=20 uucp=21 whois =22 netbios_ns=23 link=24 Z39_50=25 sunrpc=26 auth=27 netbios_dgm=28 uucp_path=29 vmnet=30 domain=31 name=32 pop_2=33 http_443=34 urp_i=35 login=36 gopher=37 exec=38 time=39 remote_job=40 ssh=41 kshell=42 sql_net=43 shell=44 hostnames=45 echo=46 daytime=47 pm_dump=48 IRC=49 netstat=50 ctf=51 nntp=52 netbios_ssn=53 tim_i=54 supdup=55 bgp=56 nnsp=57 rje=58 printer=59 efs=60 X11=61 ntp_u=62 klogin=63 tftp_u=64 red_i=65 urh_i=66 http_8001=67 aol=68 http_2784=69 harvest=70
Flag value	REJ=1 SF=2 RSTO=3 S0=4 RSTR=5 SH=6 S3=7 S2=8 S1=9 RSTOS0=10 OTH=11
Classification of attack	neptune=1 normal=2 saint=3 mscan=4 guess_passwd=5 smurf=6 apache2=7 satan=8 buffer_overflow=9 back=10 warezmaster=11 snmpgetattack=12 processtable=13 pod=14 httptunnel=15 nmap=16 ps=17 snmpguess=18 ipsweep=19 mailbomb=20 portsweep=21 multihop=22 named=23 sendmail=24 loadmodule=25 xterm=26 worm=27 teardrop=28 rootkit=29 xlock=30 perl=31 land=32 xsnoop=33 sqlattack=34 ftp_write=35 imap=36 udpstorm=37 phf=38 warezclient=39 spy=40.

Fig. 6 Transform Methodology [19]

grey wolf attacks the prey, but the random numbers $\vec{r}_1$ and $\vec{r}_2$ may cause the grey wolf to fall into local optimal [38].

The hunting phase is guided by Alpha (α), Beta (β) and Delta (δ), which may also be involved in hunting occasionally. Until now we don't know the location of prey. Therefore, the best candidate solution Alpha, Bata and Delta have been assumed to have good knowledge about the position of the prey [37].

$$\vec{D}_{\alpha} = \left|\vec{C}_1.\vec{X}_{\alpha} - \vec{X}\right| \tag{6}$$

$$\vec{D}_{\beta} = \left|\vec{C}_2.\vec{X}_{\beta} - \vec{X}\right| \tag{7}$$

$$\vec{D}_{\delta} = \left|\vec{C}_3.\vec{X}_{\delta} - \vec{X}\right| \tag{8}$$

After obtaining the above position vector, the wolves will perform the last update by adopting the following [37].

$$\vec{X}_{1=}\vec{X}_{\alpha} - \vec{A}_1.(\vec{D}_{\alpha}) \tag{9}$$

$$\vec{X}_{2=}\vec{X}_{\beta} - \vec{A}_2.(\vec{D}_{\beta}) \tag{10}$$

$$\vec{X}_{3=}\vec{X}_{\delta} - \vec{A}_3.(\vec{D}_{\delta}) \tag{11}$$

$$\vec{\mathbf{X}}(\mathbf{t}+1) = \frac{\vec{\mathbf{X}}_1 + \vec{\mathbf{X}}_2 + \vec{\mathbf{X}}_3}{3} \tag{12}$$

During each iteration update, the grey wolf's position is estimated by the best three levels of positions. $\vec{X}(t+1)$ is the updated position of the next generation of wolves. Each candidate solution will update the distance between them and the prey. Figure 12 presents the Pseudo code, and

255	10	0.04	0.06	0	0	0	0	1	1
255	1	0	0.06	0	0	0	0	1	1
134	86	0.61	0.04	0.61	0.02	0	0	0	0
3	57	1	0	1	0.28	0	0	0	0
29	86	0.31	0.17	0.03	0.02	0	0	0.83	0.71
155	255	1	0	0.01	0.03	0.01	0	0	0
255	28	0.11	0.72	0	0	0	0	0.72	0.04
255	255	1	0	0	0	0.01	0.01	0.02	0.02
151	255	1	0	0.01	0.03	0	0	0	0
52	26	0.5	0.08	0.02	0	0	0	0	0
255	128	0.5	0.01	0	0	0	0	0.66	0.32
255	129	0.51	0.03	0	0	0	0	0.33	0
255	2	0.01	0.07	0	0	0	0	1	1
235	171	0.73	0.07	0	0	0.69	0.95	0.02	0
38	73	0.16	0.05	0.03	0.04	0	0.77	0	0.07
71	255	1	0	0.01	0.04	0	0	0	0
255	255	1	0	0	0	0	0	0	0
35	255	1	0	0.03	0.05	0	0	0	0
255	255	1	0	1	0	0	0	0	0
255	18	0.07	0.07	0	0	0	0	1	1
255	19	0.07	0.05	0	0	0	0	1	1
255	87	0.34	0.01	0.01	0	1	1	0	0
255	255	1	0	0	0	0	0	0	0
36	255	1	0	0.03	0.02	0	0	0	0
255	8	0.03	0.06	0	0	0	0	1	1
255	13	0.05	0.06	0	0	0	0	1	1
180	255	1	0	0.01	0.01	0	0	0	0

Fig. 7 Dataset before normalization phase

0.448141	0.019569	0	0	1	1	0.04	0.06	0	1	0.039216	0.04	0.06	0	0	0	0	1
0.266145	0.001957	0	0	1	1	0.01	0.06	0	1	0.003922	0	0.06	0	0	0	0	1
0.001957	0.001957	0	0	0	0	1	0	0	0.52549	0.337255	0.61	0.04	0.61	0.02	0	0	0
0.001957	0.127202	0	0	0	0	1	0	1	0.011765	0.223529	1	0	1	0.28	0	0	0
0.001957	0.015656	0	0.12	1	0.5	1	0	0.75	0.113725	0.337255	0.31	0.17	0.03	0.02	0	0	0.83
0.007828	0.007828	0	0	0	0	1	0	0	0.607843	1	1	0	0.01	0.03	0.01	0	0
0.001957	0.005871	0	0	0	0	1	0	1	1	0.109804	0.11	0.72	0	0	0	0	0.72
0.001957	0.001957	0	0	0	0	1	0	0	1	1	1	0	0	0	0.01	0.01	0.02
0.064579	0.091977	0	0	0	0	1	0	0.04	0.592157	1	1	0	0.01	0.03	0	0	0
0.001957	0.001957	0	0	0	0	1	0	0	0.203922	0.101961	0.5	0.08	0.02	0	0	0	0
0.001957	0.001957	0	0	0	0	1	0	0	1	0.501961	0.5	0.01	0	0	0	0	0.66
0.001957	0.003914	0	0	0	0	1	0	1	1	0.505882	0.51	0.03	0	0	0	0	0.33
0.217221	0.003914	0	0	1	1	0.02	0.07	0	1	0.007843	0.01	0.07	0	0	0	0	1
0.234834	0.234834	1	1	0	0	1	0	0	0.921569	0.670588	0.73	0.07	0	0	0.69	0.95	0.02
0.001957	0.001957	0	0	0	0	1	0	0	0.14902	0.286275	0.16	0.05	0.03	0.04	0	0.77	0
0.015656	0.015656	0	0	0	0	1	0	0	0.278431	1	1	0	0.01	0.04	0	0	0
0.046967	0.046967	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0
0.031311	0.031311	0	0	0	0	1	0	0	0.137255	1	1	0	0.03	0.05	0	0	0
0.988258	0.988258	0	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0
0.399217	0.035225	0	0	1	1	0.09	0.07	0	1	0.070588	0.07	0.07	0	0	0	0	1

Fig. 8 Dataset after normalization phase

Fig. 13 shows the Grey Wolf Optimization (GWO) algorithm's flowchart.

3.3.2 Particle swarm optimization (PSO)

This algorithm was introduced by Kennedy and Eberhart in 1995 [39]. It was inspired by how birds flock while searching for food. Once one bird finds the food, it would send a message to the remaining birds to keep them updated about the fresh food's position. The PSO algorithm aims to find global optimization; every swarm bird is called a particle [15]. The main advantage of PSO is that it has fewer parameters to adjust and fast convergence. It comprises the following steps [40]; initialize each particle with random position and velocity, evaluate the fitness of each particle, update P_{best} and G_{best} of each particle, update velocity of each particle using Eq. (13), and update position of each particle using Eq. (14). Figure 14 shows a flowchart of the PSO algorithm.

$$V_p{}^{(t+1)} = \mathrm{w}.V_p{}^{(t)} + c_1 * r_1 * \left(P_{best} - x_p{}^{(t)}\right) + c_1 * r_1 * \left(G_{best} - x_p{}^{(t)}\right) \tag{13}$$

$$x_p{}^{(t+1)} = x_p{}^{(t)} + V_p{}^{(t+1)} \tag{14}$$

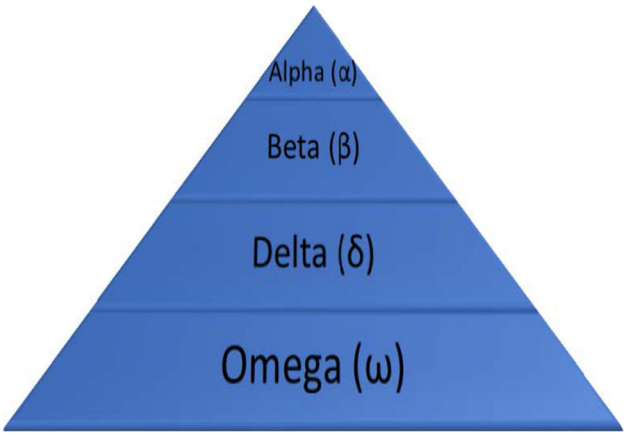


Fig. 9 Hierarchy of grey wolf

3.3.3 The proposed technique based on hybrid GWO-PSO algorithm

As mentioned earlier, GWO is strong at exploitation but weak at avoiding a premature convergence and local optimum. The PSO algorithm has a solid exploration capability, but it lacks exploitation. In this section, the GWO-PSO hybrid algorithm is proposed to combine the GWO exploitation capability with the PSO exploration capability to obtain a better global optimization capability. Hybridization is to acquire the balance between exploitation and exploration to extract the optimal subset of features and reduce the data dimensions by excluding the irrelevant or redundant features. The hybrid GWO-PSO is proven as an effective optimization technique when seeking the global best solution to an optimization problem.

Figure 15 shows the flowchart of the GWO-PSO algorithm. The proposed technique consists of the following steps; initialization of the search agents and defining the solution area, running the GWO technique, generating the lowest values for all agents, passing these agents to the PSO technique as initial points, returning the modified positions to the GWO, and repeating these steps until the stopping criteria are reached.

GWO-PSO alternately uses the PSO algorithm for exploration in the search space and the GWO algorithm for exploitation to search the global optimum without changing the general operation of the GWO. In order to do this hybridization technique, the updated position of the next generation of Wolves will perform the last update by adopting the Eq. (15) Instead of using Eq. (12) in GWO algorithm.

$$\vec{X}(t+1) = \vec{X}(t+1) + V_p^{(t+1)} \tag{15}$$

3.4 Classification process

3.4.1 K-means technique

K-means is an unsupervised machine learning algorithm used to cluster and analyze the data, first introduced by James MacQueen in 1967 [42]. It aims to divide the features (attributes of a dataset: the output of the selection

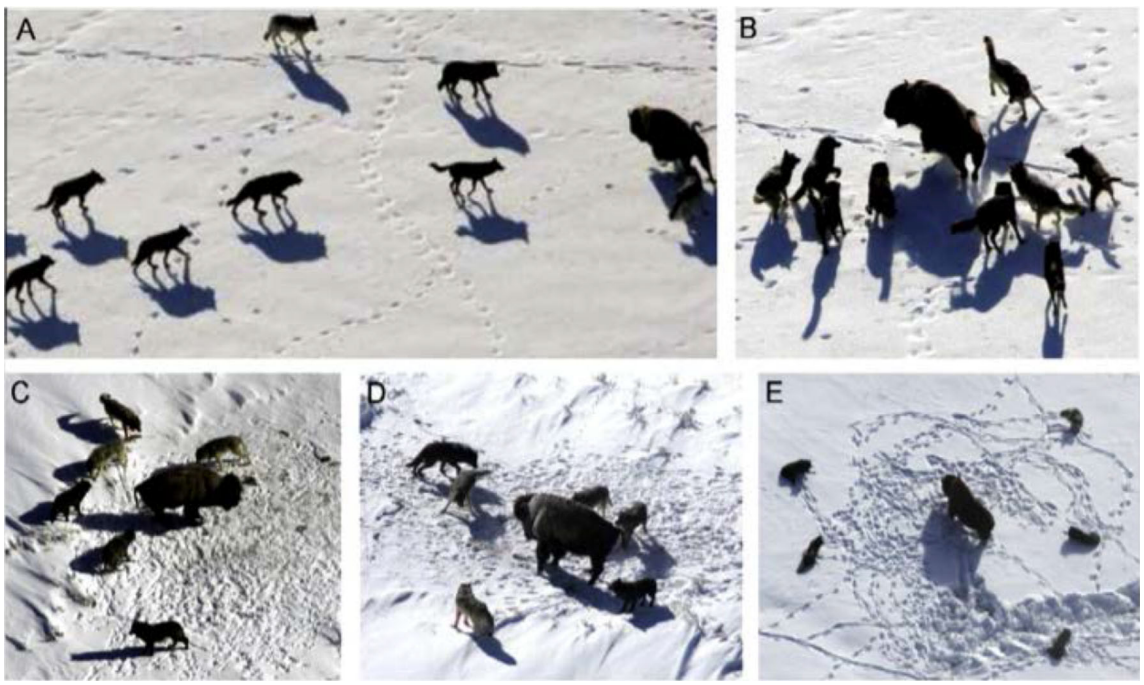


Fig. 10 Hunting behavior of grey wolves [37]

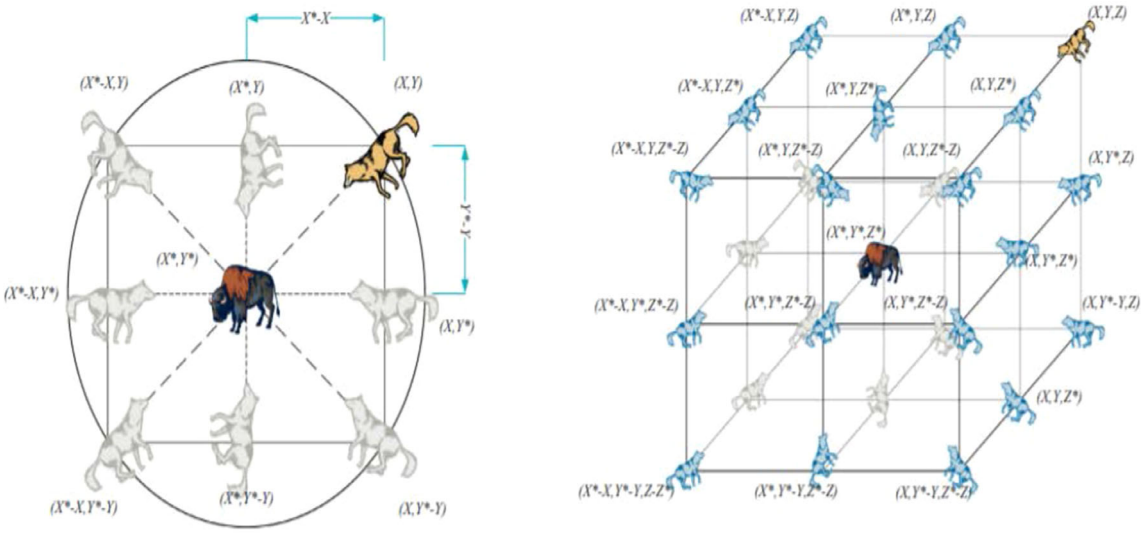


Fig. 11 Position vectors and their possible next locations [37]

feature process in this study) into separate groups or clusters. It seeks to classify the given dataset into a certain number (k) of sets based on the similarity degree. It attempts to make the data within a cluster as similar as possible while maintaining a low degree of similarity between groups and reducing the data's complexity. Moreover, K-means is very useful and famous in the data mining field. It has many advantages, including its simplicity in implementation, efficiency, and low memory consumption compared to other clustering techniques [43]. Clustering refers to an aggregated number of points together because of some similarities. The 'means' refers to averaging the data. Finally, K refers to the number of centroids you need in the dataset.

As shown in Fig. 16, to proceed with the clustering process, the first step should identify the K initial centroids. It is well known that K-means are used for the clustering process. However, in our case, it is used as a classifier since the clusters are known, customary, and attacks; therefore, the K-means separates the dataset into two classes depending on the distance between the data, so the K value should be two, referring to the number of clusters; either normal or abnormal (Attacks), For example, a higher average in the number of packets can be taken as an indicator for a strange cluster. Then, iterative calculations are performed to centroids until they have stabilized. No additional change in centroid values occurs either because the clustering has been successful or the defined number of iterations has been achieved, as shown in Fig. 17.

3.4.2 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm used for either classification or regression, developed in [45]. SVM aims to determine an optimal separating hyperplane (OSH). It is mainly used for classification issues. We map each data object in the SVM algorithm as a point in n-dimensional space (where n is the number of features). Then, the classification is performed by finding the hyper-plane that differentiates the two classes, as shown in Fig. 18. SVM is preferable by many as it provides high accuracy with less processing power [46]. Therefore, in this paper, another experiment is done using the SVM technique to measure classification performance.

4 Experimental results

The experimental results of the proposed technique will be discussed in this chapter. Also, the assessment metrics mention in Sect. 1.7 used to evaluate the performance of the proposed method. MATLAB R2020b is used to implement the proposed approach, a powerful computational package based on a proprietary computational language that provides tools for users with a wide range of programming knowledge. The software package will direct the project from end-to-end, from graphical user interfaces that can run the experiment to real-time data collection, analysis, and data production. MATLAB can perform calculations based on a large dataset that would be time-prohibitive in conventional statistical rows and column packs. Table 4 shows the environment in which these experiments were applied.

4.1 Testing and analysis

This section provides a detailed evaluation and comparison of the proposed technique GWO-PSO, GWO, and PSO

Initialize the grey wolf population $\boldsymbol{x_i(i = 1, 2, \ldots \ldots, n)}$

Initialize α, $\boldsymbol{A}$ *and* $\boldsymbol{C}$

Calculate the fitness of each search agent

$\boldsymbol{x_\alpha}$ = ***the best search agent***

$\boldsymbol{x_\beta}$ = ***the second best search agent***

$\boldsymbol{x_\delta}$ = ***the third best search agent***

While (t<Max number of iterations)

For each search agent

Update the position of the current search agent

End for

Update α, $\boldsymbol{A}$ *and* $\boldsymbol{C}$

Calculate the fitness of all search agents

Update $\boldsymbol{x_\alpha, x_\beta}$ *and* $\boldsymbol{x_\delta}$

t=t+1

end while

return $\boldsymbol{x_\alpha}$

Fig. 12 Pseudo code of the GWO algorithm

used to select the relevant feature. The K-means and SVM algorithms were used in the classification process to illustrate the extent of improvement based on the proposed technique. The measurement is done on the assessment metrics mention in Sect. 1.7. The below equations are used to calculate the accuracy, detection rate, and false alarm rate [19]. Figure 19 shows the confusion chart values.

$$Accuracy = \left(\frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}\right) \quad (16)$$

$$DetectionRate = \left(\frac{true\ positive}{true\ positive + false\ negative}\right) \quad (17)$$

$$Falsealarm = \left(\frac{false\ positive}{true\ negative + false\ positive}\right) \quad (18)$$

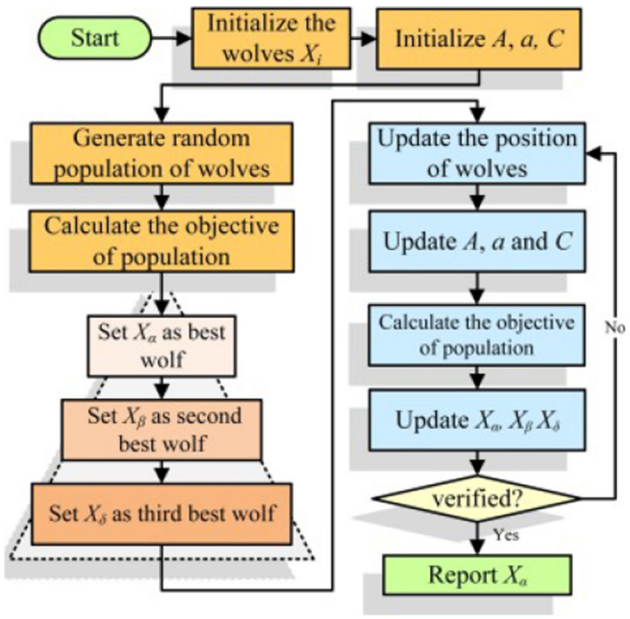


Fig. 13 Flowchart of Grey Wolf Optimization (GWO) algorithm [37]

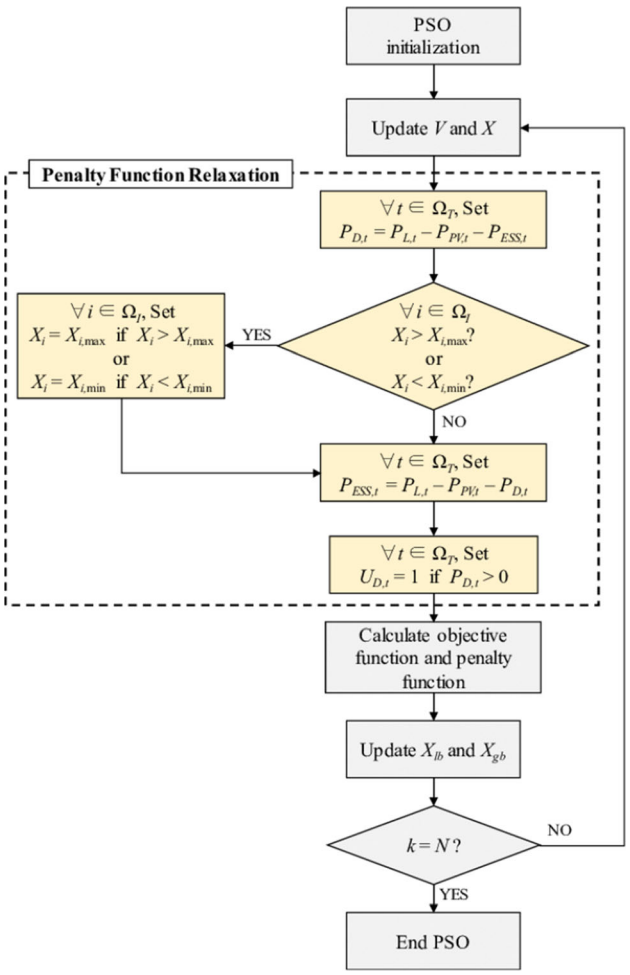


Fig. 14 Flowchart of PSO algorithm [41]

4.2 Results and discussion

In this section, the proposed technique was used to select the relevant features as shown in Table 5; 20 components were chosen out of 41 features in the original datasets mentioned in Sect. 3.2, representing 48.78% of the total features. The bold font in the given tables refer to the best result.

Figure 20 shows the confusion chart for K-means results after using the appropriate features that resulted from the proposed GWO-PSO technique. The proposed method predicts 9455 records as regular records out of 9711 regular records, representing 97.363% as True Negative (TN). The proposed approach indicates 256 records as attack records out of 9711 regular records, representing 2.636% as False Positive (FP). The proposed technique predicts 5497 records as regular records out of 12,833 attack records, representing 42.834% as False Negative (FN). The proposed approach indicates 7336 records as attack records out of 12,833 attack records, representing 57.165% as True Positive (TP).

Figure 21 shows the confusion chart for SVM results after using the appropriate features that resulted from the proposed GWO-PSO technique. The proposed method predicts 9660 records as regular records out of 9711 regular records, representing 99.475% as True Negative (TN). The proposed approach indicates 51 records as attack records out of 9711 regular records, representing 0.525% as False Positive (FP). The proposed technique predicts 182 records as regular records out of 12,833 attack records, representing 1.418% as False Negative (FN). The proposed approach indicates 12,651 records as attack records out of 12,833 attack records, representing 98.582% as True Positive (TP). Table 6 shows the results based on the assessment metrics: accuracy, detection rate, false alarm, process time, and feature number.

The GWO algorithm was used in this section to select the relevant features as shown in Table 7; 26 features were chosen out of 41 features in the original datasets that were mentioned in Sect. 3.2, which represent 63.414% of the total features.

Figure 22 shows the confusion chart for K-means results after using the appropriate features that resulted from the GWO algorithm. The proposed technique predicts 9493 records as regular records out of 9711 regular records, representing 97.755% as True Negative (TN). The proposed approach indicates 218 records as attack records out of 9711 regular records, representing 2.245% as False Positive (FP). The proposed technique predicts 6061 records as regular records out of 12,833 attack records, representing 47.230% as False Negative (FN). The proposed approach indicates 6772 records as attack records

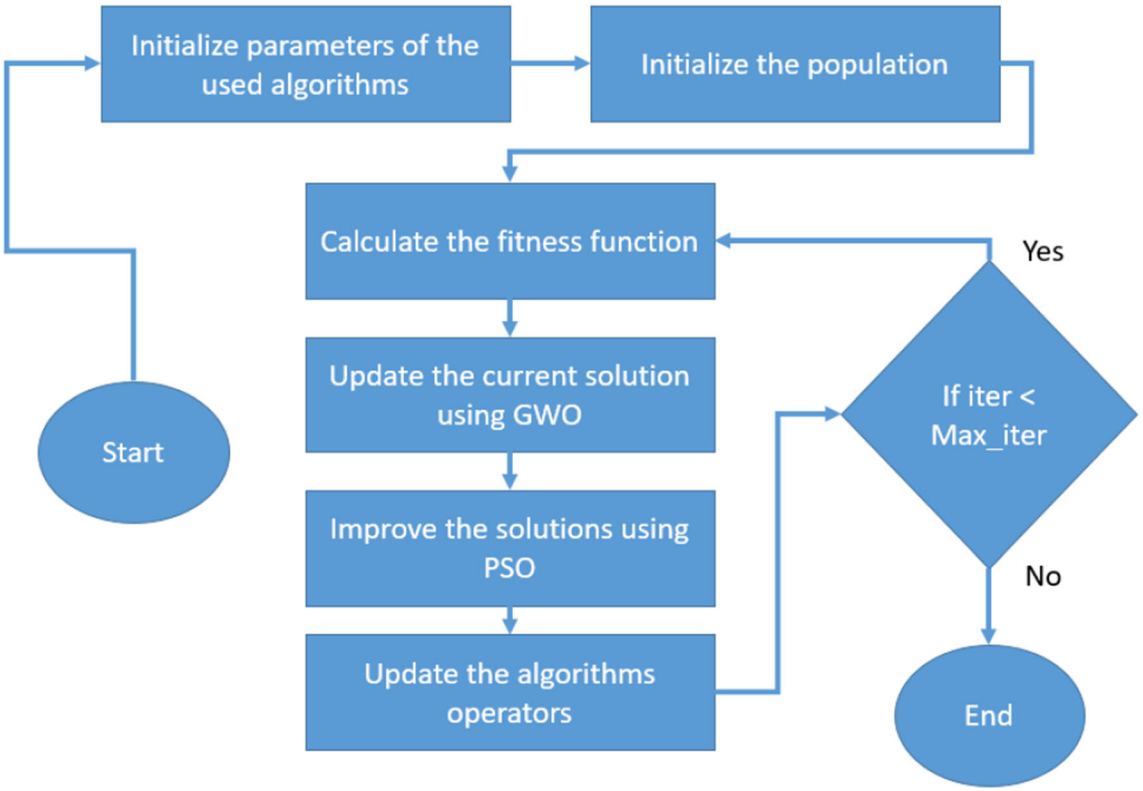
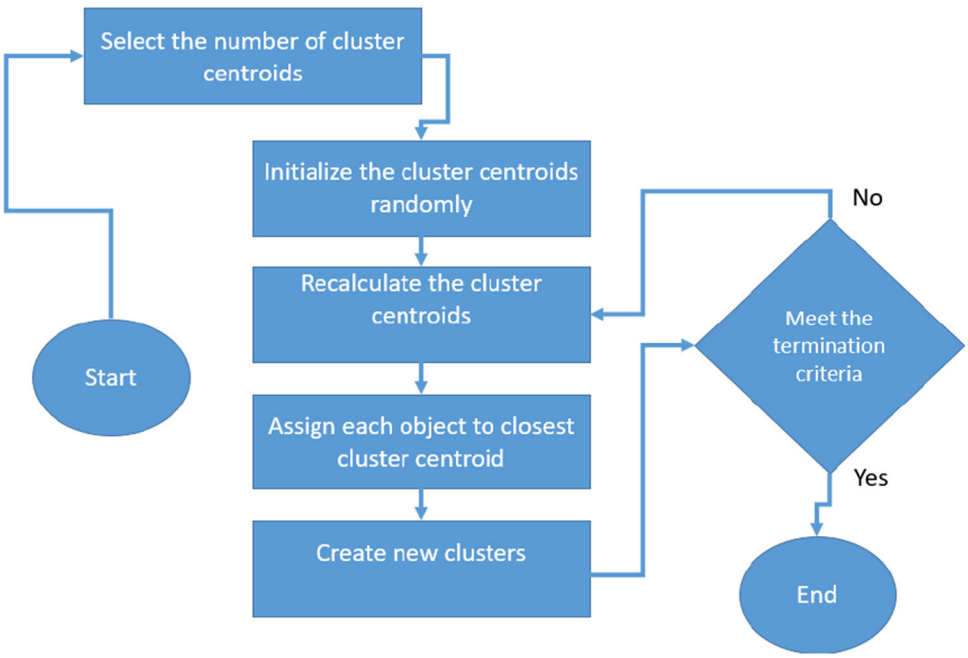


Fig. 15 Flowchart of GWO-PSO algorithm

Fig. 16 Flowchart of K-means [44]



out of 12,833 attack records, representing 52.770% as True Positive (TP).

Figure 23 shows the confusion chart for SVM results after using the appropriate features that resulted from the GWO algorithm. The proposed technique predicts 9654 records as regular records out of 9711 regular records, representing 99.413% as True Negative (TN). The proposed method indicates 57 records as attack records out of 9711 regular records, representing 0.587% as False Positive (FP). The proposed technique predicts 288 records as

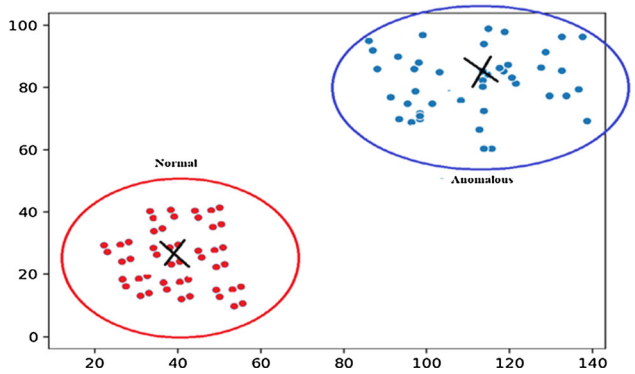


Fig. 17 Attack detection using k-means algorithm

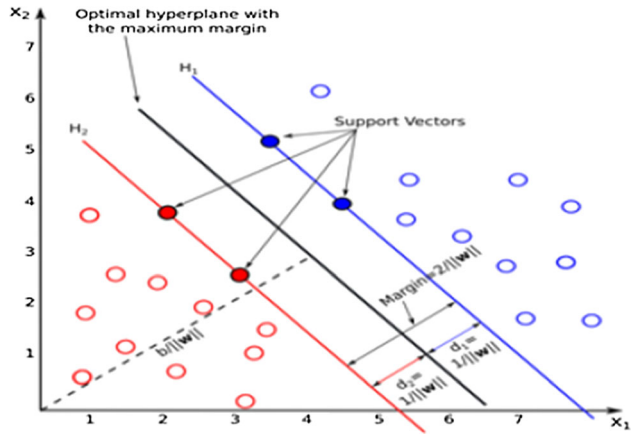


Fig. 18 SVM algorithm [47]

Table 4 Environment Specifications

Operating system	
Operating system	Windows Server 2012 R2 Datacenter
System type	64-bit Operating System, × 64-based processor
Hardware	
Processor	Intel(R) Xeon(R) Platinum 817 M CPU @ 2.60 GHz 2.10 Hz
RAM	16.0 GB
Tools	
MATLAB	R2020b
Other	Excel 2013

regular records out of 12,833 attack records, representing 2.244% as False Negative (FN). The proposed approach indicates 12,545 records as attack records out of 12,833 attack records, representing 97.756% as True Positive (TP). Table 8 shows the results based on the assessment metrics**:**

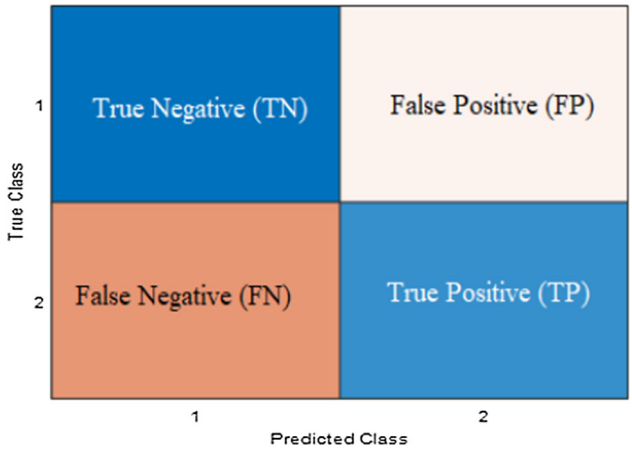


Fig. 19 Confusion chart

accuracy, detection rate, false alarm, process time, and feature number.

The PSO algorithm was used to select the relevant features as shown in Table 9; 24 features were chosen out of 41 features in the original datasets mentioned in Sect. 3.2, representing 58.536% of the total features. Figure 24 shows the confusion chart for K-means results after using the appropriate features that resulted from the PSO algorithm. The proposed technique predicts 9483 records as regular records out of 9711 regular records, representing 97.652% as True Negative (TN). The proposed method indicates 228 records as attack records out of 9711 regular records, representing 2.348% as False Positive (FP). The proposed technique predicts 5789 records as regular records out of 12,833 attack records, representing 45.110% as False Negative (FN). The proposed method indicates 7044 records as attack records out of 12,833 attack records, representing 54.890% as True Positive (TP).

Figure 25 shows the confusion chart for SVM results after using the appropriate features that resulted from the PSO algorithm. The proposed technique predicts 9666 records as normal records out of 9711 normal records, representing 99.537% as True Negative (TN). The proposed technique predicts 45 records as attack records out of 9711 normal records, representing 0.463% as False Positive (FP). The proposed technique predicts 287 records as normal records out of 12,833 attack records, representing 2.236% as False Negative (FN). The proposed technique predicts 12,546 records as attack records out of 12,833 attack records, representing 97.764% as True Positive (TP). Table 10 shows the results based on the assessment metrics**:** accuracy, detection rate, false alarm, process time, and feature number.

In terms of feature numbers, a comparison of GWO-PSO, GWO, and PSO is made. The proposed technique achieved the study's objective by reducing the number of

Table 5 List of selected features by GWO-PSO

Method	Selected features	Total
GWO-PSO	2, 6, 7, 8, 9, 11, 12, 14, 15, 16, 21, 24, 25, 26, 28, 29, 30, 32, 35, 38	**20**

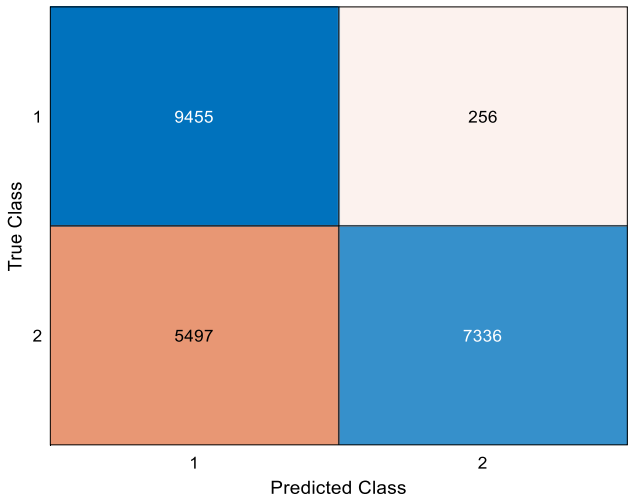


Fig. 20 Confusion chart of K-means using GWO-PSO selected features

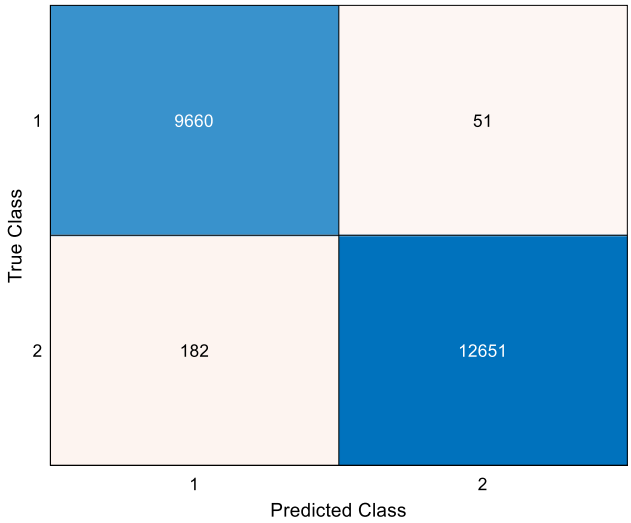


Fig. 21 Confusion chart of SVM using GWO-PSO selected features

features and selecting the relevant features. 20 relevant features were chosen from GWO-PSO. In comparison, 24 and 26 relevant features were selected from PSO and GWO, respectively, as shown in Fig. 26.

Table 11 provides a comparison of the GWO-PSO-K-means, GWO-K-means, and PSO-K-means algorithms in terms of accuracy, detection rate, and false alarm and process time. It is clear that the proposed GWO-PSO-K-means got a smaller number of features with a higher accuracy rate compared to other methods. As seen in Table 11, except for the false alarm rate relative to GWO, the proposed technique achieved the target by enhancing the GWO with K-means. The proposed method's accuracy has reached 74.48% compared with GWO and PSO, which gained 72.15% and 73.31%, respectively, as shown in Fig. 27. The proposed technique achieved 57.17% in terms of detection rate, while GWO and PSO reached 52.77% and 54.89%, respectively, as shown in Fig. 28. Also in this figure, the proposed GWO-PSO-K-means got better and higher accuracy rate compared to other methods.

In terms of False Alarm, the proposed technique achieved 2.64% compared to GWO, which reached 2.24%, and PSO reached 2.35%, as shown in Fig. 29. In terms of processing time, the PSO achieved the best time, following by the GWO-PSO-K-means, and the GWO gives the longest process time, as shown in Fig. 30. Table 12 shows a comparison of GWO-PSO-SVM, GWO-SVM, and PSO-SVM algorithms in terms of accuracy, detection rate, false alarm, process time. As seen in Table 12, the proposed technique achieved the objective of enhancing the GWO and PSO with SVM, except for the false alarm rate relative to PSO. The proposed method achieved 98.97%, while GWO achieved 98.47%, and PSO gave 98.52% accuracy, as shown in Fig. 31.

The proposed technique achieved 98.58%, while GWO and PSO achieved 97.76% in detection rate, as seen in Fig. 32. Compared to GWO that reached 0.59%, the proposed technique gained 0.53%, and PSO achieved 0.46% in terms of false alarm terms, as seen in Fig. 33. In terms of processing time, the proposed technique achieved the best time, then the PSO, and finally, the GWO, as shown in Fig. 34.

Table 13 summarized all experiments results of GWO-PSO, GWO, and PSO using classification algorithms SVM

Table 6 The results of proposed technique based on the assessment metrics

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO-PSO-K-means	**0.7448**	**0.5717**	**0.0264**	**1648.0341**	**20**
GWO-PSO-SVM	**0.9897**	**0.9858**	**0.0053**	**1698.7152**	

Table 7 List of selected features by GWO

Method	Selected features	Total
GWO	2, 4, 5, 6, 8, 9, 11, 13, 18, 19, 21, 23, 25, 26, 27, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 41	**26**

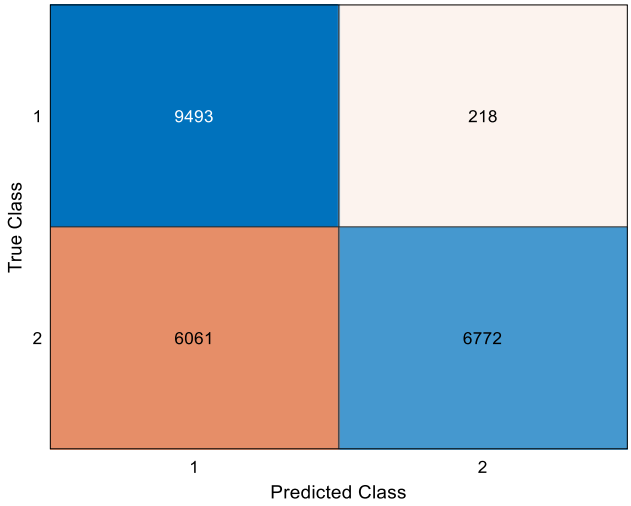


Fig. 22 Confusion chart of K-means using GWO selected features

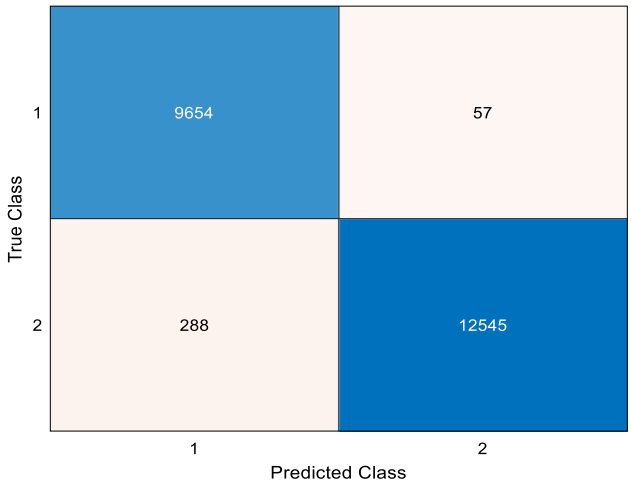


Fig. 23 Confusion chart of SVM using GWO selected features

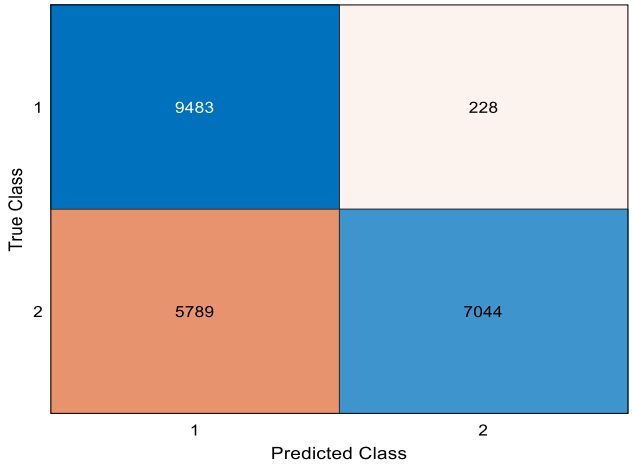


Fig. 24 Confusion chart of K-means using PSO selected features

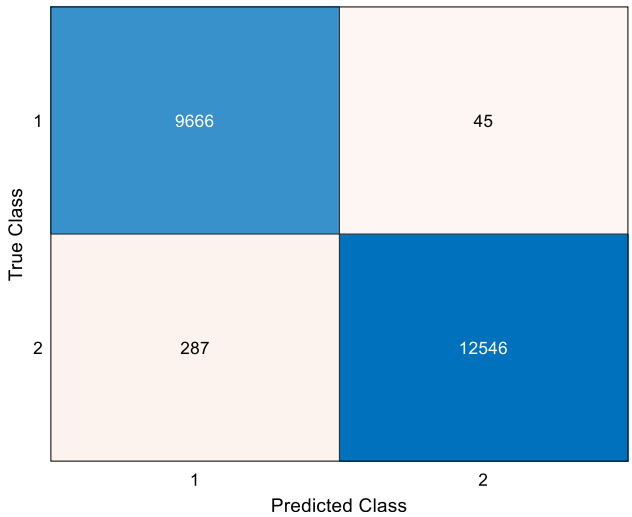


Fig. 25 Confusion chart of SVM using PSO selected features

Table 8 The results of GWO algorithm based on the assessment metrics

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO-K-means	**0.7215**	**0.5277**	**0.0224**	**1966.3061**	**26**
GWO-SVM	**0.9847**	**0.9776**	**0.0059**	**2343.6629**	

Table 9 List of selected features by PSO

Method	Selected features	Total
PSO	2, 3, 4, 8, 11, 12, 19, 23, 24, 25, 26, 28, 29, 34, 35, 38, 40, 30, 32, 33, 35, 36, 39, 41	**24**

Table 10 The results of PSO algorithm based on the assessment metrics

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
PSO-K-means	**0.7331**	**0.5489**	**0.0235**	**1437.1534**	**24**
PSO-SVM	**0.9852**	**0.9776**	**0.0046**	**1881.8307**	

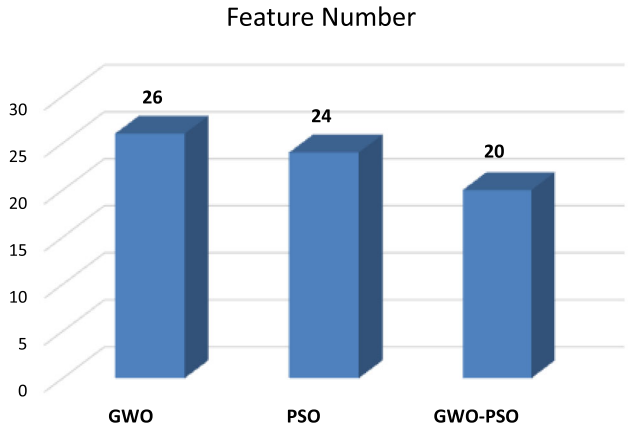


Fig. 26 Selected number of features by the comparative methods

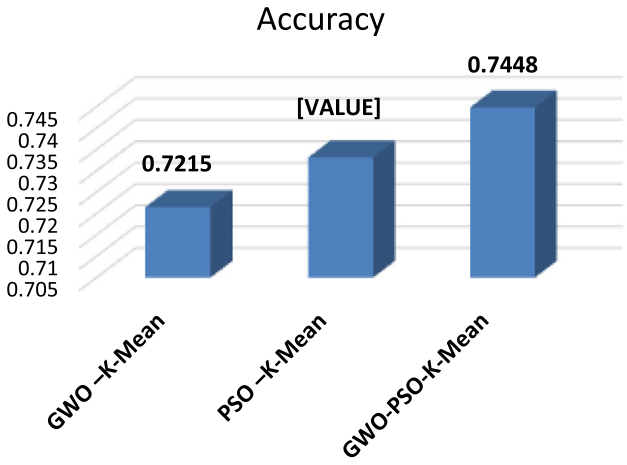


Fig. 27 Accuracy results of the comparative methods using K-means

and K-means in terms of accuracy, detection rate, false alarm, and process time. The proposed GWO-PSO-K-means got a smaller number of selected features similar to the proposed GWO-PSO-SVM. But, GWO-PSO-SVM got better results in terms of accuracy compared to all other methods in Table 13. Also in this table, other comparisons using the state-of-the-art methods published in the literature (i.e., Machine learning belief networks [48], Deep belief networks [49], and KELM [50]) are conducted to validate the performance of the proposed method. It is clear that the proposed method got better results compared to all other comparative methods. Which proved the ability of the proposed method in selecting the optimal features to determine the intrusion data.

The proposed technique's enhancement ratio on GWO-Kmeans, PSO-Kmeans, GWO-SVM, and PSO-SVM is given in this part of the results. The results are summarized in Tables 14 and 15. All values in the below tables are determined using this equation [51]: Enhancement

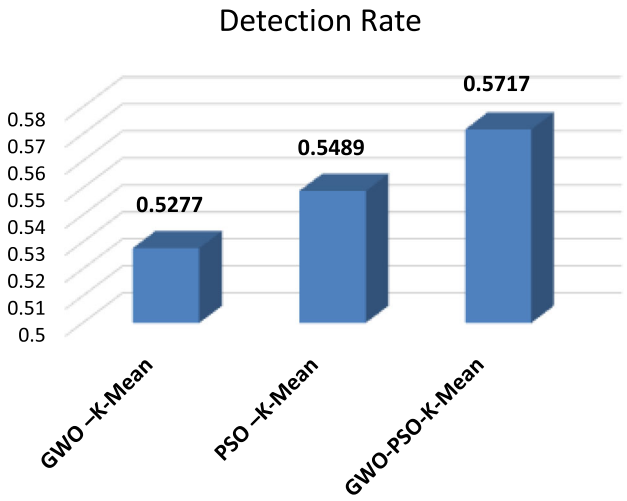


Fig. 28 Detection Rate of the comparative methods using K-means

percentage = (|Old value—New value|)/Old value as given in Eq. (19). As shown in Table 14, the proposed technique

Table 11 Comparison of GWO-PSO-K-means, GWO-K-means and PSO-K-means algorithms

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO-K-means	0.7215	0.5277	0.0224	1966.3061	26
PSO-K-means	0.7331	0.5489	0.0235	1437.1534	24
GWO-PSO-K-means	0.7448	0.5717	0.0264	1648.0341	20

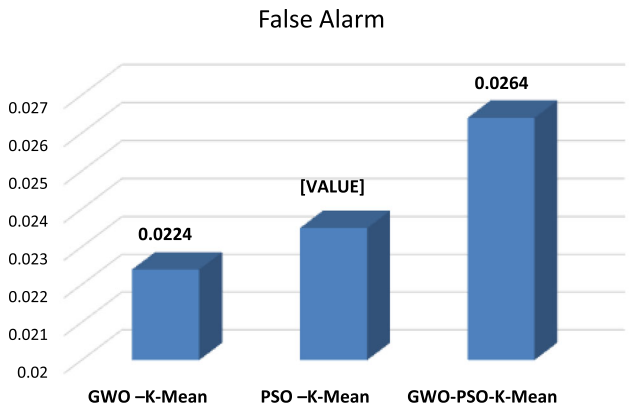


Fig. 29 False Alarm of the comparative methods using K-means

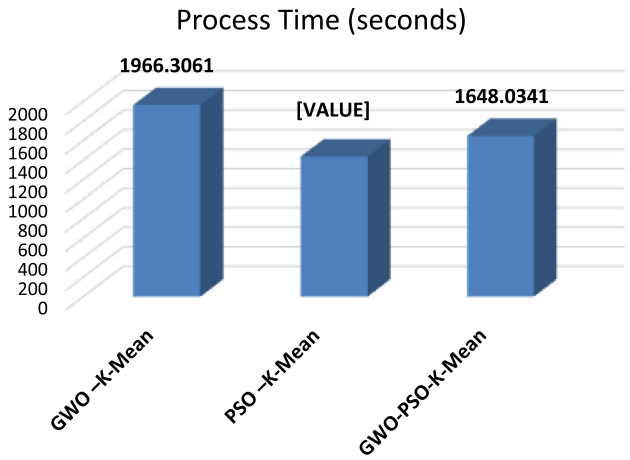


Fig. 30 Process Time (seconds) of the comparative methods using K-means

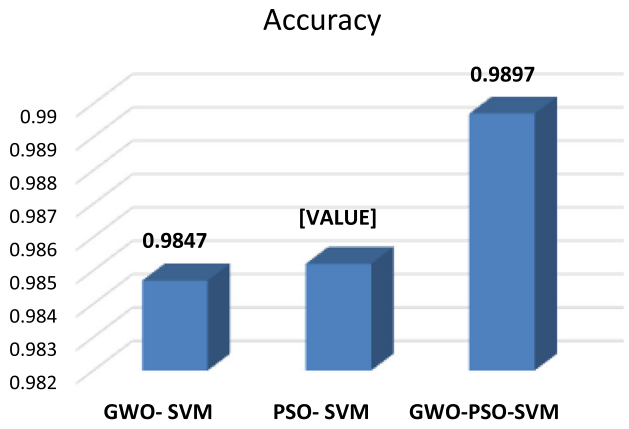


Fig. 31 Accuracy of the comparative methods using SVM

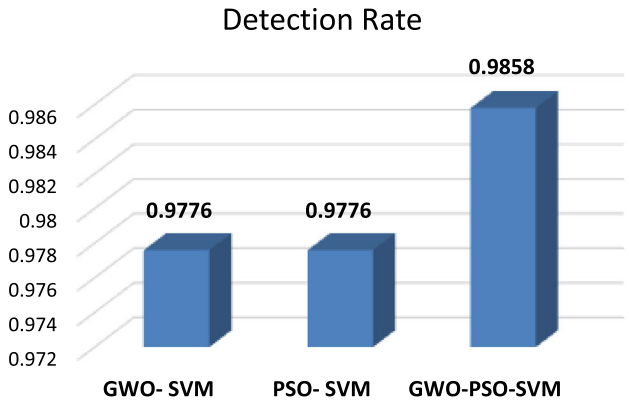


Fig. 32 Detection Rate of the comparative methods using SVM

enhanced the GWO-Kmeans by 3.229% accuracy, 8.338% in terms of detection rate, 16.186% in terms of processing time, and 23.076% in terms of feature number. Besides, the proposed technique enhanced the PSO-Kmeans by 1.596% in terms of accuracy, 4.153% in terms of detection rate, and 16.666% in terms of feature number; Fig. 35 illustrate these ratios.

As shown in Table 15, the proposed technique enhanced the GWO-SVM by 0.507% in terms of accuracy, 0.838% in terms of detection rate, 10.169% in terms of the false alarm, 27.518% in terms of processing time, and 23.076% in terms of feature number. Besides, the proposed technique enhanced the PSO- SVM by 0.456% in terms of accuracy, 0.8388% in terms of detection rate, 9.730% in terms of processing time, and 16.66% in terms of feature

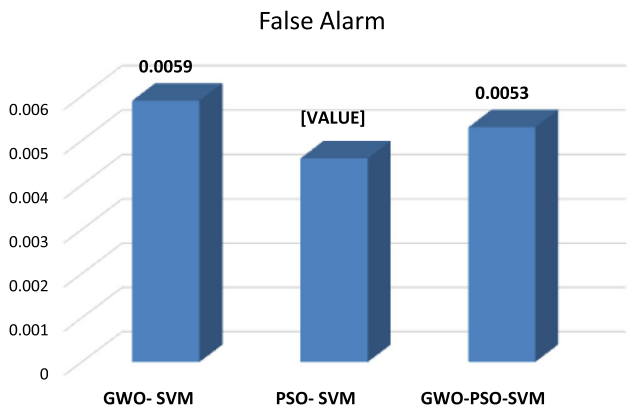


Fig. 33 False Alarm of the comparative methods using SVM

Table 12 Comparison of GWO-PSO-K-means, GWO-SVM and PSO-SVM algorithms

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO- SVM	**0.9847**	**0.9776**	**0.0059**	**2343.6629**	**26**
PSO- SVM	**0.9852**	**0.9776**	**0.0046**	**1881.8307**	**24**
GWO-PSO-SVM	**0.9897**	**0.9858**	**0.0053**	**1698.7152**	**20**

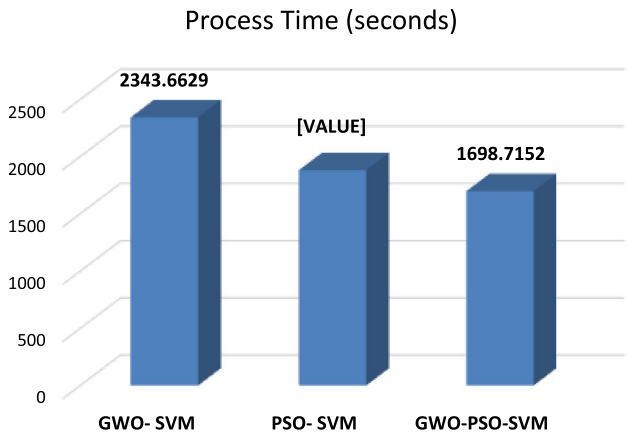


Fig. 34 Process Time of the comparative methods using SVM

number. These ratios are illustrated in Fig. 36. It is clear that the proposed method got better results and also new best solutions for the given problems.

5 Conclusions and future works

In this paper, GWO was improved by using hybridization with the PSO algorithm. Therefore, this improvement would be reflected in the level of protection of the IDS. The NSL KDD dataset was used to test the proposed technique in terms of accuracy, detection rate, false alarm rate, processing time, and the number of features. The results have shown this improvement by selecting the relevant features that have improved the classification process, whether K-means or SVM classification. The proposed technique was compared with the original PSO and GWO separately to measure this improvement. The results demonstrated that the proposed method outperforms the original PSO and GWO in terms of accuracy, detection rate, and the number of features. This technique enhanced the GWO-K means by 3.23% accuracy, 8.34% detection rate, 16.19% processing time, and 23.08% feature number. It also improved the PSO-Kmeans by 1.6% accuracy, 4.15% detection rate, and 16.67% feature number. For the SVM algorithm, the proposed technique enhanced the GWO-SVM by 0.51%

Table 13 The results of the proposed methods in terms of several evaluation measures

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO- SVM	0.9847	0.9776	0.0059	2343.6629	26
GWO –K-means	0.7215	0.5277	0.0224	1966.3061	26
PSO –K-means	0.7331	0.5489	0.0235	1437.1534	24
PSO- SVM	0.9852	0.9776	0.0046	1881.8307	24
GWO-PSO-SVM	0.9897	0.9858	0.0053	1698.7152	20
Machine learning belief networks [48]	–	88.10	–	–	–
Deep learning belief networks [49]	–	92.33		–	–
KELM [50]	–	94.01	–	–	–
GWO-PSO-K-means	0.7448	0.5717	0.0264	1648.0341	20

Table 14 The enhancements of GWO-PSO-Kmeans

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO-Kmeans	3.2294	8.3381	17.8571	16.1863	23.0769
PSO-Kmeans	1.5960	4.1538	12.3404	14.6735	16.6667

Table 15 The enhancements of GWO-PSO-SVM

Method	Accuracy	Detection rate	False alarm	Process time (seconds)	Feature number
GWO-SVM	**0.5078**	**0.8388**	**10.1695**	**27.5188**	**23.0769**
PSO-SVM	**0.4568**	**0.8388**	**15.2174**	**9.7307**	**16.6667**

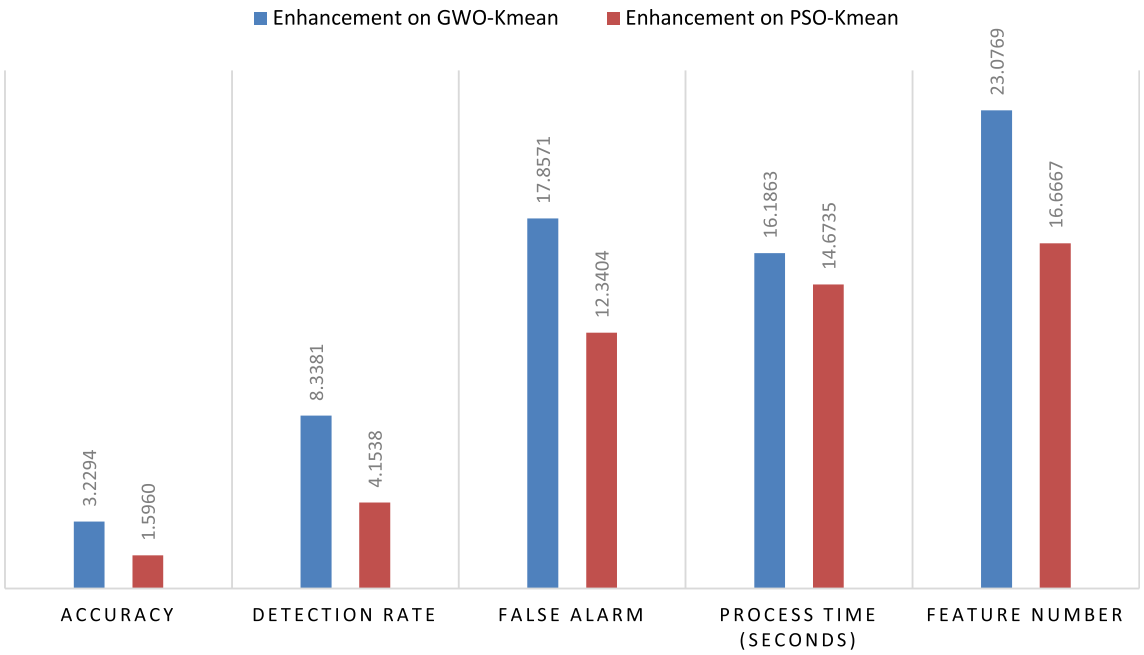


Fig. 35 The enhancements of GWO-PSO-Kmeans

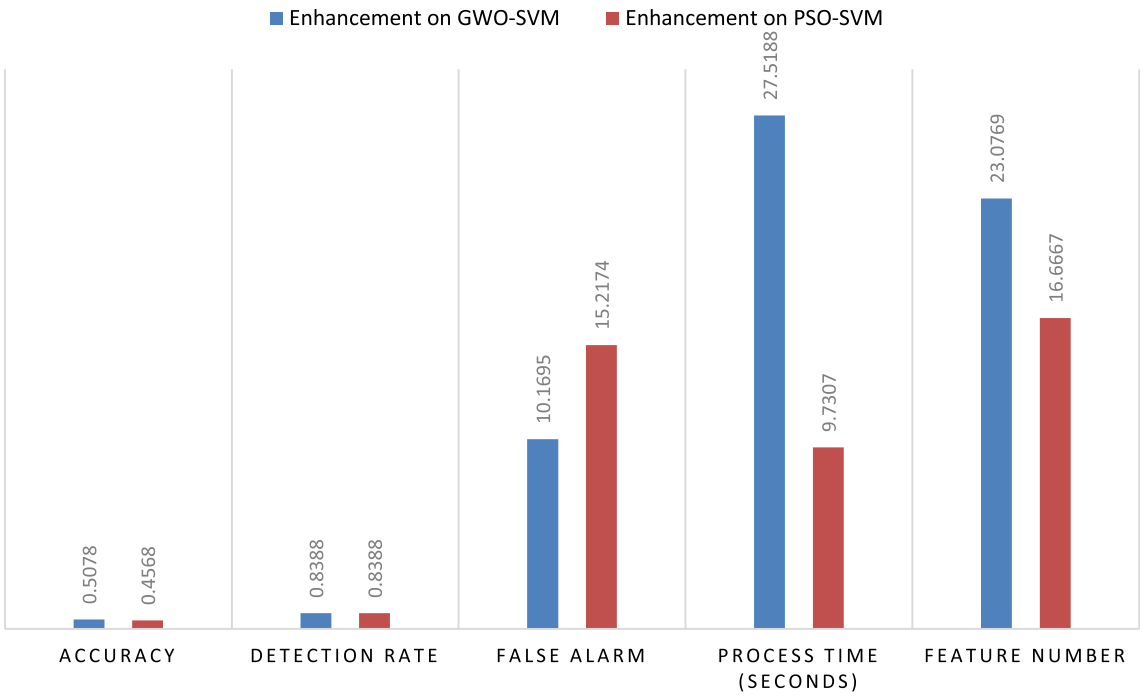


Fig. 36 The enhancements of GWO-PSO-SVM

accuracy, 0.84% detection rate, 10.17% false Alarm, 27.52% processing time, and 23.08% feature number. The proposed technique enhanced the PSO- SVM 0.46% accuracy, 0.84% detection rate, and 16.67% feature number. In the future, the Bagging (Bootstrap Aggregating) algorithm could be a classifier instead of K-means. It is one of the ensemble learning methods and can improve regression and classification accuracy to increase the detection rate in the WSN environment, especially IDS. Other optimization algorithms can solve the same problem, such as Arithmetic Optimization Algorithm (AOA). In future work, the proposed method can be applied to solve

other optimization problems such as data mining problems, task scheduling problems, wind energy problems, industrial engineering problems, benchmark function problems, feature selection problems, image segmentation problems and others.

Acknowledgements This study was financially supported via a funding grant by Deanship of Scientific Research, Taif University Researchers Supporting Project number (TURSP-2020/300), Taif University, Taif, Saudi Arabia

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors. No animal studies were carried out by the authors for this article. The used data will be avilaibe opon the request.

References

1. Ashton, K. (2009). That 'internet of things' thing. *RFID Journal* (on-line). Available: https://www.rfidjournal.com/that-internet-of-things-thing.
2. Abualigah, L., Diabat, A., & Elaziz, M. A. (2021). Intelligent workflow scheduling for Big Data applications in IoT cloud computing environments. *Cluster Computing*, *24*, 2957–2976. https://doi.org/10.1007/s10586-021-03291-7.
3. Singh, A., Nagar, J., Sharma, S., & Kotiyal, V. (2021). A Gaussian process regression approach to predict the k-barrier coverage probability for intrusion detection in wireless sensor networks. *Expert Systems With Applications, 172*, 114603.
4. Almomani, I., & Alromi, A. (2020). Integrating software engineering processes in the development of efficient intrusion detection systems in wireless sensor networks. *Sensors, 20*(5), 1375.
5. Ullo, S. L., & Sinha, G. R. (2020). Advances in smart environment monitoring systems using IoT and sensors. *Sensors, 20*(11), 3113.
6. Fahmy, H. M. A. (2020). *Wireless sensor networks: Energy harvesting and management for research and industry*. Springer.
7. Huo, G., & Wang, X. (2008). DIDS: A dynamic model of intrusion detection system in wireless sensor networks. In *2008 International Conference on Information and Automation* (pp. 374–378). IEEE.
8. Bace, R., & Mell, P. (2001). *NIST special publication on intrusion detection systems*. Booz-allen and Hamilton Inc MCLEAN VA.
9. Lu, M., & Reeves, J. (2014). Types of cyber attacks. *Trustworthy Cyber Infrastructure for the Power Grid, 18*, 2017.
10. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16–24.
11. Özgür, A., & Erdem, H. (2016). A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ Preprints, 4*, e1954v1.
12. Abualigah, L., & Diabat, A. (2020). A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. *Neural Computing and Applications*, 1–24.
13. Abualigah, L., & Diabat, A. (2021). Advances in sine cosine algorithm: A comprehensive survey. *Artificial Intelligence Review*, 1–42.
14. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2020). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering, 376*, 113609.
15. Singh, N., & Singh, S. B. (2017). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*, *2017*.
16. Singh, N. (2018). A modified variant of grey wolf optimizer. *Int J Sci Technol Sci Iran.* http://scientiairanica.sharif.edu.
17. Teng, Z. J., Lv, J. L., & Guo, L. W. (2019). An improved hybrid grey wolf optimization algorithm. *Soft Computing, 23*(15), 6617–6631.
18. Alrajeh, N. A., Khan, S., & Shams, B. (2013). Intrusion detection systems in wireless sensor networks: A review. *International Journal of Distributed Sensor Networks, 9*(5), 167575.
19. Safaldin, M., Otair, M., & Abualigah, L. (2020). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 1–18.
20. Islam, M. S., & Rahman, S. A. (2011). Anomaly intrusion detection system in wireless sensor networks: Security threats and existing approaches. *International Journal of Advanced Science and Technology, 36*(1), 1–8.
21. Tiwari, P., Saxena, V. P., Mishra, R. G., & Bhavsar, D. (2015). Wireless sensor networks: Introduction, advantages, applications and research challenges. *HCTL Open International Journal of Technology Innovations and Research (IJTIR), 14*, 1–11.
22. Ashoor, A. S., & Gore, S. (2011). Importance of intrusion detection system (IDS). *International Journal of Scientific and Engineering Research, 2*(1), 1–4.
23. Jyothsna, V. V. R. P. V., Prasad, V. R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications, 28*(7), 26–35.
24. Sadek, R. A., Soliman, M. S., & Elsayed, H. S. (2013). Effective anomaly intrusion detection system based on neural network with indicator variable and rough set reduction. *International Journal of Computer Science Issues (IJCSI), 10*(6), 227.
25. Al-Jarrah, O. Y., Siddiqui, A., Elsalamouny, M., Yoo, P. D., Muhaidat, S., & Kim, K. (2014). Machine-learning-based feature selection techniques for large-scale network intrusion detection. In *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (pp. 177–181). IEEE.
26. Chahal, J. K., & Kaur, A. (2016). A hybrid approach based on classification and clustering for intrusion detection system. *International Journal of Mathematical Sciences & Computing, 2*(4), 34–40.
27. Malviya, V., & Jain, A. (2015). An efficient network intrusion detection based on decision tree classifier & simple k-mean clustering using dimensionality reduction–a review. *International Journal on Recent and Innovation Trends in Computing and Communication, 3*(2), 789–791.
28. Shukla, V., & Vashishtha, S. (2014). New hybrid intrusion detection system based on data mining technique to enhanced performance. *International Journal of Computer Science and Information Security, 12*(6), 14.
29. Aljarah, I., & Ludwig, S. A. (2013). Mapreduce intrusion detection system based on a particle swarm optimization clustering algorithm. In *2013 IEEE Congress on Evolutionary Computation* (pp. 955–962). IEEE.
30. Duque, S., & Bin Omar, M. N. (2015). Using data mining algorithms for developing a model for intrusion detection system (IDS). *Procedia Computer Science, 61*, 46–51.

31. Li, Z., Li, Y., & Xu, L. (2011). Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization. In *2011 International Conference of Information Technology, Computer Engineering and Management Sciences* (Vol. 2, pp. 157–161). IEEE.
32. http://wiki.analytica.com/Optimization_Characteristics
33. Abd Rahman, M. A., Ismail, B., Naidu, K., & Rahmat, M. K. (2019). Review on population-based metaheuristic search techniques for optimal power flow. *Indonesian Journal of Electrical Engineering and Computer Science, 15*(1), 373–381.
34. NSL-KDD Dataset. (n.d.). *Canadian Institute for Cybersecurity*. https://www.unb.ca/cic/datasets/nsl.html
35. Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering, 4*(6), 446–452.
36. Dash, T. (2017). A study on intrusion detection using neural networks trained with evolutionary algorithms. *Soft Computing, 21*(10), 2687–2700.
37. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.
38. Guo, M. W., Wang, J. S., Zhu, L. F., Guo, S. S., & Xie, W. (2020). An improved grey wolf optimizer based on tracking and seeking modes to solve function optimization problems. *IEEE Access, 8*, 69861–69893.
39. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE.
40. Prabha, K. A., & Visalakshi, N. K. (2014). Improved particle swarm optimization based k-means clustering. In *2014 International Conference on Intelligent Computing Applications* (pp. 59–63). IEEE.
41. Umar, R., Mohammed, F., Deriche, M., & Sheikh, A. U. (2015). Hybrid cooperative energy detection techniques in cognitive radio networks. *Handbook of research on software-defined and cognitive radio technologies for dynamic spectrum management* (pp. 1–37). IGI Global.
42. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, No. 14, pp. 281–297).
43. Morissette, L., & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology, 9*(1), 15–24.
44. Younus, Z. S., Mohamad, D., Saba, T., Alkawaz, M. H., Rehman, A., Al-Rodhaan, M., & Al-Dhelaan, A. (2015). Content-based image retrieval using PSO and k-means clustering algorithm. *Arabian Journal of Geosciences, 8*(8), 6211–6224.
45. Osuna, E., Freund, R., & Girosi, F. (1997). An improved training algorithm for support vector machines. In *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop* (pp. 276–285). IEEE.
46. Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection: support vector machines and neural networks. In *proceedings of the IEEE International Joint Conference on Neural Networks (ANNIE), St. Louis, MO* (pp. 1702–1707).
47. Tharwat, A. (2019). Parameter investigation of support vector machine classifier with kernel functions. *Knowledge and Information Systems, 61*(3), 1269–1302.
48. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Mingcheng, G., Haixia, H., & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee Access, 6*, 35365–35381.
49. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41–50.
50. Ghasemi, J., Esmaily, J., & Moradinezhad, R. (2020). Intrusion detection system using an optimized kernel extreme learning machine and efficient features. *Sādhanā, 45*(1), 1–9.
51. Odat, A., Otair, M., & Shehadeh, F. (2015). Image denoising by comprehensive median filter. *International Journal of Applied Engineering Research, 10*(15), 36016–36022.

Mohammed Otair has a Ph.D. in Computer Information System/ Artificial Intelligence. His teaching experience includes information system courses over 22 years to University students. He has been working at Amman Arab University since 2010. He held different leading positions at the university: head for several departments (Computer Science, Computer Information System, and Software Engineering) and computer center manager in two universities, also he has an experience in management of projects. He is the author of 73 scientific publications including an organizer and a participant of more than 28 international scientific conferences and more than 45 international journals (most of them are about E-learning, Mobile Learning, and its Infrastructure). He supervised 32 master theses in Computer Science field.

Osama Talab Ibrahim has a Master degree in computer science and AI from Amman Arab University. His main interests are feature selection, optimization algorithms, and data mining.

Laith Abualigah is an Assistant Professor at the Computer Science Department, Amman Arab University, Jordan. He is also a distinguished researcher at the School of Computer Science, Universiti Sains Malaysia, Malaysia. He received his first degree from Al-Albayt University, Computer Information System, Jordan, in 2011. He earned a Master's degree from Al-Albayt University, Computer Science, Jordan, in 2014. He received a Ph.D. degree from the School of Computer Science in Universiti Sains Malaysia (USM), Malaysia, in 2018. According to the report published by Stanford University in 2020, Abualigah is one of the 2% influential scholars, which depicts the 100,000 top scientists in the world. Abualigah has published more than 100 journal papers and books, which collectively have been cited more than 4400 times (H-index = 32). His main research interests focus on Arithmetic Optimization Algorithm (AOA), Bio-inspired Computing, Nature-inspired Computing, Swarm Intelligence, Artificial Intelligence, Meta-heuristic Modeling, and Optimization Algorithms, Evolutionary Computations, Information Retrieval, Text clustering, Feature Selection, Combinatorial Problems, Optimization, Advanced Machine Learning, Big data, and Natural Language Processing. Abualigah currently serves as an associate editor of the Journal of Cluster Computing (Springer), the Journal of Soft Computing (Springer), and Journal of King Saud University - Computer and Information Sciences (Elsevier).

Maryam Altalhi works in the Management Information System Department of Taif University, Saudi Arabia. She completed her Ph.D. inInformation Systems from University Technology Malaysia, Malaysia in 2017. She also holds a Master of Management Information System from La Trobe University, Australia (2013) and a Bachelor of Computer Science fromTaif University, Saudi Arabia (2006). Dr. Maryam MutiAltalhi has many published worksin reputable journals.

Putra Sumari received the M.Sc. degree in computer sciences specifically in software engineering and video processing from Liverpool University in 1996 and the Ph.D. degree in computer sciences specifically in software engineering and video processing from Liverpool John Moores University, England, in 2001. From 2006 to 2008, he was a Visiting Fellow with the Department of Computer Sciences, National University of Singapore. He is a Professor with the School of Computer Science, University of Science, Malaysia. His research interests include image and video analysis for medical, video on demand systems, and watermarking and compression applications. He was also interested in the use of brain signals for computer application.