Journal of Ambient Intelligence and Humanized Computing (2021) 12:1559–1576
https://doi.org/10.1007/s12652-020-02228-z

ORIGINAL RESEARCH

Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks

Mukaram Safaldin[1] · Mohammed Otair[1] · Laith Abualigah[1]

Received: 16 January 2020 / Accepted: 13 June 2020 / Published online: 26 June 2020

Abstract

Intrusion in wireless sensor networks (WSNs) aims to degrade or even eliminating the capability of these networks to provide its functions. In this paper, an enhanced intrusion detection system (IDS) is proposed by using the modified binary grey wolf optimizer with support vector machine (GWOSVM-IDS). The GWOSVM-IDS used 3 wolves, 5 wolves and 7 wolves to find the best number of wolves. The proposed method aims to increase intrusion detection accuracy and detection rate and reduce processing time in the WSN environment through decrease false alarms rates, and the number of features resulted from the IDSs in the WSN environment. Indeed, the NSL KDD'99 dataset is used to demonstrate the performance of the proposed method and compare it with other existing methods. The proposed methods are evaluated in terms of accuracy, the number of features, execution time, false alarm rate, and detection rate. The results showed that the proposed GWOSVM-IDS with seven wolves overwhelms the other proposed and comparative algorithms.

Keywords Intrusion detection system · Binary grey wolf optimizer · Wireless sensor networks · Intrusions · NSL-KDD

1 Introduction

Wireless sensor networks (WSNs) is a heterogeneous system containing small actuators and sensors which have general-purpose computing elements. WSN consists of hundreds or thousands of low-cost wireless nodes, low-power, self-organizing, and deployed to control and monitor the environment (Abdollahzadeh and Navimipour 2016; Rashid and Rehmani 2016). Five main characteristics have to be considered during the development of WSN: self-healing, reliability, scalability, robustness, and security (Emary et al. 2016). Moreover, WSNs can be used in many applications like ocean monitoring, performance monitoring for manufacturing machines, earthquake monitoring, and various military applications. Besides, future applications, such as monitoring of pollution, highway traffic, building security, wildfires, and water quality, are likely to use WSN principles in their architectures. WSNs have many advantages, such as converting raw data into useful aggregated and grouped information (Maleh and Ezzati 2015).

WSNs often have one (or more) centralized control unit called base station(s). A base station is typically playing a gateway role to another network, it provides a great storage center and data processing, and it offers human interface access point. It can also be used as a connector to disseminate control information and to extract data from the network. The base station has also referred to the sink. All of the sensor nodes build a routing forest, where the root of each tree is a base station. The base station has more storage and power capacity than the other sensor nodes. Typically, the base station has the required battery energy to cover sensor nodes lifetime, enough memory storage to save cryptographic keys, stronger (in terms of performance and speed) processors, and means to communicate with outside WSN (Khasawneh et al. 2020; Çavuşoğlu 2019).

Ensuring a high level of surety for the critical applications that use WSN is highly required to secure their data and infrastructure from intrusions. Therefore, an intrusion detection system (IDS) should be used to detect abnormal behaviors and intrusions. In WSN, sensors gather the information from the environment where they are distributed, and then

✉ Laith Abualigah
Aligah.2020@gmail.com

Mukaram Safaldin
malqadiry@yahoo.com

Mohammed Otair
otair@aau.edu.jo

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

they transmit it to the base station node. Information should be secured from external attackers, as cryptographic security is not fully efficient in protecting this information. Therefore, the second level of defense mechanism is required, such as IDS (Guo et al. 2014; Khor et al. 2012). IDS monitor the network's traffic and sends alerts to the base station in case there is a malicious activity found by any sensor as shown in Fig. 1. In this figure, the black cycle presents a sensor node, white star presents an intruder, red star presents a cluster head, *D* presents the distance between to intruder, *R* presents the cluster area, and *A* presents the sensors area.

Feature selection (FS) is a technique used to select the most pertinent features, where these features are used to create accurate and robust IDS models. FS offers a manner to identify the relevant features and removing redundant features among given attributes from the dataset (Chizi et al. 2009; Abualigah 2019). The main objectives of FS are minimizing data dimensionality and improving the performance of the detected performance (Chandrashekar and Sahin 2014). In real applications, there are many features used in data representation, including redundant features. Thus specific features would take the role of other features, and the extra features might be segregated. In addition, the relevant features have a direct impact on the output, and they have essential information describing the behavior of the dataset (Bell and Wang 2000). Formerly, it is unpractical to get a comprehensive search for the best set of features in a high dimensional space. Many kinds of research have been conducted to model the FS as a combinatorial optimization problem, where a set of given features leads to the optimal feature space (Nakamura et al. 2012).

Despite that several IDSs have been proposed to detect intrusions in WSN such as (Haque et al. 2015; Jin et al. 2017; Yu et al. 2016), they are still suffering from: (1) low detection accuracy, (2) high false alarm alert, and (3) high processing time. Therefore, proposing more efficient IDS in WSN is needed. The Grey Wolf Optimization (GWO) is a recent optimization algorithm, proposed by Mirjalili et al. (2014), that has been successfully used for solving different FS problems. In this paper, three binary improved versions of GWO are introduced using two different techniques. In the first technique, the first three best solutions are recognized by the individual, and then they are binarized, after that, the stochastic crossover is done between the three main moves in order to find the next location for the binary gray wolf. In the second technique, frequent update to wolves' location is performed by sigmoidal function, and these values follow a threshold stochastically in order to find the next location of the binary gray wolf. This paper enhance the performance of GWO-IDS in WSN by increasing the number of wolves and using multi-objective function to increase the overall performance of the prediction system is based on how it predicts unknown classes efficiently in terms of accuracy and the total number of selected features, and speed-up the execution. These two techniques of binary GWO are commonly used in the FS domain to find an optimal subset of features used to maximize the classification accuracy (i.e., detection rate) with the minimum number of features used. Experiments are conducted using the NSL KDD'99 dataset to demonstrate the performance of the proposed method and compare it with other existing techniques including PSO-IDS (since it suffers from long execution time and low detection rate) and grey wolf optimizer with support vector machine (GWOSVM-IDS) with three wolves (since it suffers from long execution time and low accuracy). The proposed methods are evaluated in terms of accuracy, number of features, execution time, false alarm rate, and detection rate. The results showed that the proposed GWOSVM-IDS with five wolves overwhelm PSO-IDS by 3, 40, 42, 64, and 3%, and it enhances GWOSVM-IDS with three wolves by 16, 50, 13, 60, and 14% respectively. Besides, the proposed GWOSVM-IDS with seven wolves enhances PSO-IDS by 7, 40, 46, 88, and 15% respectively and GWOSVM-IDS with 3 wolves by 21, 50, 24, 87 and 15% respectively. Therefore, the proposed technique enhances the existing techniques noticeably in terms of accuracy, detection rate, number of features, and execution time.

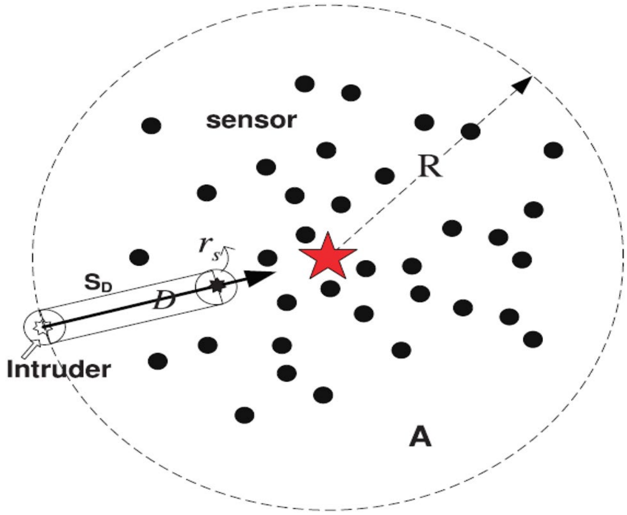


Fig. 1 Example of the intrusion detection system in wireless sensor networks

The results of this paper are organized as: Sect. 2 presents the more related work to the intrusion detection using optimization algorithms. Section 3 presents the proposed method for solving intrusion detection. Section 4 presents experiments and results. Finally, Sect. 5 presents the conclusion and future works.

2 Literature review

In this section, a comprehensive review of the related works is presented to investigate the existing works related to feature selection based on IDS. Also, a comparison of the proposed technique is conducted against some of the prior techniques in order to validate the result of the proposed one.

2.1 An overview of wireless sensor networks

As aforementioned, WSN holds several devices called sensor nodes. These nodes are characterized by their small size and their limited power and processing capabilities. Besides, sensor nodes sense and gather information from the environment that is distributed (García-Hernández et al. 2007). Although, the WSN is used in many critical applications, it is still suffering from different challenges. These challenges are discussed in the next subsection.

2.1.1 WSNs challenges

WSN poses unique challenges because it deploys inaccessible (open) areas, presented in the added risks of physical attack; also the interaction with their users and physical environments may carry serious security problems (Mahmood et al. 2015; Curiac 2016). So, the traditional security mechanisms are not enough and inadequate to protect WSNs, and new ideas are needed. Indeed, many security issues may face WSNs, like (Guo et al. 2014). Data confidentiality is the process of keeping the information secure from unauthorized parties. It can be achieved since a network sensor does not trickle sensor readings to the sensors neighboring in WSNs. In different applications, network nodes highly communicate sensitive and critical data. Data encryption is the standard approach that can be used to protect sensitive data from unauthorized access. Public-key cryptography is used in data encryption used for encrypting the data. On the other hand, this approach is costly in terms of time and performance.

In WSNs, the receiver node has to make sure that form the originator of the message before making any decision because intruders can easily inject messages in the network. Therefore, data authentication protects the network from unauthorized parties' participation in the network. In the case of two-party communication, data authentication can be done by using a purely symmetric mechanism. Data Integrity verifies that the receiver will receive an un-altered message, as an adversary can alter the message during the transmission of the message. Moreover, data authentication offers data integrity. Data freshness implies that the data is recent, and there are no replayed old messages by an adversary. Sensor network encryption protocol (SNEP) uses a common defense by using generating unique and order number for each message (Nakamura et al. 2012). This method ensures that the message with the first occurrence must be passed, and the fake message with more than one occurrence must be prevented. WSN should be robust and robust against various attacks. However, if an attack succeeds to enroll in the network, its impact must be decreased as much as possible. WSNs are vulnerable and susceptible to security attacks due to the medium of message transmission and the broadcasting nature. Furthermore, because of the WSNs nodes are placed in a hostile and dangerous environment, WSNs face an additional vulnerability.

2.1.2 WSN threats and attacks

In this section, these threats and attacks are discussed (Chelli 2015). The intruder can receive the messages interception, which is containing the sensor nodes' physical locations. After that, an attacker can destroy the nodes after locating them. An adversary can observe the sensor nodes' locations, the content of messages such as: (time-stamps, and message ID) (Hammoudeh et al. 2017). Robust encryption techniques can be used to reduce the threats of gathering passive information.

Capturing the network node may reveal this node information, including cryptographic keys, which can compromise the whole network. The intruders could add a fake node into the network to feed wrong or false data or to prevent accurate data passaging (Pritchard et al. 2017). Moreover, such messages consume resource energy and slow down network performance. This type of attack could be called "sleep deprivation." A single node in the network has a present multi identity to the other nodes inside the WSN. It poses a critical threat to the protocol of geographic routing, as the location-based routing requires exchanging coordination information with the node and its neighbors to address the data packets geographically to the route efficiently. Authentication and encryption techniques can be used to protect the network from Sybil attack in WSNs. Every two nodes in WSNs can employ Needham-Schroeder protocol in order to verify their identities and to build public key. A neighboring node pair can use the generated key in order to implement an encrypted and authenticated link between them. A sinkhole attack attracts a considerable number of the transmission traffic by a specific node through a specific area, and it constructs fake road or sinkhole at the network center (Maza and Touahria 2019). A sinkhole attack typically makes a particular node that seems more so attractive for its neighbors.

An adversary in wormhole attack tunnels messages which is arrived at the receiver over a low latency network link and replays it to different parts. In this type of attack, between every two nodes, there is single that node has been situated to forward messages between them. Figure 2 presents

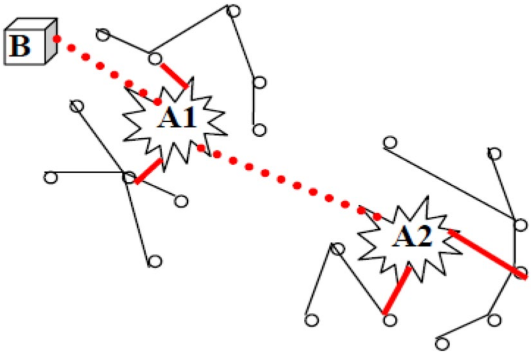


Fig. 2 Wormholes attack (Chelli 2015)

a wormhole example. From Fig. 2, it can be noticed that the attacker receives packets at one point in the network, and then it tunnels them to another point in the network and replays them into the network from that point.

2.2 Intrusion detection systems based on feature selection

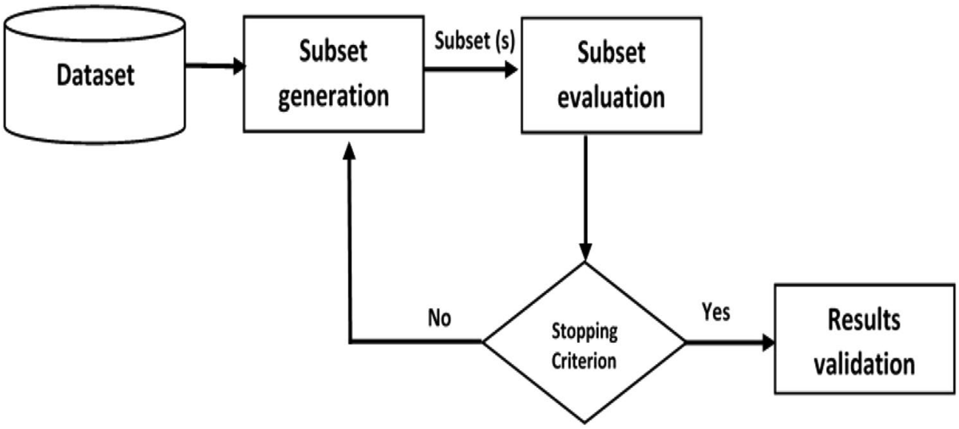
As mentioned earlier in chapter one, the IDS is an essential element of security over any network type. It ensures a high level of security for the network against possible threats by preventing or detecting all intrusions and hosts (Devi and Suganthe 2017). However, its primary goal is to ensure the adaptability to detect any new type of attack. IDS can be classified into misuse IDS and anomaly IDS. The misuse IDS employs the signatures to find any new attack, but the anomaly IDS uses statistical patterns and intelligent methods to discover either the behavior is healthy or not (Stein et al. 2005; Amiri et al. 2011; Paulauskas and Auskalnis 2017).

Several techniques have been proposed to introduce IDS are based on anomaly detection, mainly, these techniques use intelligent classification methods and artificial intelligence algorithms to classify the normality and abnormality (Aljawarneh et al. 2018). The classifier in IDS ensures the detection of attacks and provides intelligent computation to the detection process. Every classifier might be characterized by its pattern, detection rate, false alarm rate, and accuracy.

Regrettably, most of the classifiers suffer from the low level of accuracy, detection rate, or high level of false alarm rate, in addition to the issue related to the complexity of the classifier model and computational cost. Moreover, the high level of classification, high complexity, long processing time, and massive storage capacity impact the performance of the detection process (Jaiganesh et al. 2013). Therefore, enhancing the classifier performance and minimizing the processing costs remain the main issues in IDS and need further improvement. However, reducing the dimensionality of the dataset would enhance the effectiveness of the detection process and overcome the overhead classification challenge. Recently, the feature selection process is used to overcome such problems (Ambusaidi et al. 2016).

Feature selection is emerging to minimize the dimensionality of data by selecting the essential features with minimum redundancy and maximum relevancy to get high performance and accuracy. Those are the main objectives that provide us with suitable data recognizing out noise, addressing the over-fitting issues, and select in the optimal feature that could be used in the classification process efficiently. Feature selection can be defined as “a preprocessing optimization process, which is mainly used to reduce the dimensionality of the dataset by selecting the interesting features without redundancy and irrelevance” (Bins and Draper 2001). These selected features resemble the optimal subset which provide an enhanced classification performance and accuracy, avoidance of the overhead of classification issue, a minimized processing time and storage capacities required to store data, a data out noise, and no over-fitting problem (Stein et al. 2005). Figure 3 illustrates the main steps of the feature selection process, which include subset generation

Fig. 3 Feature selection processes



from the dataset, evaluation of this subset, then result invalidation after stopping criteria has met.

2.3 Related works

A few related works in prior literature consider anomaly detection in WSN or propose detection techniques to address malicious attacks and intrusions in WSN. However, an IDS technique is proposed for WSN based on trust values (Jin et al. 2017). The features of the technique are as follows: (1) Mahalanobis distance is used to judge abnormalities for the node trust feature, which provides more accuracy. (2) To reduce the false-positive rate, a tolerance factor is used in the trust value computations. Thus the WSN security and flexibility are improved. (3) The proposed method is implemented using a multi-agent model to ensure scalability and fault tolerance. The findings revealed that the modified technique ensures a high detection rate and minimal false-positive rate.

A new anomaly detection model is introduced based traffic WSN (Yu et al. 2016). The proposed technique ensures a high level of detection accuracy by frequently updated model and static size sliding window; also, the exponentially weighted technique is used to enhance the detection rate, and using a reference and delay technique also increases the detection rate.

A hybrid IDS for WSN, and their technique is working based on hybridizing two existing techniques (i.e., anomaly and signature) (Maleh and Ezzati 2015). Hybridizing these two techniques provide an IDS with a 94% detection rate. Since these two models are integrated into a cluster topology in order to minimize communication costs between nodes, the lifetime of WSN is improved. In addition, their technique takes advantage of SVM and signature-based techniques to detect suspicious behaviors and to offer a lightweight intrusion detection in cluster topology. This technique employs a discovery protocol depending on the specifications in order to detect intruder's nodes and to prohibit network disruptions. In specific, the role of this protocol is to classify the behavior into normal behavior or abnormal behavior based on a set of rules. However, this research ignores the other essential factors, such as accuracy, number of features, and execution time.

In addition, a new IDS is proposed in order to detect malicious attacks by classifying true alarms from false alarms, where this technique is applied in healthcare applications (Haque et al. 2015). Their technique is working based on historical data to predict sensor value, and then this value is directly compared with actual sensed value. Majority voting and dynamic threshold are used to identify alarm type. Their technique is evaluated using medical datasets. The findings show that their technique has a high detection rate (91%) and a low false-positive rate (16%). However, still, this technique needs more improvement in terms of execution time, number of features, and accuracy.

The PSO-IDS technique is introduced along with principal component analysis to detect an attack in the WSN environment (Ahmad 2015). The proposed technique obtained noticeable results in terms of the number of selected features and false alarm rate, but it requires more enhancements in terms of accuracy and detection rate. Besides, the execution time required to detect all attacks is totally ignored when proposing this technique. Besides, Sedjelmaci and Feham (2011) proposed a hybrid IDS technique in the WSN environment. Misuse IDs and SVM have been used to detect attacks. The proposed technique is evaluated using the NSL KDD'99 dataset, and the results revealed that the proposed technique achieves high accuracy (96%) and detection rate (93%), but it ignores false alarm rate and execution time. Other optimization algorithms can be investigated to solve the IDS (Abualigah and Khader 2017; Abualigah et al. 2018; Abualigah 2020; Abualigah and Diabat 2020).

As noticed from Table 1, the existing IDS techniques are still suffering from different issues, such as: high false alarm rate, low detection rate, high number of selected features, and/or high processing time. Therefore, this paper is conducted to overcome all these issues by proposing a modified binary GWO (GWOSVM-IDS) to detect intrusions in WSN environment efficiently.

Table 1 Overview of the related works

References	False alarm rate	Accuracy	Detection rate	No. of features	Execution time
Jin et al. (2017)	Low	Low	Very high	NA	NA
Yu et al. (2016)	Low	Low	Very high	NA	High
Maleh and Ezzati (2015)	Very low	High	Very high	NA	NA
Haque et al. (2015)	Low	High	Very high	NA	NA
Ahmad (2015)	High	High	Low	Very high	NA
Benmessahel et al. (2018)	Very high	Very high	Very high	Low	NA
Proposed technique (GWOSVM-IDS)	Very low	Very high	Very high	Very low	Very low

NA not addressed

3 The proposed method

In this section, a new intrusion detection method is proposed to achieve the main objective of this study. As shown in Fig. 4, typically, the quantitative method starts with data acquisition, where the dataset is acquired from NSL-KDD dataset, because it is a verified and valid data set (Haque et al. 2015; Maleh and Ezzati 2015), which is extensively used in testing different intrusion detection technique.

Figure 4 shows the steps of the proposed method that deals with the dataset to detect intrusions in WSN. The dataset is divided into two main parts with different sizes (i.e., training dataset with 80% size from the overall dataset and testing dataset with 20% size from the overall dataset). Generally, the proposed method as shown in Fig. 4 is divided into three main stages, the first stage is the future selection applied on the given data, the second stage is the classification process for the data, and the third stage is the evaluation stage to prove the performance of the applied methods. The following sub-sections describe the stages of the proposed method in detail.

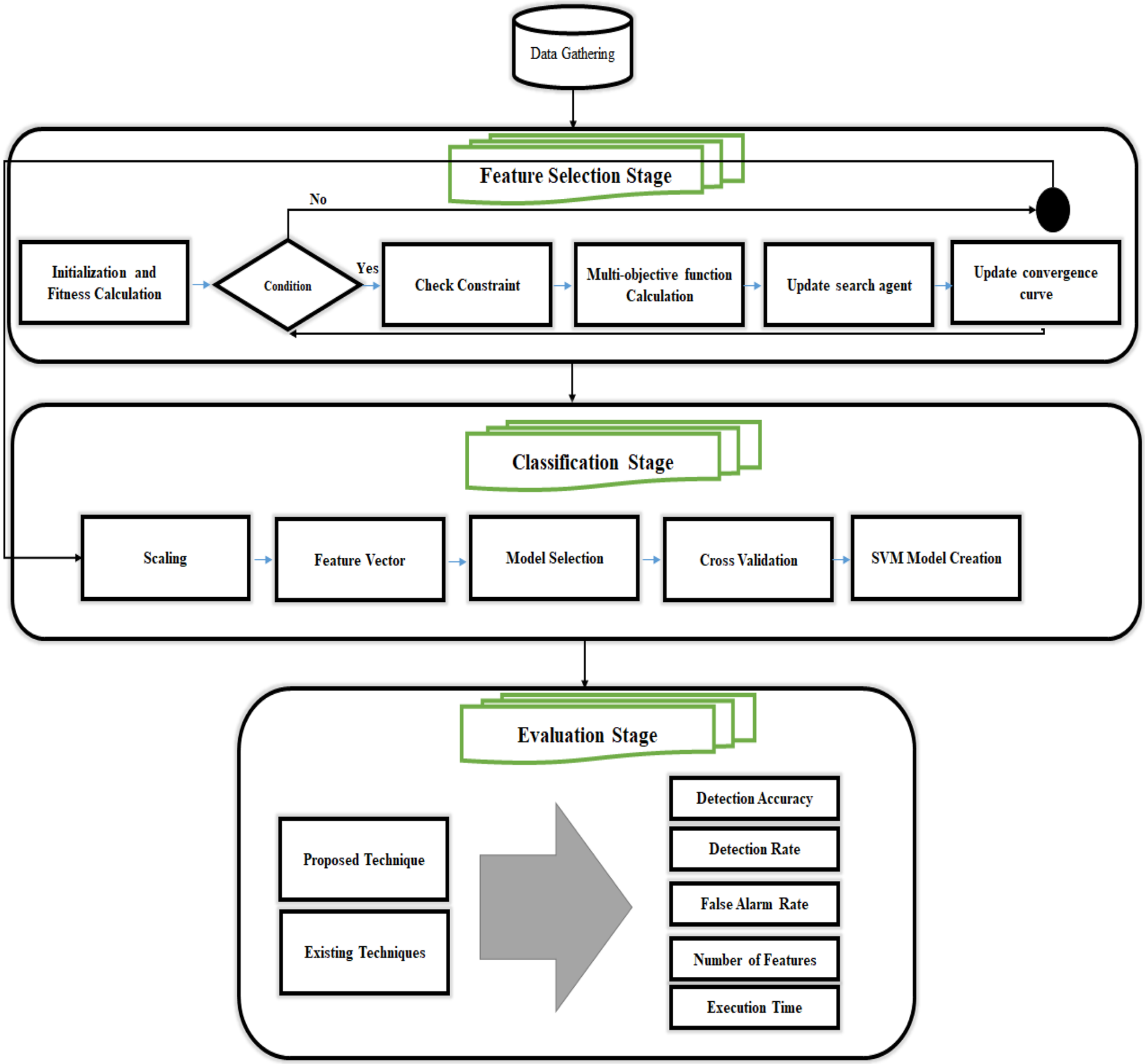
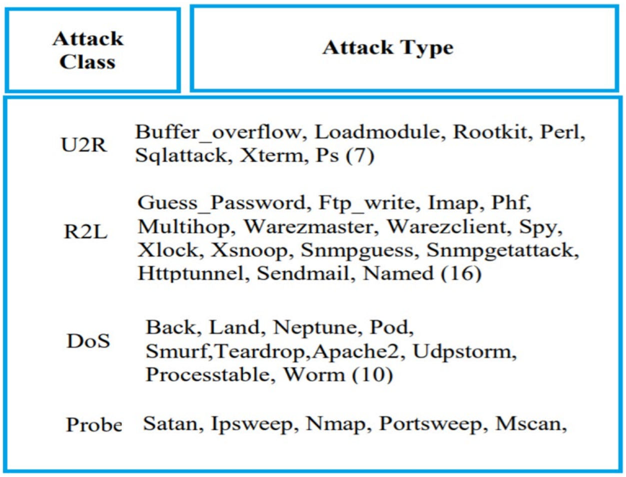


Fig. 4 Procedures of the proposed intrusion detection system

Table 2 NSL KDD'99 statistics

	Original records	Distinct records	Reduction rate (%)
Attack	3,925,650	262,178	93.32
Normal	972,781	812,814	14.44
Total	4,898,431	1,074,992	78.05



Attack Class	Attack Type
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps (7)
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named (16)
DoS	Back, Land, Neptune, Pod, Smurf,Teardrop,Apache2, Udpstorm, Processtable, Worm (10)
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan,

Fig. 5 Types of attacks in NSL-KDD'99

3.1 Preparing dataset

In this section, the network security laboratory dataset is used (NSL-KDD'99) to evaluate the proposed technique. It is an open-source dataset, where it can be downloaded online. Indeed, this dataset is an enhanced version of the KDD-Cup 99 (Tavallaee et al. 2009). Several kinds of data analyses have been conducted on the NSL-KDD'99 dataset using different methods and tools with a global security aim, which is to develop active IDS. It has been used in this research due to several advantages such as no redundant records are included; therefore, no biased classification occurs. Indeed, it always provides more intrusion detection rates; and it can be used to compare different IDS techniques effectively. Table 2 summarizes statistics of NSL KDD'99 related to the number of attacks and standard records in the dataset.

The number of original records of NSL KDD'99 is 4,898,431 (3,925,650 attacks records and 972,781 official records). The filtering and reduction process is performed to increase the reliability, validity, and accuracy of this dataset by removing redundant and biased records. The number of distinct records that are eliminated from the refined version of NSL KDD'99 is 1,074,992 (262,178 attacks records and 812,814 standard records). In other words, the total reduction percentage is 78.05% (93.32% reduction in attacks records and 16.44% in standard records).

Besides, Fig. 5 depicts the attacks classes included in the dataset. These classes include: U2R attacks (user to root attacks where the attacker has local access to victim machine, hence tries to gain access), R2L attacks (remote to local attack where the attacker does not have an account on the victim machine, hence tries to gain access), DoS attacks (denial of service attack where the attacker tries to prevent legitimate users from using a service), and Probe attacks (attacker tries to gain information about the target host).

Data normalization has been performed on NSL-KDD'99 to be more suitable for feature selection and classification techniques, in which dataset records are transferred in a form that is more appropriate to the classification and FS algorithms. Thus the size of this dataset is reduced, and the processing might be faster with fewer hardware processing requirements. Figure 6 shows a snapshot of the dataset before normalization. The steps of this normalization model are discussed in details as follows:

In this section, the preprocessing of the dataset is performed in two main phases, namely: (1) encoding of non-numerical values, (2) normalize numerical values. Figure 7 summarizes these phases.

Figure 7 is used to encode the content of the dataset. For example, each protocol type is given an integer number to covert the original content of dataset to numerical one, TCP = 1, UDP = 2, and ICMP = 3. The same encoding methodology has followed the encoding of the non-numerical content of the dataset.

The final form of the normalized dataset is as the following snapshot, which is shown below in Fig. 8.

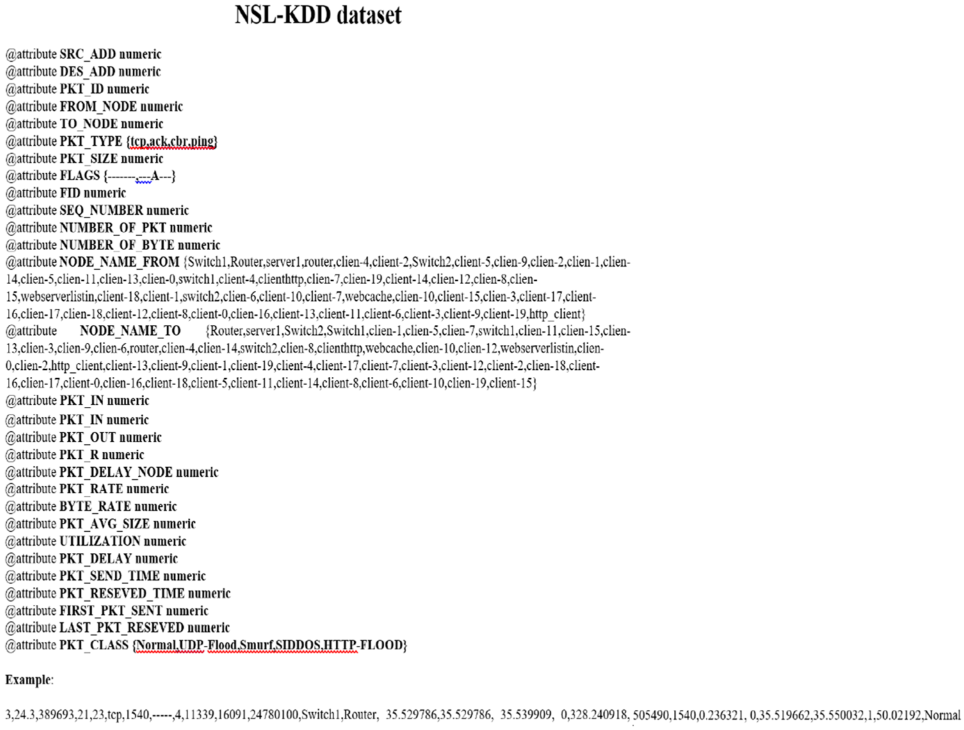
After encoding the non-numerical content of the dataset, the normalization method is used to scale the content of the data set and ensure that the content of the dataset. Therefore, the following Eq. 1 has been used to scale the dataset values to get the normalized value (Jain and Bhandare 2011):

$$X' = (original\,value - \min\_value)/\left(\max_{value} - min\_value\right). \tag{1}$$

3.2 Feature selection using modified grey wolf optimization

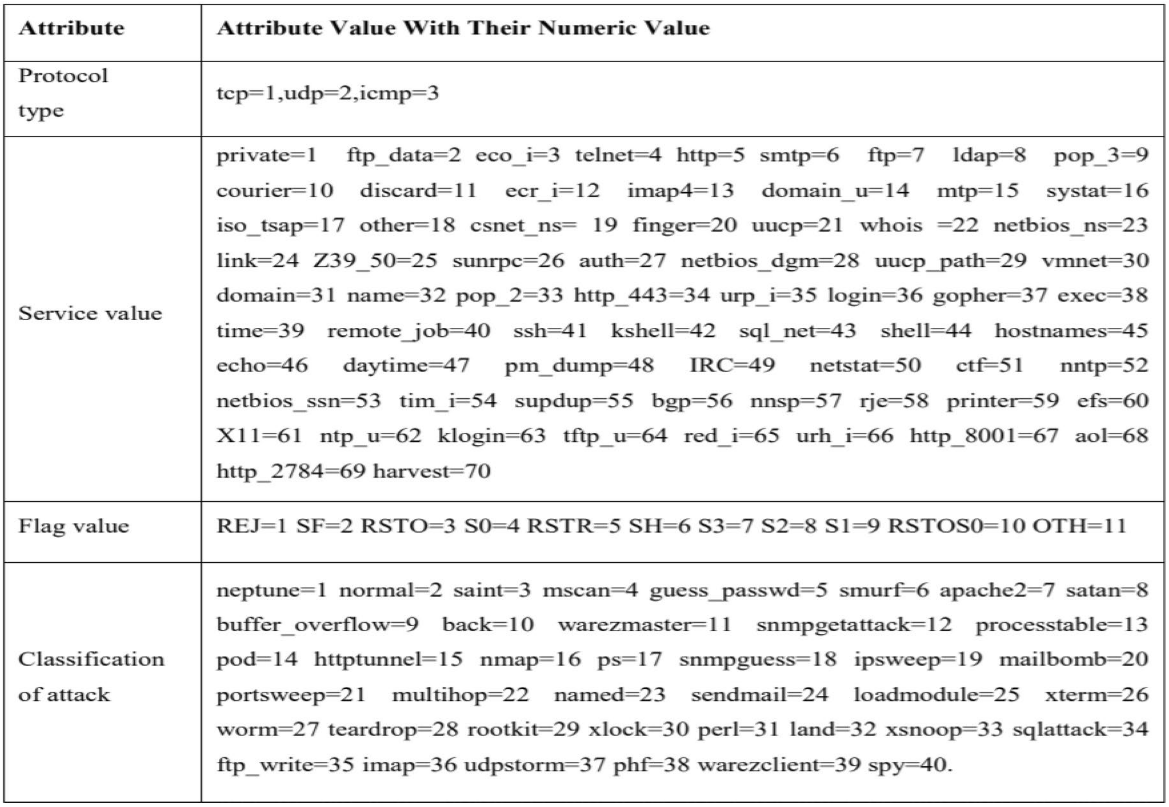
In this paper, the feature selection algorithm (i.e., modified binary grey wolf algorithm) for attribute selection is used to identify the optimal feature set. Thus, it ensures to find out a subset out of full-length features in data set (i.e., 29 features), whose performance is higher than or equal the performance given by the 29 features. Figure 9 presents the flowchart of GWO.

Fig. 6 Snapshot of the original dataset



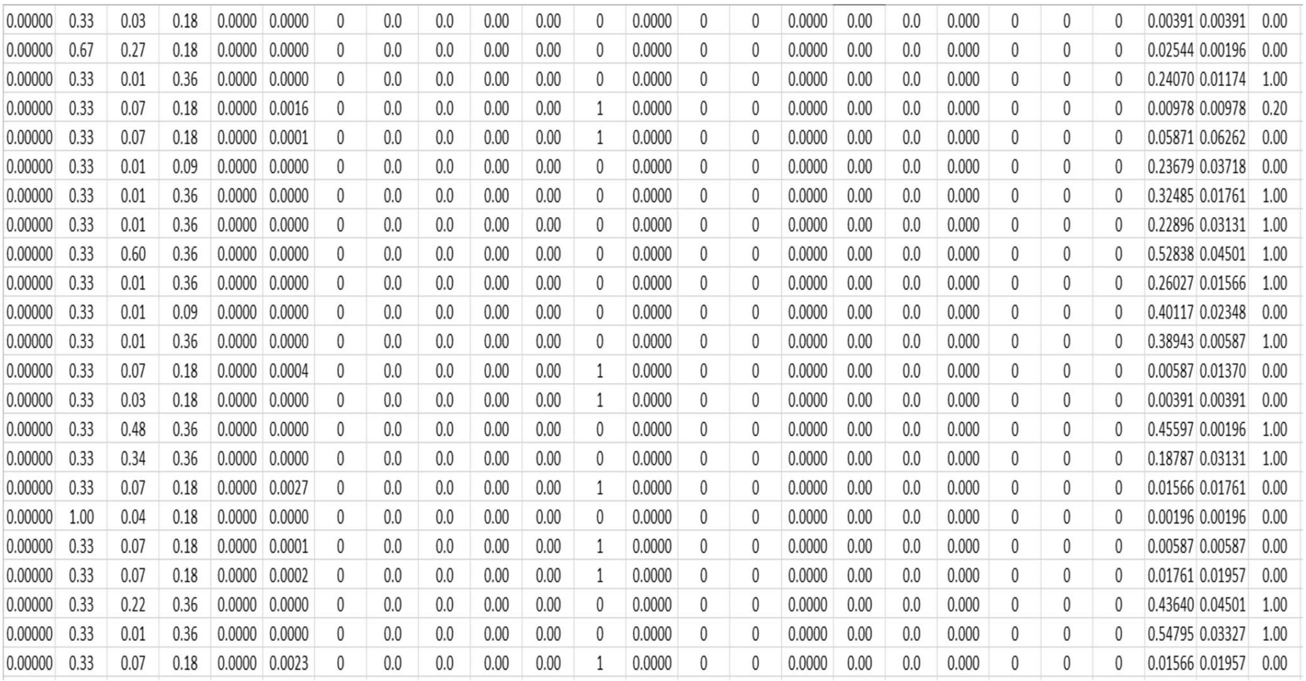
NSL-KDD dataset

```
@attribute SRC_ADD numeric
@attribute DES_ADD numeric
@attribute PKT_ID numeric
@attribute FROM_NODE numeric
@attribute TO_NODE numeric
@attribute PKT_TYPE {tcp,ack,cbr,ping}
@attribute PKT_SIZE numeric
@attribute FLAGS {-------,---A---}
@attribute FID numeric
@attribute SEQ_NUMBER numeric
@attribute NUMBER_OF_PKT numeric
@attribute NUMBER_OF_BYTE numeric
@attribute NODE_NAME_FROM {Switch1,Router,server1,router,clien-4,client-2,Switch2,client-5,clien-9,clien-2,clien-1,clien-
14,clien-5,clien-11,clien-13,clien-0,switch1,client-4,clienthttp,clien-7,clien-19,client-14,clien-12,clien-8,clien-
15,webserverlistin,client-18,client-1,switch2,clien-6,client-10,client-7,webcache,clien-10,client-15,clien-3,client-17,client-
16,clien-17,clien-18,client-12,client-8,client-0,clien-16,client-13,client-11,client-6,client-3,client-9,client-19,http_client}
@attribute      NODE_NAME_TO      {Router,server1,Switch2,Switch1,clien-1,clien-5,clien-7,switch1,clien-11,clien-15,clien-
13,clien-3,clien-9,clien-6,router,clien-4,clien-14,switch2,clien-8,clienthttp,webcache,clien-10,clien-12,webserverlistin,clien-
0,clien-2,http_client,client-13,client-9,client-1,client-19,client-4,client-17,client-7,client-3,client-12,client-2,clien-18,client-
16,clien-17,client-0,clien-16,client-18,client-5,client-11,client-14,client-8,client-6,client-10,clien-19,client-15}
@attribute PKT_IN numeric
@attribute PKT_IN numeric
@attribute PKT_OUT numeric
@attribute PKT_R numeric
@attribute PKT_DELAY_NODE numeric
@attribute PKT_RATE numeric
@attribute BYTE_RATE numeric
@attribute PKT_AVG_SIZE numeric
@attribute UTILIZATION numeric
@attribute PKT_DELAY numeric
@attribute PKT_SEND_TIME numeric
@attribute PKT_RESEVED_TIME numeric
@attribute FIRST_PKT_SENT numeric
@attribute LAST_PKT_RESEVED numeric
@attribute PKT_CLASS {Normal,UDP-Flood,Smurf,SIDDOS,HTTP-FLOOD}

Example:

3,24.3,389693,21,23,tcp,1540,-----,4,11339,16091,24780100,Switch1,Router,  35.529786,35.529786,  35.539909,  0,328.240918, 505490,1540,0.236321, 0,35.519662,35.550032,1,50.02192,Normal
```



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.

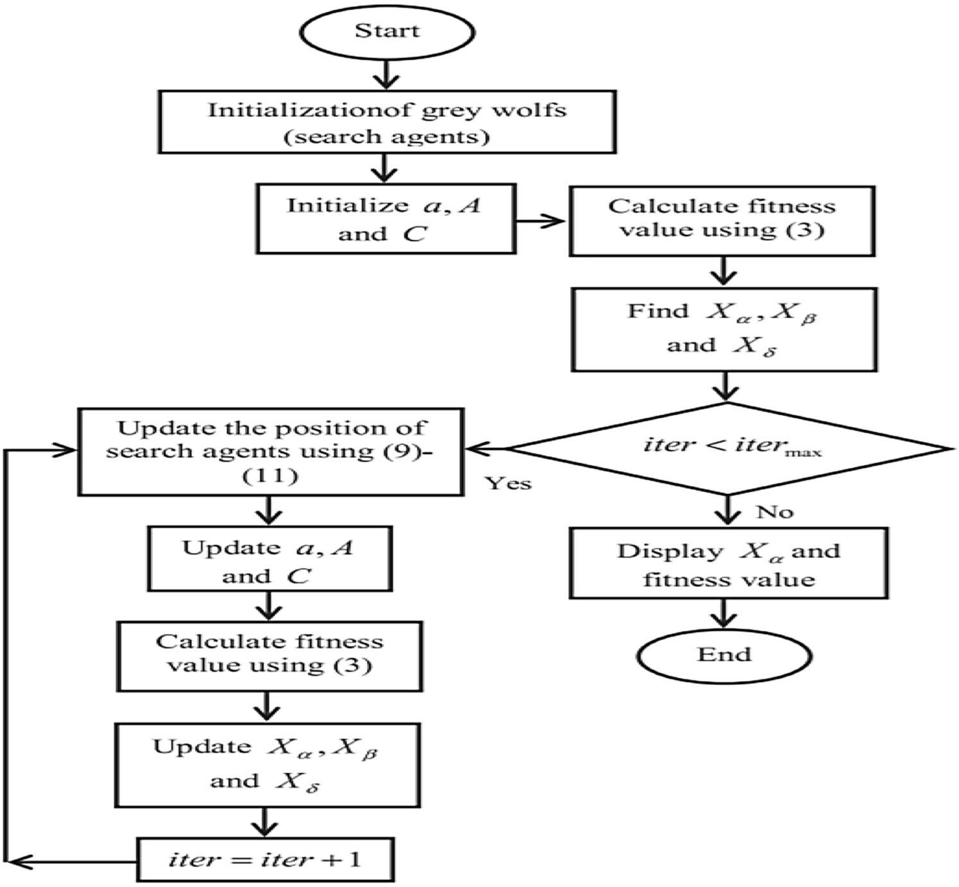
Fig. 7 Encoding dataset



0.00000	0.33	0.03	0.18	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.00391	0.00391	0.00
0.00000	0.67	0.27	0.18	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.02544	0.00196	0.00
0.00000	0.33	0.01	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.24070	0.01174	1.00
0.00000	0.33	0.07	0.18	0.0000	0.0016	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.00978	0.00978	0.20
0.00000	0.33	0.07	0.18	0.0000	0.0001	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.05871	0.06262	0.00
0.00000	0.33	0.01	0.09	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.23679	0.03718	0.00
0.00000	0.33	0.01	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.32485	0.01761	1.00
0.00000	0.33	0.01	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.22896	0.03131	1.00
0.00000	0.33	0.60	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.52838	0.04501	1.00
0.00000	0.33	0.01	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.26027	0.01566	1.00
0.00000	0.33	0.01	0.09	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.40117	0.02348	0.00
0.00000	0.33	0.01	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.38943	0.00587	1.00
0.00000	0.33	0.07	0.18	0.0000	0.0004	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.00587	0.01370	0.00
0.00000	0.33	0.03	0.18	0.0000	0.0000	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.00391	0.00391	0.00
0.00000	0.33	0.48	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.45597	0.00196	1.00
0.00000	0.33	0.34	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.18787	0.03131	1.00
0.00000	0.33	0.07	0.18	0.0000	0.0027	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.01566	0.01761	0.00
0.00000	1.00	0.04	0.18	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.00196	0.00196	0.00
0.00000	0.33	0.07	0.18	0.0000	0.0001	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.00587	0.00587	0.00
0.00000	0.33	0.07	0.18	0.0000	0.0002	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.01761	0.01957	0.00
0.00000	0.33	0.22	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.43640	0.04501	1.00
0.00000	0.33	0.01	0.36	0.0000	0.0000	0	0.0	0.0	0.00	0.00	0	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.54795	0.03327	1.00
0.00000	0.33	0.07	0.18	0.0000	0.0023	0	0.0	0.0	0.00	0.00	1	0.0000	0	0	0.0000	0.00	0.0	0.000	0	0	0	0.01566	0.01957	0.00

Fig. 8 Snapshot of dataset after preprocessing

Fig. 9 Flowchart of GWO

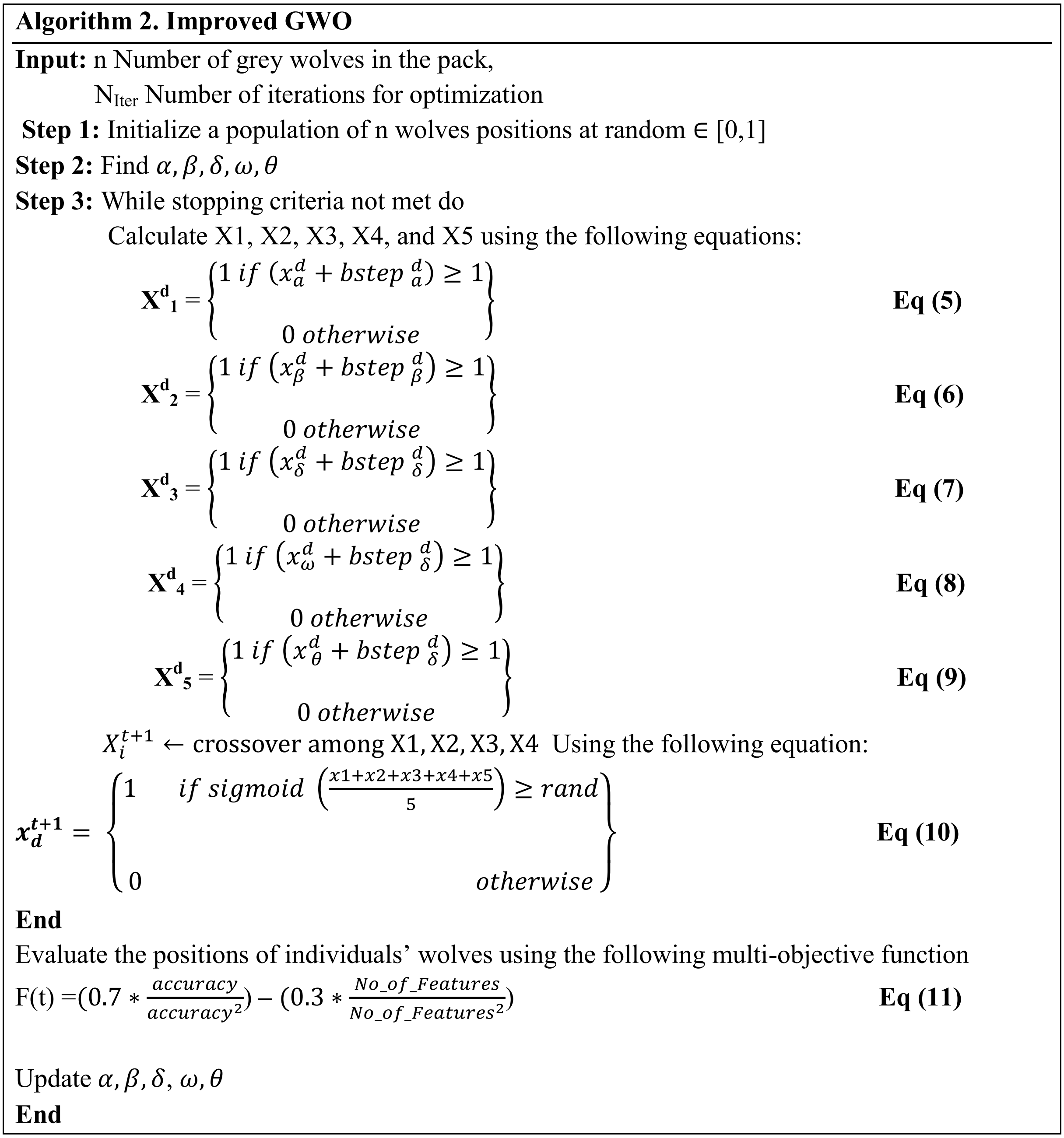

Start
Initializationof grey wolfs (search agents)
Initialize a, A and C
Calculate fitness value using (3)
Find X_α, X_β and X_δ
$iter < iter_{\max}$
Yes
No
Update the position of search agents using (9)-(11)
Update a, A and C
Calculate fitness value using (3)
Update X_α, X_β and X_δ
$iter = iter + 1$
Display X_α and fitness value
End

GWO is one of the most used meta-heuristic feature selection algorithms (see Algorithm 1), which depends mainly on the hierarchy of leadership and the mechanism hunting by grey wolves in the real world. It has four types of wolves, including alpha wolves, beta wolves, delta wolves, and omega wolves (Al-Tashi et al. 2020). These wolves usually prefer residing in a pack, where the average size for each group is ranging from five to twelve. Alpha wolves are located in the 1st place, and beta and delta wolves are considered the second and the third places, respectively. The fourth wolves (omega) are the least expected solution. In other words, the first three wolves guide the optimization, and the fourth one is responsible for tracking the other wolves (i.e., alpha, beta, and delta).

The main element of modified GWO is: (1) increasing the number of wolves to be (five) wolves instead of being (three) wolves, and (2) proposing a new fitness function (multi-objective function) that aims to determine if a subset fits the core goals or not. The importance of accuracy and the total number of selected features are taking into consideration when proposing fitness function in order to evaluate each subset of the selected feature. Thus, the overall performance of the prediction system is based on how it predicts unknown classes efficiently in terms of accuracy and the total number of selected features. The algorithm of enhanced GWO shows in Algorithm 2.

Algorithm 1. Original GWO (Mirjalili et al. 2014).

```
Input: n Number of grey wolves in the pack,
       N_Iter Number of iterations for optimization
Step 1: Initialize a population of n wolves positions at random ∈ [0,1]
Step 2: Find $\alpha, \beta, \delta$
Step 3: While stopping criteria not met do
    Calculate X1, X2, X3 using the following equations:
        $\mathbf{X^d}_1 = \begin{cases} 1\ if\ (x_a^d + bstep\ _a^d) \geq 1 \\ 0\ otherwise \end{cases}$      Eq (2)
        $\mathbf{X^d}_2 = \begin{cases} 1\ if\ \left(x_\beta^d + bstep\ _\beta^d\right) \geq 1 \\ 0\ otherwise \end{cases}$      Eq (3)
        $\mathbf{X^d}_3 = \begin{cases} 1\ if\ \left(x_\delta^d + bstep\ _\delta^d\right) \geq 1 \\ 0\ otherwise \end{cases}$      Eq (4)
    $X_i^{t+1}$ ← crossover among X1, X2, X3
End
Evaluate the positions of individuals' wolves, and Update $\alpha, \beta, \delta$
End
```



```
Algorithm 2. Improved GWO
Input: n Number of grey wolves in the pack,
        $N_{Iter}$ Number of iterations for optimization
Step 1: Initialize a population of n wolves positions at random ∈ [0,1]
Step 2: Find $\alpha, \beta, \delta, \omega, \theta$
Step 3: While stopping criteria not met do
        Calculate X1, X2, X3, X4, and X5 using the following equations:
            $X^{d}_{1} = \begin{Bmatrix} 1\ if\ (x^{d}_{a} + bstep^{d}_{a}) \geq 1 \\ 0\ otherwise \end{Bmatrix}$    Eq (5)
            $X^{d}_{2} = \begin{Bmatrix} 1\ if\ (x^{d}_{\beta} + bstep^{d}_{\beta}) \geq 1 \\ 0\ otherwise \end{Bmatrix}$    Eq (6)
            $X^{d}_{3} = \begin{Bmatrix} 1\ if\ (x^{d}_{\delta} + bstep^{d}_{\delta}) \geq 1 \\ 0\ otherwise \end{Bmatrix}$    Eq (7)
            $X^{d}_{4} = \begin{Bmatrix} 1\ if\ (x^{d}_{\omega} + bstep^{d}_{\delta}) \geq 1 \\ 0\ otherwise \end{Bmatrix}$    Eq (8)
            $X^{d}_{5} = \begin{Bmatrix} 1\ if\ (x^{d}_{\theta} + bstep^{d}_{\delta}) \geq 1 \\ 0\ otherwise \end{Bmatrix}$    Eq (9)
        $X^{t+1}_{i}$ ← crossover among X1, X2, X3, X4  Using the following equation:
    $x^{t+1}_{d} = \begin{Bmatrix} 1 & if\ sigmoid\ \left(\frac{x1+x2+x3+x4+x5}{5}\right) \geq rand \\ 0 & otherwise \end{Bmatrix}$    Eq (10)
End
Evaluate the positions of individuals' wolves using the following multi-objective function
F(t) =$(0.7 * \frac{accuracy}{accuracy^{2}}) - (0.3 * \frac{No\_of\_Features}{No\_of\_Features^{2}})$    Eq (11)

Update $\alpha, \beta, \delta, \omega, \theta$
End
```

3.3 Classification using SVM

After that, a reduced dataset with fewer features is produced to be used as input for the SVM algorithm for classification purposes (i.e., the output of a modified binary grey wolf is used as an input for SVM classifier).

The support vector machine is a machine learning algorithm used for classification purposes. In detail, data is represented in n-dimensional space (n is the number of features), where each feature has a particular coordinate. The classification process is performed by identifying the hyper-plane then classify the classes. Figure 10 summarizes the workflow of the SVM algorithm.

In detail, SVM uses the number of free parameters, which mainly depend on the separation margin that is used to spate different points, as illustrated below in Fig. 11. Thus, using SVM with a reduced number of feature resulted from a feature selection stage avoid over fitting. Besides, SVM introduces a generic technique to adjust the surface of the hyper plane to the selected features by using a kernel function.

However, SVM is adopted in this research because of two main reasons, namely: (1) speed, which allows it to be used in real-time application efficiently, and (2) high scalability,

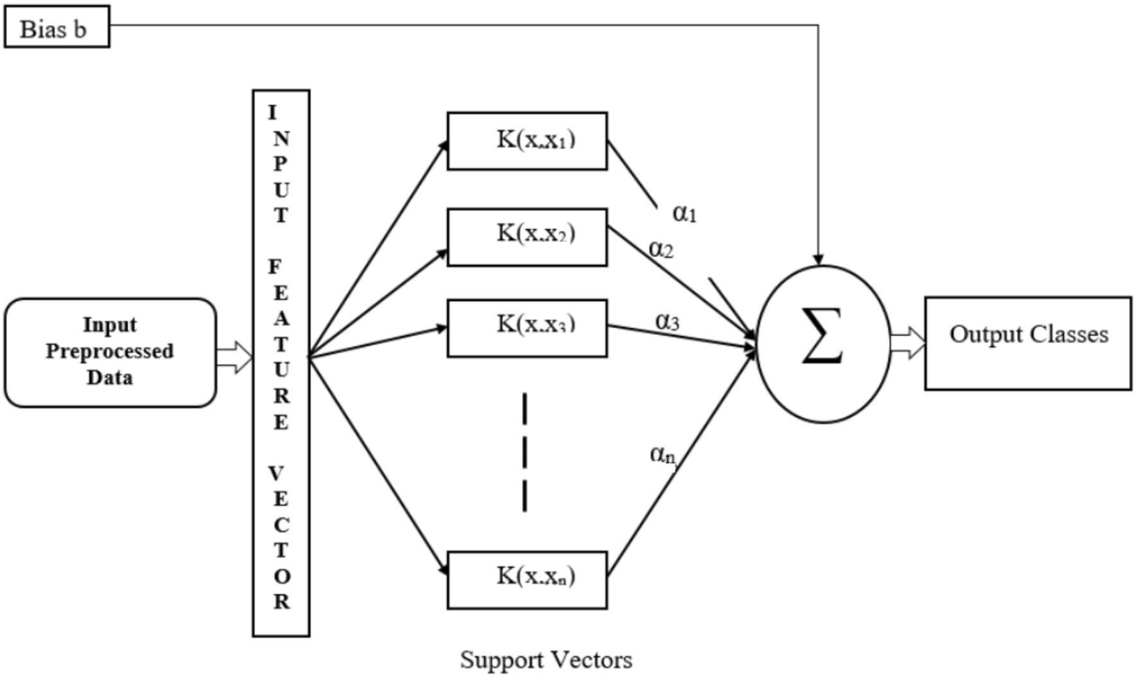


Fig. 10 Flowchart of SVM (Ahmad 2015)

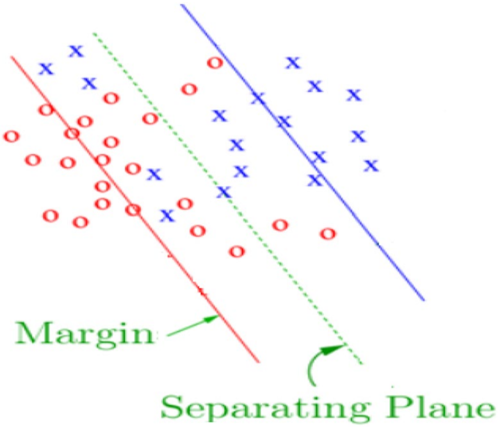


Fig. 11 Separating planes in SVM

while its complexity is not affected by dimensionality of the feature space (Ahmad 2015).

4 Experimental results

In our experiments, Matlab R2019b is used to implement the proposed technique. Matlab R2019b is a widely used analysis and modeling tool used in various systems in different applications. Indeed, it could help in taking solutions' ideas beyond the desktop by executing analyses using a massive volume of data sets. Besides, code written in Matlab might be integrated with other programming languages to provide algorithms deployment with web and enterprises. The hardware specifications used to demonstrate the performance of the proposed solutions is summarized in Table 3.

Table 3 Software and hardware specifications

Software specifications	
Operating system	Win 10 64-bit
Implementation environment	Matlab R2019b
Others	MS-Excel and MS-Word 2016
Hardware specifications	
RAM	12 GB per core
CPU	4 cores, 3.2 GHz for each
HDD	2 TB

4.1 Evaluation criteria

Moreover, five main evaluation metrics will be used to evaluate the proposed technique, namely:

(a) Accuracy, which is the proportion of data correctly classified that is a true positive (TP) and true negative (TN) (Sabri et al. 2011):

$$\text{Accuracy} = \left(\frac{True_{positive} + True_{negative}}{True_{positive} + False_{negative} + False_{positive} + True_{negative}}\right). \quad (12)$$

(b) Number of features, which is average number of selected features used to classify the normal and abnormal behavior efficiently:

$$\text{Number of Features} = (\text{Total}_{\text{number of features}} - \text{Nonselected}_{\text{Features}}). \quad (13)$$

(c) Execution time, which is the time consumed to finish normality and abnormality classification:

$$\text{Execution time} = (\text{Finishing}_{\text{Time}} - \text{Starting}_{\text{Time}}). \quad (14)$$

(d) Detection rate, which is rate of true positive over true positive and false negative:

$$\text{Detection rate} = \left(\frac{\text{True}_{\text{Positive}}}{\text{True}_{\text{Positive}} + \text{False}_{\text{Negative}}}\right). \quad (15)$$

(e) False alarm rate, which is rate of false positive over true negative and false positive:

$$\text{False alarm} = \left(\frac{\text{False}_{\text{Positive}}}{\text{True}_{\text{Negative}} + \text{False}_{\text{Positive}}}\right), \quad (16)$$

where True-positive is the number of intrusions that are detected successfully, False-positive is the number of the typical behaviors that are classified as intrusions, True-negative is the number of normal behaviors that are classified as normal behaviors as well, and False-negative is the number abnormal behaviors that are classified as normal behaviors.

4.2 Testing and analysis

This section provides a detailed evaluation comparison of particle swarm optimization (PSO-IDS, original GWO-IDS, and the GWOSVM-IDS techniques.

4.2.1 Evaluation based on accuracy

This section provides a detailed evaluation comparison of particle swarm optimization (PSO-IDS, original GWO-IDS, and the GWOSVM-IDS techniques.

The proposed methods, GWOSVM-IDS7, GWOSVM-IDS5, and GWOSVM-IDS3, are investigated to show the ability of the proposed method in solving the IDS problem. These three versions of the improved GWO are adjusting to find better wolves configuration that help the basic version in avoiding its drawbacks such as premature conference, diversity of the solutions, and the slow search. Each change in the wolf size will affect the optimizer process, either better or worse. Hence, the number of wolves is examined in this section to find the most suitable number to make a better trade-off between the exploration and exploitation searches.

In terms of Accuracy, the proposed technique outperforms PSO-IDS and GWO-IDS. In details, GWOSVM-IDS with 7 wolves achieves 96% accuracy, GWOSVM-IDS with 5 wolves achieves 92% accuracy, PSO-IDS achieves 89% accuracy, and GWOSVM-IDS with 3 wolves achieves 79% accuracy. Giving the high weight to Accuracy over several features, in the proposed multi-objective function, increases classification accuracy. Therefore, enhancing the accuracy of intrusion detection in the WSN environment (i.e., objective number 1) is achieved. Figure 12 presents the results.

4.2.2 Testing of false alarm rate

In terms of false alarm rate, the proposed technique outperforms PSO-IDSPSO-IDS and GWO based IDS. In detail, GWOSVM-IDS with seven wolves achieves a 3% false alarm rate, GWOSVM-IDS with five wolves achieves 9% false alarm rate, PSO-IDS and GWOSVM-IDS with three wolves achieve 26, 24% false alarm rates respectively. Therefore, decreasing the false alarms rates and the number of

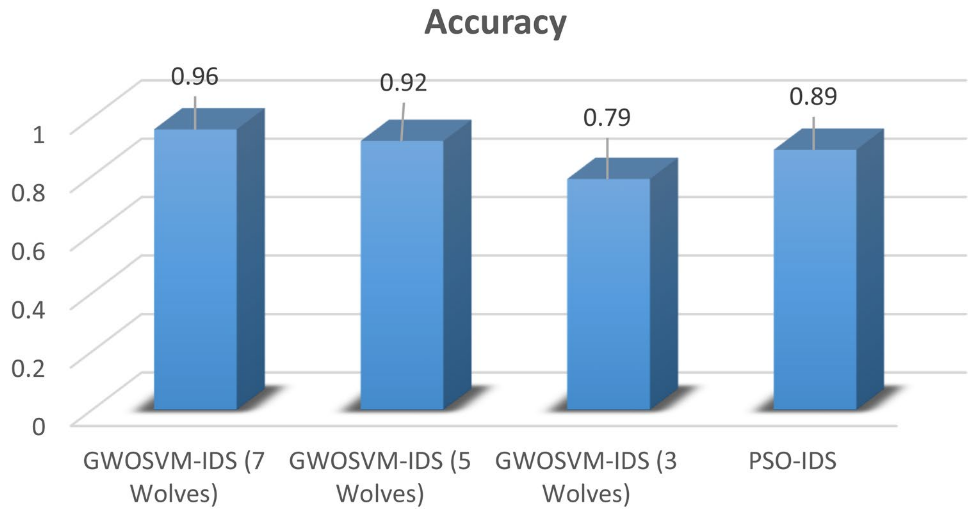


Fig. 12 Evaluation results based on accuracy

features resulted from the IDSs in the WSN environment (i.e., objective number 2) is achieved. Figure 13 summarizes the obtained results in terms of detection rate.

4.2.3 Testing of detection rate

In terms of detection rate, the proposed technique outperforms PSO-IDS and GWO-IDS. In detail, GWOSVM-IDS with seven wolves and GWOSVM-IDS with five wolves achieve 96% detection rate, PSO-IDS, and GWOSVM-IDS with three wolves achieve 93, 83% false detection rate respectively. Therefore, enhancing the detection rate of IDS in the WSN environment is achieved (i.e., objective number 3). Figure 14 summarizes the obtained results in terms of detection rate.

Based on repeatable experiments, using GWOSVM-IDS with 5 wolves and 7 wolves provide consistent and reliable results, since using more wolves would increase complexity and decrease overall detection rate.

4.2.4 Testing of number of features

In terms of number of features, the proposed technique outperforms PSO-IDS and GWO-IDS. In details, GWOSVM-IDS with 7 wolves and GWOSVM-IDS with 5 wolves select 12% features to detect intrusions efficiently, PSO-IDS and GWOSVM-IDS with 3 wolves select 20, 27% features to detect intrusions respectively. Therefore, decreasing the number of features resulted from IDS in WSN environment is achieved (i.e. objective number 4). Figure 15 summarizes the obtained results in terms of number of features.

Based on repeatable experiments, using GWOSVM-IDS with five wolves and seven wolves provide consistent and reliable results since using more wolves would increase complexity and increase the number of selected features.

Fig. 13 Evaluation results based on false alarm rate

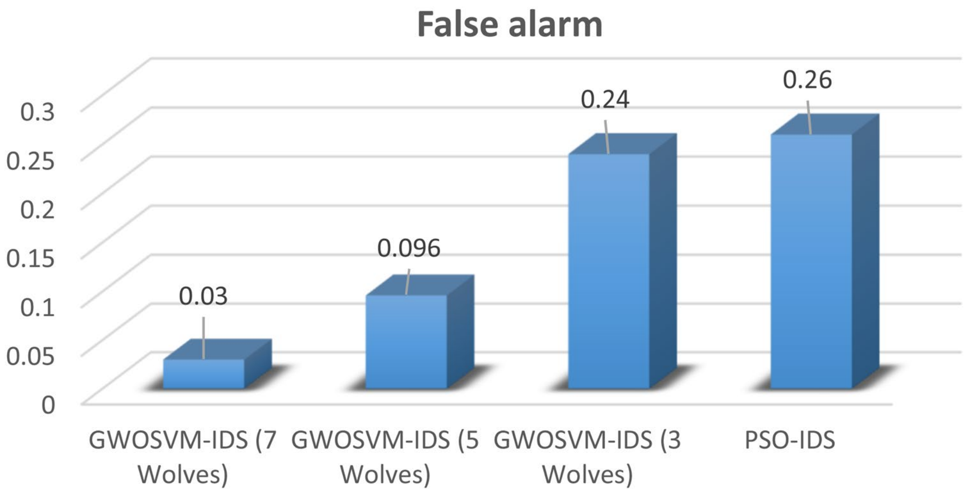
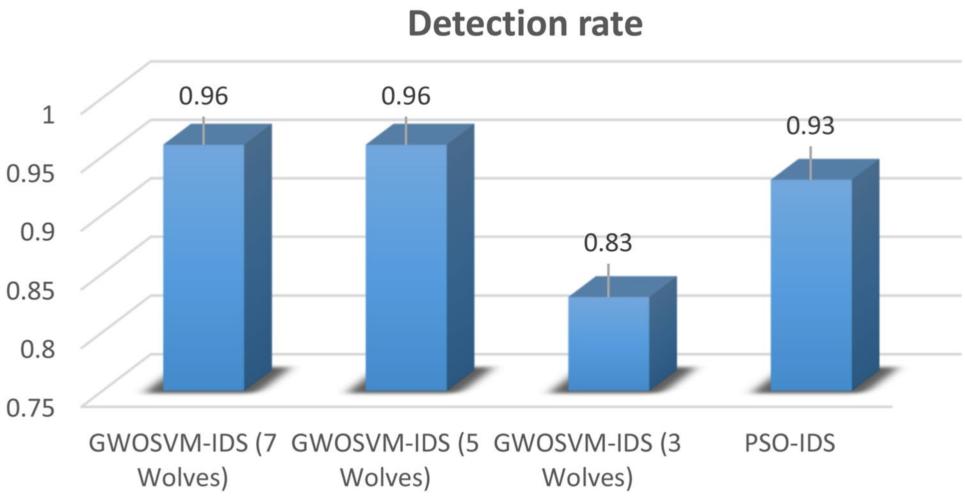


Fig. 14 Evaluation results based on detection rate



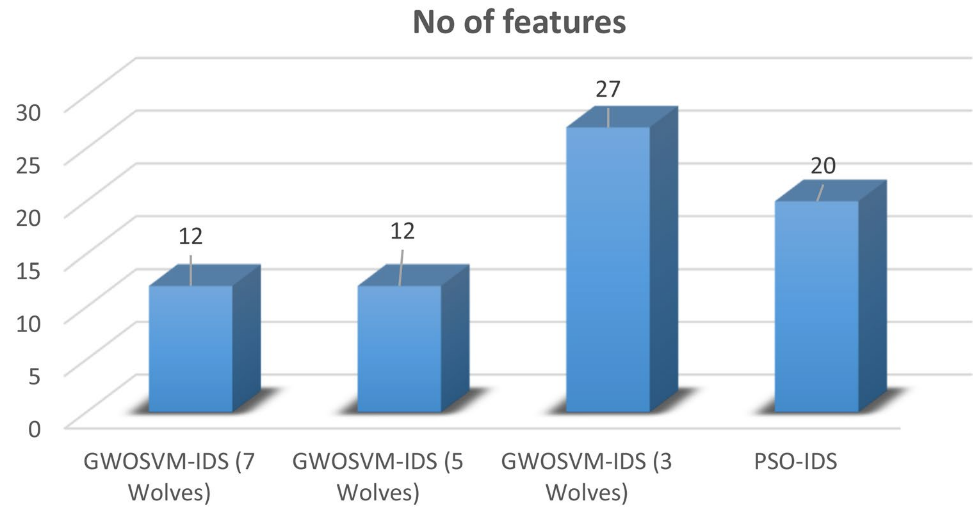


Fig. 15 Evaluation results based on number of features

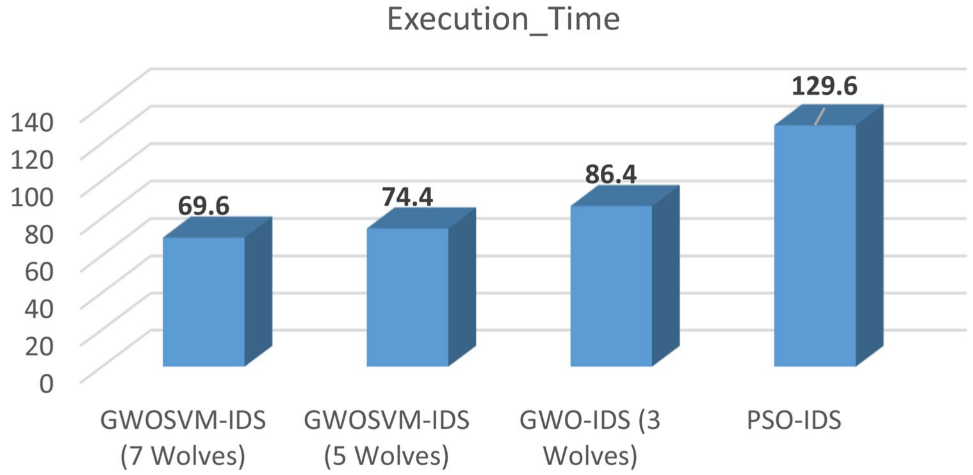


Fig. 16 Evaluation results based on execution time (hours)

4.2.5 Testing of number of execution time

In addition, the proposed method outperforms the other techniques in terms of execution time ash shown below in Fig. 16. As GWOSVM-IDS with seven wolves and GWOSVM-IDS with five wolves consume 69.6 and 74.4 h respectively, but PSO-IDS consumes 129.6 h and GWOSVM-IDS with three wolves consume 86.4 h to detect all intrusions. Therefore, decreasing the processing time required by the IDS in the WSN environment (i.e., objective number 5) is achieved (Table 4).

To determine the enhancement of the proposed technique, the enhancement percentages for GWOSVM-IDS with 7 wolves and with 5 wolves are calculated using the following equation:

$$\text{Enhancment}_{\text{Percentage}} = \left(\left|\text{Old}_{\text{Value}} - \text{New}_{\text{Value}}\right|\right)/\text{Old}_{\text{Vlaue}}. \quad (17)$$

For example, let's say we want to calculate the enhancement percentage of GWOSVM-IDS 7 wolves on GWOSVM-IDS 3 wolves in terms of accuracy, then:

$$\text{Enhancment}_{\text{Percentage\_Accurcay}} = \left(\left|\text{Accurcay}_{3\text{Wolves}} - \text{Accurcay}_{7\text{Wolves}}\right|\right)/\text{Accurcay}_{3\text{Wolves}} = (|\,0.79 - 0.96|)\,/\,0.79 = 0.21.$$

Table 4 Results summary of the three versions of the GWO compared with others

Technique	Accuracy	Detection rate	False alarm	Execution time	Number of features
GWOSVM-IDS 7 wolves	0.96	0.96	0.03	69.6	12
GWOSVM-IDS 5 wolves	0.92	0.96	0.096	74.4	12
GWOSVM-IDS 3 wolves	0.79	0.83	0.24	86.4	27
PSOSVM-IDS	0.89	0.93	0.26	129.6	20

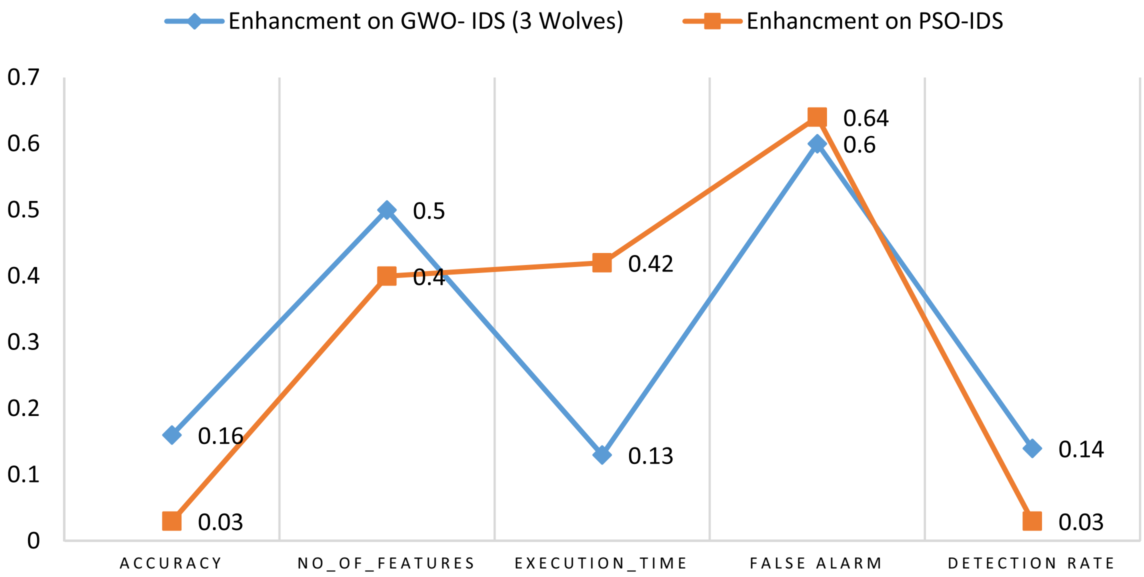


Fig. 17 Enhancement percentage of GWOSM-IDS with 5 wolves on existing IDS techniques

The enhancement percentages for the remaining metrics for each technique are calculated using the same manner. The obtained enhancement results are summarized in Figs. 17 and 18.

Figures 17 and 18 show the enhancement percentage of GWOSVM-IDS with seven wolves and GWOSVM-IDS with five wolves on PSO-IDS and GWOSVM-IDS with three wolves. Indeed, GWOSVM-IDS with five wolves enhances PSO-IDS in terms of accuracy, number of features, and execution time, false alarm, and detection rate 3, 40, 42, 64, and 3% respectively. Further, GWOSVM-IDS with five wolves enhances GWOSVM-IDS with three wolves in terms of accuracy and number of features, execution time, false alarm, and detection rate by 16, 50, 13, 60, and 14% respectively.

Besides, GWOSVM-IDS with seven wolves enhances PSO-IDS in terms of accuracy, number of features, and execution time, false alarm, and detection rate 7, 40, 46, 88, and 15% respectively. Further, GWOSVM-IDS with seven wolves enhances GWOSVM-IDS with three wolves in terms of accuracy and number of features, execution time, false alarm, and detection rate by 21, 50, 24, 87, and 15% respectively. Using the proposed multi-objective function with the proposed IDS techniques plays a significant role in enhancing classification accuracy, detection rate, and decreasing and false alarm rate number of features used to detect attacks efficiently. Also, using more number of wolves might decrease the execution time.

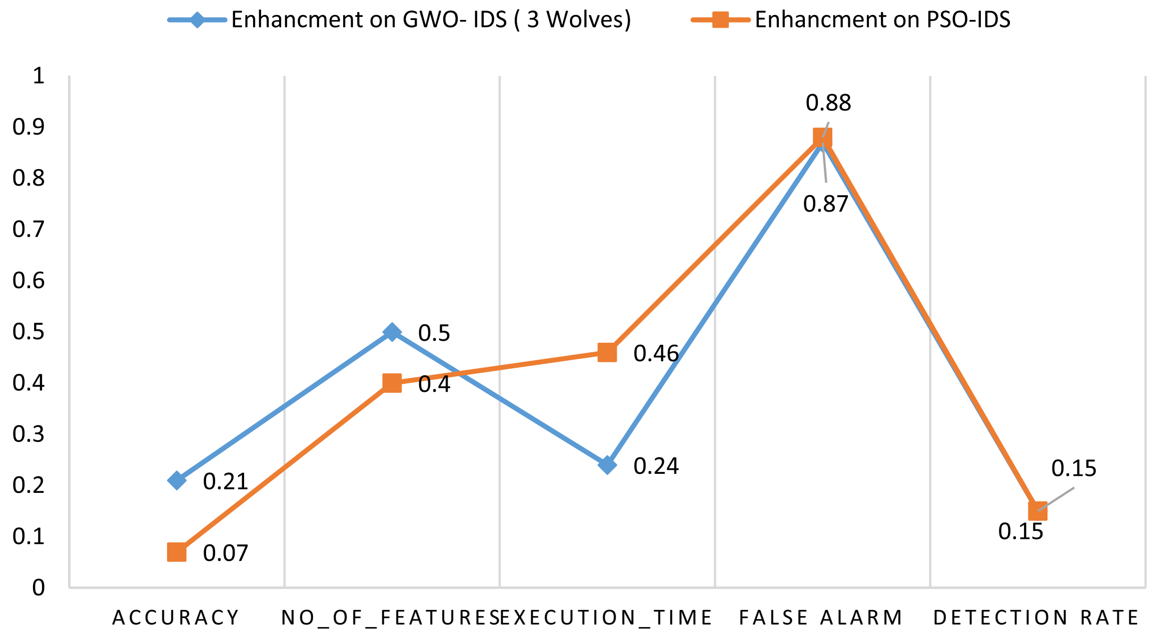


Fig. 18 Enhancement percentage of GWOSM-IDS with 7 wolves on existing IDS techniques

5 Conclusion and future works

GWO is one of the widespread and robust feature selection algorithms used in different fields, such as IDSs. It selects the most important features that can improve the classification accuracy and intrusion detection rate. Despite the effectiveness of the GWO, it has shortcomings related to the accuracy, number of selected features, and execution time. Several studies have been proposed to enhance the performance of the GWO-IDS, but, still, they need more improvements in terms of accuracy, number of selected features, and execution time. This study is conducted to enhance the performance of GWO-IDS in WSN by increasing the number of wolves and using a multi-objective function to increase the overall performance of the prediction system is based on how it predicts unknown classes efficiently in terms of accuracy and the total number of selected features, and speed-up the execution.

NSL KDD’99 has been used to train and test the proposed technique, and the results are compared with the results of original GWO-IDS and PSO-IDS techniques. Indeed, the findings demonstrate that the GWOSVM-IDS technique outperforms the performance of original GWO-IDS and PSO-IDS techniques in terms of accuracy, detection rate, false alarm rate, number of selected features, and execution time.

As future work, In the future, a mathematical prediction technique could be used to predict the next location of wolves, in order to increase the performance of GWO. Also, hybridizing the modified GWO with other efficient and recent feature selection could be conducted to increase the detection rate in the WSN environment. Besides, other classifiers might be used instead of using SVM to enhance the performance of the classification process.

References

Abdollahzadeh S, Navimipour NJ (2016) Deployment strategies in the wireless sensor network: a comprehensive review. Comput Commun 91:1–16

Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer, Berlin

Abualigah L (2020) Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications. Neural Comput Appl. https://doi.org/10.1007/s00521-020-04839-1

Abualigah L, Diabat A (2020) A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. Cluster Comput. https://doi.org/10.1007/s10586-020-03075-5

Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73(11):4773–4795

Abualigah LM, Khader AT, Hanandeh ES (2018) Hybrid clustering analysis using improved krill herd algorithm. Appl Intell 48(11):4047–4071

Ahmad I (2015) Feature selection using particle swarm optimization in intrusion detection. Int J Distrib Sens Netw 11(10):806954

Al-Tashi Q, Rais HM, Abdulkadir SJ, Mirjalili S, Alhussian H (2020) A review of grey wolf optimizer-based feature selection methods for classification. In: Evolutionary machine learning techniques. Springer, Singapore, pp 273–286

Aljawarneh S, Aldwairi M, Yassein MB (2018) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci 25:152–160

Ambusaidi MA, He X, Nanda P, Tan Z (2016) Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans Comput 65(10):2986–2998

Amiri F, Yousefi MR, Lucas C, Shakery A, Yazdani N (2011) Mutual information-based feature selection for intrusion detection systems. J Netw Comput Appl 34(4):1184–1199

Bell DA, Wang H (2000) A formalism for relevance and its application in feature subset selection. Mach Learn 41(2):175–195

Benmessahel I, Xie K, Chellal M (2018) A new evolutionary neural networks based on intrusion detection systems using multiverse optimization. Appl Intell 48(8):2315–2327

Bins J, Draper BA (2001) Feature selection from huge feature sets. In: Proceedings eighth IEEE international conference on computer vision, vol 2. ICCV 2001, IEEE, pp 159–165

Çavuşoğlu Ü (2019) A new hybrid approach for intrusion detection using machine learning methods. Appl Intell 49(7):2735–2761

Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28

Chelli K (2015) Security issues in wireless sensor networks: attacks and countermeasures. In: Proceedings of the world congress on engineering, vol 1, issue 20

Chizi B, Rokach L, Maimon O (2009) A survey of feature selection techniques. In: Encyclopedia of data warehousing and mining, second edition. IGI Global, pp 1888–1895

Curiac DI (2016) Wireless sensor network security enhancement using directional antennas: state of the art and research challenges. Sensors 16(4):488

Devi EM, Suganthe RC (2017) Feature selection in intrusion detection grey wolf optimizer. Asian J Res Soc Sci Human 7(3):671–682

Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381

García-Hernández CF, Ibarguengoytia-Gonzalez PH, García-Hernández J, Pérez-Díaz JA (2007) Wireless sensor networks and applications: a survey. IJCSNS Int J Comput Sci Netw Secur 7(3):264–273

Guo C, Zhou Y, Ping Y, Zhang Z, Liu G, Yang Y (2014) A distance sum-based hybrid method for intrusion detection. Appl Intell 40(1):178–188

Hammoudeh M, Al-Fayez F, Lloyd H, Newman R, Adebisi B, Bounceur A, Abuarqoub A (2017) A wireless sensor network border monitoring system: deployment issues and routing protocols. IEEE Sens J 17(8):2572–2582

Haque S, Rahman M, Aziz S (2015) Sensor anomaly detection in wireless sensor networks for healthcare. Sensors 15(4):8764–8786

Jaiganesh V, Mangayarkarasi S, Sumathi P (2013) Intrusion detection systems: a survey and analysis of classification techniques. Int J Adv Res Comput Commun Eng 2(4):1629–1635

Jain YK, Bhandare SK (2011) Min max normalization based data perturbation method for privacy protection. Int J Comput Commun Technol 2(8):45–50

Jin X, Liang J, Tong W, Lu L, Li Z (2017) Multi-agent trust-based intrusion detection scheme for wireless sensor networks. Comput Electr Eng 59:262–273

Khasawneh AM, Abualigah L, Al Shinwan M (2020) Void aware routing protocols in underwater wireless sensor networks: variants and challenges. J. Phys Conf Ser 1550(3):032145

Khor KC, Ting CY, Phon-Amnuaisuk S (2012) A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection. Appl Intell 36(2):320–329

Mahmood MA, Seah WK, Welch I (2015) Reliability in wireless sensor networks: a survey and challenges ahead. Comput Netw 79:166–187

Maleh Y, Ezzati A (2015) Lightweight intrusion detection scheme for wireless sensor networks. IAENG Int J Comput Sci 42(4):347–354

Maza S, Touahria M (2019) Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms. Appl Intell 49(12):4237–4257

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Nakamura RY, Pereira LA, Costa KA, Rodrigues D, Papa JP, Yang XS (2012) BBA: a binary bat algorithm for feature selection. In: 2012 25th SIBGRAPI conference on graphics, patterns and images. IEEE, pp 291–297

Paulauskas N, Auskalnis J (2017) Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset. In: 2017 open conference of electrical, electronic and information sciences (eStream). IEEE, pp 1–5

Pritchard SW, Hancke GP, Abu-Mahfouz AM (2017) Security in software-defined wireless sensor networks: threats, challenges and potential solutions. In: 2017 IEEE 15th international conference on industrial informatics (INDIN). IEEE, pp 168–173

Rashid B, Rehmani MH (2016) Applications of wireless sensor networks for urban areas: a survey. J Netw Comput Appl 60:192–219

Sabri FNM, Norwawi NM, Seman K (2011) Identifying false alarm rates for intrusion detection system with data mining. IJCSNS Int J Comput Sci Netw Secur 11(4):95

Sedjelmaci H, Feham M (2011) Novel hybrid intrusion detection system for clustered wireless sensor network. arXiv preprint arXiv:1108.2656

Stein G, Chen B, Wu AS, Hua KA (2005) Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd annual southeast regional conference-volume 2. ACM, pp 136–141

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE, pp 1–6

Yu Q, Jibin L, Jiang L (2016) An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks. Int J Distrib Sens Netw 12(1):9653230

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.