Multimedia Tools and Applications (2024) 83:59887–59913
https://doi.org/10.1007/s11042-023-17886-2

Modified aquila optimizer feature selection approach and support vector machine classifier for intrusion detection system

Laith Abualigah[1,2,3,4,5,6,7] · Saba Hussein Ahmed[6] · Mohammad H. Almomani[8] · Raed Abu Zitar[9] · Anas Ratib Alsoud[3] · Belal Abuhaija[10] · Essam Said Hanandeh[11] · Heming Jia[12] · Diaa Salama Abd Elminaam[13,14] · Mohamed Abd Elaziz[15,16,17]

Received: 10 July 2022 / Revised: 9 September 2023 / Accepted: 13 December 2023 /
Published online: 2 January 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
With the ever-expanding ubiquity of the Internet, wireless networks have permeated every facet of modern life, escalating concerns surrounding network security for users. Consequently, the demand for a robust Intrusion Detection System (IDS) has surged. The IDS serves as a critical bastion within the security framework, a significance further magnified in wireless networks where intrusions may stem from the deluge of sensor data. This influx of data, however, inevitably taxes the efficiency and computational speed of IDS. To address these limitations, numerous strategies for enhancing IDS performance have been posited by researchers. This paper introduces a novel feature selection method grounded in Support Vector Machine (SVM) and harnessing the innovative modified Aquila Optimizer (mAO) for Intrusion Detection Systems in Wireless Sensor Networks. To evaluate the efficacy of our approach, we employed the KDD'99 dataset for testing and benchmarking against established methods. Multiple performance metrics, including accuracy, detection rate, false alarm rate, feature count, and execution time, were utilized for assessment. Our comparative analysis reveals the superiority of the proposed method, with standout results in terms of feature reduction, detection accuracy, and false alarm mitigation, yielding significant improvements of 11%, 98.76%, and 0.02%, respectively.

Keywords Aquila optimizer · Intrusion detection system · Support vector machine classifier · Feature selection

1 Introduction

Internet and networks have been included in most daily life activities and applications. For example, during the COVID-19 pandemic, most jobs were transferred using Internet network applications, including critical sectors such as financial education and industrial corporations [1]. These applications require advanced networks that are capable of satisfying such demands. One of the shared network types used for collecting and transferring

Extended author information available on the last page of the article

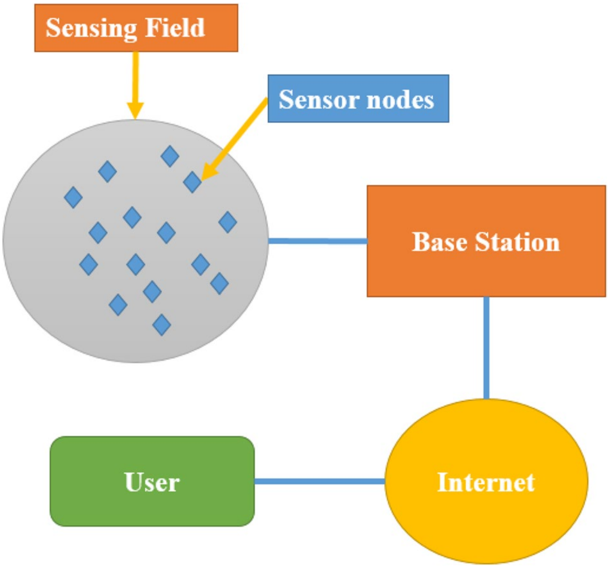


Fig. 1 Wireless Sensors Network components

data is the Wireless Sensors Network (WSN), which consists of several sensors that collect specific data and connect wirelessly [2]. This type of network is mainly used in military applications such as border control and health services to monitor patients' vital signs in the primary nurses' control room [3, 4].

WSN consists of low-power sensors called nodes deployed over a specific area, as given in Fig. 1, to collect environmental data such as sound, vibration, temperature, and motion [5]. This data is sent from the sensors to a central unit or base station for further processing regarding the intended application. The central unit has storage capacity for the data and primarily provides access for the user to the network [6]. Since the sensors are connected wirelessly, effective routing between the base station and nodes can be achieved [7]. Hence, they cover a specific area, for example, a battlefield being watched for enemy movement or a river where the water level is observed to alert early to a flood in real-time. The nodes can be used in several real-time applications to achieve smart detecting, data processing and storage, target tracking, monitoring and controlling synchronization, and node localization [8–10].

The sensors used in WSNs are usually small in size, low in power consumption, and powered mainly by a battery or small solar cell. They have transmitters and receivers to communicate with each other and the base station [11]. In the process of developing WSNs, many properties must be regarded. For example, power-wise nodes must be efficient in their consumption due to the difficulty of supplying the nodes with power because of their remote location, which is usually deployed in vast open areas, and this requires the choice of the appropriate power source. Also, a WSN must be resilient and able to cope with any network fault and maintain a stable level of exemplary service [12]. WSN can be homogeneous, consisting of one type of sensor that collects one type of data; on the other hand, heterogeneous nodes with different types or sizes collect various data [13].

Since data is transmitted wirelessly between the nodes, it is possible to hack the network once in range of the network. Due to the sensitivity of the data in most applications, for example, real-time military locations of troops or critical financial data where the security of a whole nation can be threatened, this leads to the importance of an effective security system [14]. Security threats to WSNs come mainly from the network's intrusion by hackers, where data can be stolen or manipulated [15–18]. The Intrusion Detection System (IDS) is needed to detect such activities, which can be defined as a system that can extract and examine the features of the input data and interrupt the detected anomalous data, significantly increasing the system's security [19–21].

IDS is the primary line of defense for the network against an attack that can affect the system's performance or steal critical information. The defense mechanism of IDS is represented as it monitors the network activities and events and analyzes the traffic for any unusual activity that can be a potential threat. Moreover, IDS performs other functions; for example, it analyzes the user activity to ensure that users do not break the user Abuse policy. IDS performs these tasks by analyzing the collected data, extracting certain features, and comparing them to save data from identified attacks or threats [22]. Several types of attacks might occur according to the nature of the WSN application. For example, bank account information can be stolen, or power plant data can be manipulated to cause a blackout. Hence, it is significant to use artificial intelligence and machine learning techniques to give IDS the ability to adapt to different attacks.

IDS processes a significant quantity of data that may contain irrelevant and useless features that cause a lower detection rate, accuracy, and computation time. Hence, it is necessary to use a technique to choose the most appropriate features to enhance the performance of IDS in terms of detection rate, accuracy, and computation time [23, 24]. Several IDS models were developed to adjust to different attacks. Still, most suffer from weak detection rates, high false alarm rates, and high time consumption, allowing malicious data and cyber-attacks through the network. In addition, high false alarm rates cause misdiagnosing of standard data as suspicious, resulting in weak security and more network vulnerability. Therefore, it is essential to develop more efficient and adaptive IDS to ensure the security of the WSN, and this can be done by involving optimization algorithms with IDS to increase detection accuracy. Over the years, researchers have developed many optimization methods for feature selection in IDS [25–28], which have been proven to enhance IDS performance significantly.

SVM-based IDS systems can handle a limited number of users. With the increasing use of the WSNs, the efficiency of such systems is less, and threats increase, which motivated the researchers to develop new methods to detect any threats. This paper suggests a new method using a modified Aquila Optimizer (mAO) to select the best features with the least possible number compared to the original Aquila Optimizer set before the classification process. The feature selection step intends to reduce the amount of data by removing irrelevant or similar features. Only the relevant features are used in the classification to achieve better IDS performance regarding accuracy, detection rate, and false alarm. Modifying the original Aquila Optimizer replaced the levy distribution function with the Cauchy distribution function in narrowed exploration and exploitation steps. The main aim of the research is to study the effect of feature selection using mAO on IDS in WSNs based on the SVM algorithm. Where mAO is used to select features by SVM to classify the data set. The evaluation of the proposed method is conducted using the Dataset of KDD, and the results are compared with other methods in the literature. The proposed IDS' main objectives include the preparation of the Dataset, feature selection using mAO, and data classification using SVM to detect intrusion. The achieved contributions can be indicated as follows.

- A new method is proposed to enhance the intrusion detection rate in the WSN environment called mAO.

- The proposed method used an improved Aquila Optimizer (MAO) version to solve the feature selection problem.
- The proposed method affects the false alarm rate and the processing time of intrusion detection systems in the WSN environment.

The novelty of this research lies in several key aspects:

- Innovative Feature Selection Method: The research introduces a novel feature selection method based on Support Vector Machine (SVM) in combination with the modified Aquila Optimizer (mAO). This approach is not commonly found in existing literature and offers a unique way to enhance Intrusion Detection Systems (IDS) in Wireless Sensor Networks (WSNs).
- Efficiency Improvement: The proposed method specifically targets the challenge of efficiency in IDS performance when dealing with the substantial volume of data generated by sensors in wireless networks. By effectively selecting relevant features, it aims to optimize computational efficiency and reduce the computational burden on the IDS.
- Comprehensive Evaluation: The research conducts a comprehensive evaluation using various performance metrics, including accuracy, detection rate, false alarm rate, feature count, and execution time. This thorough assessment provides a well-rounded understanding of the proposed method's capabilities and advantages compared to existing approaches.
- Significant Performance Gains: The results of the comparative analysis indicate notable performance gains in terms of feature reduction, detection accuracy, and false alarm reduction. These improvements are noteworthy, with an 11% reduction in features, a 98.76% detection rate, and a minimal false alarm rate of 0.02%.

In summary, the novelty of this research lies in its unique feature selection methodology, its focus on efficiency improvements in IDS for WSNs, and its comprehensive evaluation that demonstrates significant performance enhancements compared to existing methods.

The rest of this paper is organized as follows. Section 2 presents the most related works to the IDS. Section 3 shows the procedure of the proposed method. Section 4 shows the experiments and results. Section 5 presents the conclusion and future work directions.

2 Related works

The researchers have been continuously developing IDS models benefiting from long-term experience with different attack types and the becoming more prominent and larger scale of the Internet to enhance the efficiency of real-time intrusion detection [29]. Recently, machine learning methods have been employed for IDS in WSNs. In this section, some recent optimization methods for feature selection are represented as follows:

An intrusion detection system (IDS) is defined as one that can extract and examine the features of the input data and interrupt the detected anomalous data, significantly increasing the system's security [19]. IDS is the primary line of defense for the network against an attack that can affect the system's performance or steal critical information. The defense mechanism of IDS is to monitor the network activities and events and analyze the traffic for any unusual activity that can be a potential threat.

Wireless Sensor Network (WSN) consists of low-power sensors called nodes deployed over a specific area to collect specific data and connected wirelessly to each other. This data is sent from the sensors to a central unit or (base station) for further processing regarding the intended application. The central unit has storage capacity for the data and mainly provides access for the user to the network [2]. Since the sensors are connected wirelessly, effective routing between the base station and nodes can cover a specific area.

Support Vector Machine is a supervised learning algorithm for classification and regression problems [30]. The SVM algorithm's primary function is to find the optimal boundary, called a hyperplane, to classify the input data. Feature Selection methods remove irrelevant or similar features, and only the significant features are used in the classification [31–35]. It is essential to have a method for choosing the best features to increase accuracy and reduce training and testing time [36]. Feature selection aims to resolve some of the issues in IDS by choosing relevant features that hold basic information to aid the classification process [23]. This results in reduced cost, less storage space, and a comprehensive understanding of the data.

Machine learning (ML) techniques, such as Support Vector Machine (SVM), Logistic regression, and decision tree, use features of input data to categorize the network data into standard or suspicious data, called classification. ML is fast when analyzing a limited amount of data and a few features. However, when analyzing large amounts of data based on many features, the system becomes less efficient, and the probability of overfitting increases. Any irrelevant or similar features are removed to prevent such problems, and only the significant features are used in the classification; this process is called feature selection [37].

Over the years, many optimization algorithms have been developed for feature selection [38, 39]. Meta-heuristic algorithms are famously known for their ability to adapt to different systems, such as the Aquila Optimizer (AO) [40], Dwarf Mongoose Optimization Algorithm (MDOA) [41], Arithmetic Optimization Algorithm (AOA) [42], Ebola Optimization Search Algorithm (EOSA) [43], Whale Optimization Algorithm (WOA) [44], Starling Murmuration Optimizer (SMO) [45], Grey Wolf Optimizer (GWO) [46], and Reptile Search Algorithm (RSA) [47]. The primary function of such algorithms is to find the optimal solution using exploration and exploitation techniques to search all possible areas and obtain the best solution [48, 49].

A modified grey wolves optimizer is proposed in [25] to solve IDS, called mGWO. GWO is a nature-inspired optimization algorithm that simulates headship arrangement and hunting techniques by grey wolves in nature. The order consists of alpha, beta, delta, and omega wolves. Usually, the wolves live in packs, where the usual number for each pack ranges between five and twelve. The order of leadership consists of alpha wolves, then beta and delta wolves, respectively. Another type of wolf is omega, the least expected solution, responsible for watching other wolves. By modifying the number of wolves from three to five and suggesting a new cost function, GWO is called modified GWO (mGWO), which is the contribution of the research.

In [50], an Arithmetic operators optimization algorithm (AOA) with SVM Intrusion detection system (AS_IDS) was suggested to improve AOA-based IDS' efficiency in WSN. The outcomes showed that the suggested method is better than the GWO-based IDS method regarding accuracy, detection rate, and execution time. The AS_IDS method enhanced the evaluation metrics by 0.67% accuracy, 2.80% detection accuracy, and % execution time of 97%. On the other hand, the based IDS method was better than AS_IDS in terms of false alarms with a rate of less than 33% and the number of features by 25%.

In [51], an intrusion detection model was built to be well-matched with the properties of WSN. The model uses the information gain ratio and the online passive-aggressive classifier. Initially, the features are selected using the information gain ratio. Then, different Deny of Service attacks were used to train the online passive-aggressive algorithm for detection. The suggested model ID-GOPA has achieved a detection rate of 96%. The detection accuracy was 86%, 68%, 63%, and 46% for detecting different types of attacks, namely gray hole, coding, and blackhole attacks. These results indicate that the ID-GOPA model provided good intrusion detection to the WSN.

In [52], a technique called cross-correlation-based feature selection (CCFS) is developed and utilized with the use of four types of classification techniques, and the results of the new method were compared with the Cuttlefish Algorithm (CFA) and Mutual Information-Based Feature Selection (MIFS). The used datasets are KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017. Analyzing the results showed that the CCFS technique is the best in terms of results in accuracy, precision, recall, and $F1$-score criteria among the compared algorithms using DT as a classifier.

In [53], a wrapper feature selection algorithm for IDS using a pigeon-inspired Optimizer is developed. The proposed approach converts the continuous pigeon Optimizer to binary and compares it to the commonly used method for converting continuous swarm intelligent algorithms into binary. Three different datasets, *KDDCUP* 99, NLS-KDD, and UNSW-NB15, were used to verify the suggested method. The proposed algorithm was superior to many feature selection algorithms from the literature regarding True Positive Rate (TPR), False Positive Rate(FPR), accuracy, and F-score. In addition, the suggested cosine similarity method for binarizing the algorithm has a better convergence rate than the sigmoid method.

In [54], a method is developed for data collection and processing using DST-based fuzzy membership for indoor WLAN intrusion detection and generates a characteristic database at each marked point. The researcher used a linear independent function for the fuzzy membership estimation treatment to model the membership function. On the other hand, in DST calculation, the membership function is used as the probability mass function to assess the reference points. Finally, the calculated maximum probability and centroid modes are used to estimate the location of the suspicious event. The experimental results showed that the proposed technique has more efficient intrusion detection than the current PNN and ray tracing.

In [55], a feature selection model based on a hybrid learning mechanism was created. The proposed IDS joins feature selection and clustering. The first uses a Support Vector Machine (SVM), and the second uses a K-Medoids clustering algorithm. Also, the method utilizes the Nave Bayes classifier to assess the KDD CUP99 dataset. The suggested method is assessed by three performance measures: accuracy, detection rate, and false alarm rate. The results were compared with three other feature selection methods. The comparison methods included K-Medoids GFR Naive Bayes, K-Medoids Nave Bayes, and tenfold cross-validation Nave Bayes. The results showed that the proposed method has an accuracy (91.5%), detection rate (90.1%), and false alarm rate (6.36%).

Deep learning-based Intrusion Detection Systems (IDS) offers a promising avenue for effectively identifying intrusions with exceptional precision. Nonetheless, owing to their intricacy, these models often present a challenge, as they are regarded as enigmatic 'black boxes' by developers and security analysts due to their incomprehensible decision-making processes. Encouraged by these complexities, this paper introduces an explainable and robust IDS tailored for Industry 5.0 [56]. The proposed IDS is fashioned by integrating bidirectional long short-term memory networks (BiLSTM), a bidirectional-gated recurrent unit (Bi-GRU), fully connected layers, and a softmax classifier, all aimed at enhancing the intrusion detection capabilities within the context of Industry 5.0. Subsequently, we employ the Shapley Additive exPlanations (SHAP) mechanism to shed light on and gain insights into the most influential features driving the decisions of this cyber-resilient IDS. Assessing the proposed model's performance using explainability techniques is a critical assurance of its functionality. Empirical findings, based on the CICIDDoS2019 dataset, affirm the superior performance of the proposed IDS when compared to several recent approaches.

As an alternative, many researchers explore vehicle intrusion detection systems (IDSs) using side-channel analysis, which doesn't impact CAN bus bandwidth. However, existing solutions often fall short of pinpointing the source electronic control unit (ECU) of malicious data frames or detecting such frames from both ECUs and external nodes simultaneously, limiting their practicality. To address these limitations, they introduce an innovative IDS relying on vehicle voltage signals [57]. They map multiple identifiers (IDs) each ECU sends, even without developer documentation. Additionally, they pioneer a Feature-Bagging-CNN hybrid model to detect malicious intrusions precisely. This system excels at detecting and tracing the origin of malicious data frames sent by external nodes or compromised ECUs, enhancing its practical utility.

The escalating diversity, decentralization, and sheer volume of consumer electronic (CE) devices have caused a significant surge in data traffic. Conventional static network infrastructure approaches require manual setup and exclusive management of these CE devices. In response to these challenges, this article introduces an innovative approach that combines Software-Defined Networking (SDN) with Deep Learning (DL) to create an intelligent Intrusion Detection System (IDS) for smart CE networks [58]. The approach first leverages SDN architecture as a dynamic solution capable of reconfiguration within static network infrastructure. It effectively addresses the distributed nature of smart CE networks by decoupling control and data planes. Next, a DL-based IDS employing Cuda-enabled Bidirectional Long Short-Term Memory (Cu-BLSTM) is developed to detect various types of attacks within the smart CE network. Simulation results using the CICIDS-2018 dataset demonstrate the effectiveness of this approach, surpassing recent state-of-the-art security solutions. It solidifies its position as a promising choice for securing next-generation smart CE networks.

In [59], a feature selection technique is proposed based on the binary grey wolf optimization. The main goal of the technique is to find the optimal position of the related features during the classification. The method utilizes stochastic crossover and a sigmoidal function to obtain the updated grey wolf position. This leads to increasing the classification accuracy and reducing the selected features. A dataset from the UCI (UC Irvine) repository was used, and the results were compared to the genetic algorithms and particle swarm Optimizer. The results show that the suggested approach is superior to genetic algorithms and particle swarm Optimizers in solution search, feature selection fitness, and accuracy. An overview of the given studies is presented in Table 1.

As stated, most of the proposed methods in the literature have achieved moderate accuracy, detection rate, and false alarm results. It was known that the execution time was rarely accounted for in most of the proposed methods, which leads to weak security and a more vulnerable WSN. Hence, a method for optimizing the selected features along with other metrics is still needed to enhance the accuracy of intrusion detection. As a widely used technology, WSN has become an exciting field for researchers and scientists to develop more secure and efficient networks for sensitive applications. Because of the continuously changing attack methods, researchers have been motivated to develop various models of

Table 1 An overview of the literature review

Reference & Year	Title	Summary	Limitations
[?] 2021	The Arithmetic Optimization Algorithm for Intrusion Detection System in Wireless Sensor Networks	The outcomes showed that the suggested method was better than the GWO-based IDS method regarding the accuracy, Detection rate, and execution time	High number of features
[52] 2021	Anomaly Detection using Machine Learning Techniques in Wireless Sensor Networks	The suggested model ID-GOPA has achieved good results of detection rate of 96%. And good detection accuracy for detecting different types of attacks These results indicated that this provides good intrusion detection to the WSN	Execution time is not considered
[60] 2020	Cross-correlation-based feature selection (CCFS)	- The used datasets were KDD Cup 99, NSL-KDD, AWID, and CIC-IDS2017 - Analyzing the results showed that the CCFS technique was the best in terms of results in accuracy, precision, recall, and $F1$-score criteria among the compared algorithms using DT as a classifier	Number of features and execution time are not considered
[53] 2020	Wrapper feature selection algorithm for IDS using pigeon inspired optimizer	- datasets, *KDDCUP* 99, NLS-KDD and UNSW-NB15 were used -The proposed algorithm was superior to many algorithms in terms of True positive rate, False Positive Rate, accuracy, and F-score. In addition, the suggested cosine similarity method for binarizing the algorithm has a better convergence rate than the sigmoid method	Number of features and execution time are not considered
[50] 2020	Data collection and processing using DST based fuzzy membership for indoor WLAN intrusion detection	- linear independent function was used to model the membership function for the fuzzy membership estimation treating - the membership function was used as the probability mass function to assess the reference points - The experimental results showed that the proposed technique has more efficient intrusion detection than the current PNN and ray tracing	False alarm, number of features and execution time are not considered

Table 1 (continued)

Reference & Year	Title	Summary	Limitations
[55] 2018	Feature selection model based on a hybrid learning mechanism	- KDD CUP99 dataset - The results showed that the proposed method has an accuracy (91.5%), detection rate (90.1%), and false alarm rate (6.36%)	Low accuracy and high false alarm
[59] 2016	Feature selection technique based on the binary grey wolf optimization	- Dataset from the UCI (UC Irvine) repository was used The results show that the suggested approach was superior to genetic algorithms and particle swarm optimizer in solution search, feature selection fitness, and accuracy	False alarm and execution time are not considered

IDS and used optimization algorithms to enhance their performance, mainly in terms of time and accuracy, as traditional IDS suffered from weak performance and long execution time, causing security issues for WSNs. The AO algorithm has achieved superior results as an optimization method. In this paper, we suggested a modified AO algorithm (mAO) for the feature selection process with the ambition of achieving superior results compared to the literature in terms of the evaluation metrics mentioned above, especially in terms of accuracy and execution time which means more efficient IDS in WSN.

3 The proposed method

In this chapter, a new method using the mAO algorithm is represented to enhance the performance of IDS by performing a feature selection process to select the best set of features to be classified by SVM for intrusion detection.

3.1 The original aquila optimizer (AO)

AO is a new optimization algorithm inspired by Aquila's hunting behavior in nature [59]. AO is represented in four methods of search that mimic Aquila hunting actions: expanded exploration, narrowed exploration, expanded exploitation, and narrowed exploitation [61]. The method uses the condition if t⩽(2/3)*T to transfer from the exploration to the exploitation method and levy flight as a distribution function [62].

3.1.1 Expanded exploration

This method is the first step of the optimization process and is represented mathematically as in Eq. (1).

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + \left(X_M(t) - X_{best}(t) * rand\right) \tag{1}$$

where, $X_1(t+1)$ is the solution of the next run of t, found by (X_1). $X_{best}(t)$ is the best-found solution up to t^{th} iteration. The formula $\left(1 - \frac{t}{T}\right)$ is utilized to control the wide search by the number of iterations. $X_M(t)$ is the locations mean value of the current solutions at t^{th} iteration and calculated using Eq. (2). *rand* is a random value between 0 and 1. t and T is the current iteration and the maximum iteration number, respectively [63].

$$X_M(t) = \frac{1}{N} \sum\nolimits_{i=1}^{N} X_i(t), \forall j = 1, 2, \ldots.., Dim \tag{2}$$

where *Dim* is the dimension size of the problem and *N* is the number of population size.

3.1.2 Narrowed exploration

This is the second step of the optimization process with the following mathematical representation.

$$X_2(t+1) = X_{best}(t) \times Levy(D) + X_R(t) + (y - x) * rand \tag{3}$$

where, $X_2(t+1)$ is the solution to the next iteration. D is the dimension space. Levy (D) is the levy flight distribution function, which is calculated by Eq. (4). $X_R(t)$ is a random solution taken in the range of $[1, N]$ at the i^{th} iteration.

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{P}}} \tag{4}$$

where, s is a constant value fixed to 0.01, u, and υ are random numbers between 0 and 1. σ is calculated using Eq. (5).

$$\sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right) \tag{5}$$

where β is a constant value fixed to 1.5 and gamma function. $\Gamma(x)$ is calculated as given in Eq. (6).

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt \tag{6}$$

3.1.3 Expanded exploitation

In this step, the AO starts to converge the search space for the solution, where the mathematical representation is the presented in Eq. (7).

$$X_3(t+1) = \left(X_{best}(t) - X_M(t)\right) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta \tag{7}$$

where $X_3(t+1)$ the solution of the next run of t, and $X_{best}(t)$ is the estimated location of the prey until i^{th} iteration. $X_M(t)$ is the average value of the current solution at t^{th} iteration. *rand* is a random number that varies from 0 to 1. α and δ are the exploitation tuning parameters equal to (0.1). *LB* refers to the lower bound, and *UB* refers to the upper bound of suggested solution.

3.1.4 Narrowed exploitation

In the last step of the optimization process, the AO narrows the search space to find the final optimal solution, which is represented mathematically in the Eq. (8).

$$X_4(t+1) = QF \times X_{best}(t) - \left(G_1 \times X(t) \times rand\right) - G_2 \times Levy(D) + rand \times G_1 \tag{8}$$

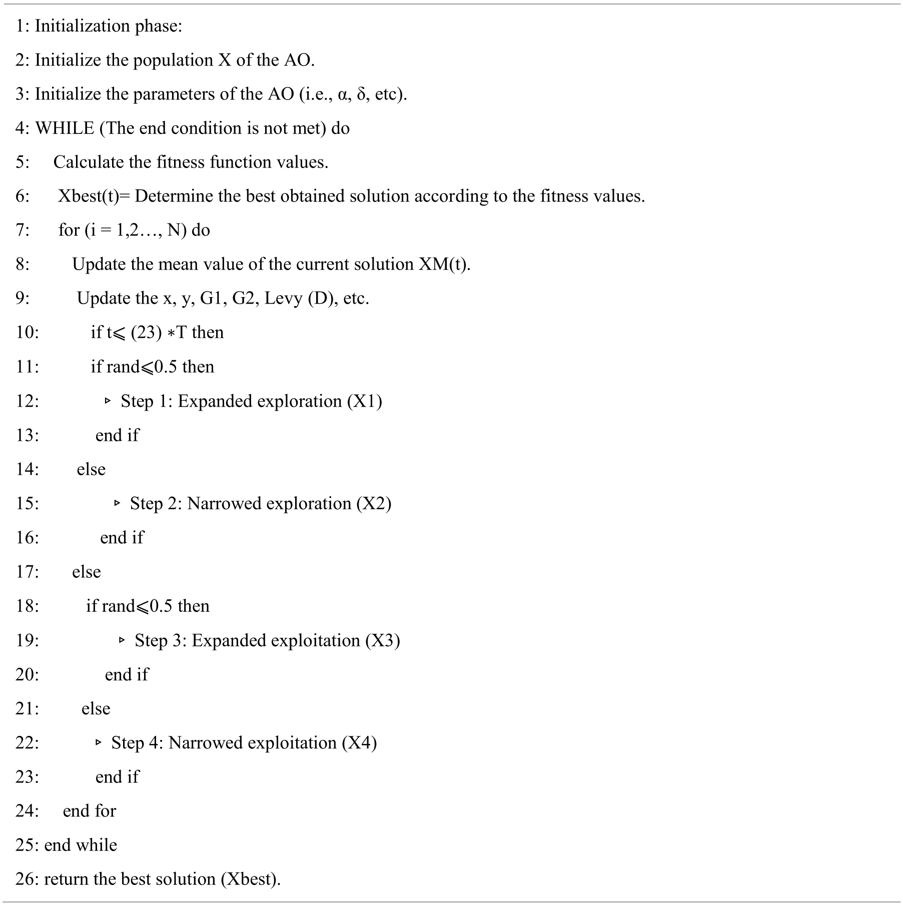
$X_4(t+1)$ is the best solution. *QF* is the Quality Function, and it is utilized to balance the search methods and calculated using Eq. (9). *G1* indicates the movement of AO to follow the prey, which is calculated using Eq. (10). *G2* presents decreasing values from 2 to 0, which refers to the incline flying of the AO that follows prey during the hunt from position (1) to position (t), which is calculated using Eq. (11). *X(t)* is the current solution at the t^{th} iteration.

$$QF(t) = t^{\frac{2 \times rand - 1}{(1-T)^2}} \tag{9}$$

$$G_1 = 2 \times rand - 1 \tag{10}$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{11}$$

$QF(t)$ is the quality function value at the t^{th} iteration. *rand* is a random number between 0 and 1. t and T is the current run and the maximum number of iteration, respectively. *Levy(D)* is the levy flight distribution function. The pseudo-code of the original AO is presented in Algorithm 1.



```
Initialization phase:
Initialize the population X of the AO.
Initialize the parameters of the AO (i.e., α, δ, etc).
WHILE (The end condition is not met) do
    Calculate the fitness function values.
    Xbest(t)= Determine the best obtained solution according to the fitness values.
    for (i = 1,2…, N) do
        Update the mean value of the current solution XM(t).
        Update the x, y, G1, G2, Levy (D), etc.
            if t⩽ (23) ∗T then
            if rand⩽0.5 then
                ▹ Step 1: Expanded exploration (X1)
              end if
        else
                ▹ Step 2: Narrowed exploration (X2)
              end if
       else
           if rand⩽0.5 then
                ▹ Step 3: Expanded exploitation (X3)
               end if
        else
             ▹ Step 4: Narrowed exploitation (X4)
             end if
    end for
end while
return the best solution (Xbest).
```

Algorithm 1 Pseudo-code of the Aquila Optimizer

3.2 Modified aquila optimizer algorithm (mAO)

In this section, we proposed the modified Aquila optimization algorithm. The main modification is replacing the Levy distribution function with the Cauchy distribution function in narrowed exploration and exploitation steps.

The Cauchy distribution function is a dominant search tool in literature. For example, in [50], Ant colony optimization with Cauchy and greedy Levy mutations for multilevel COVID-19 X-ray image segmentation was used. The Cauchy distribution was found to improve the convergence rate [64], so in this paper, we employed the Cauchy function to enhance the optimization performance of the AO algorithm. The proposed mAO algorithm is represented to enhance the performance of IDS by performing a feature selection process to select the best set of features to be classified by SVM for intrusion detection, as shown in Fig. 2.

The first step is to prepare the data by splitting the Dataset of the Dataset (NSL-KDD) into two groups (training and testing). After that, Normalization is done on the

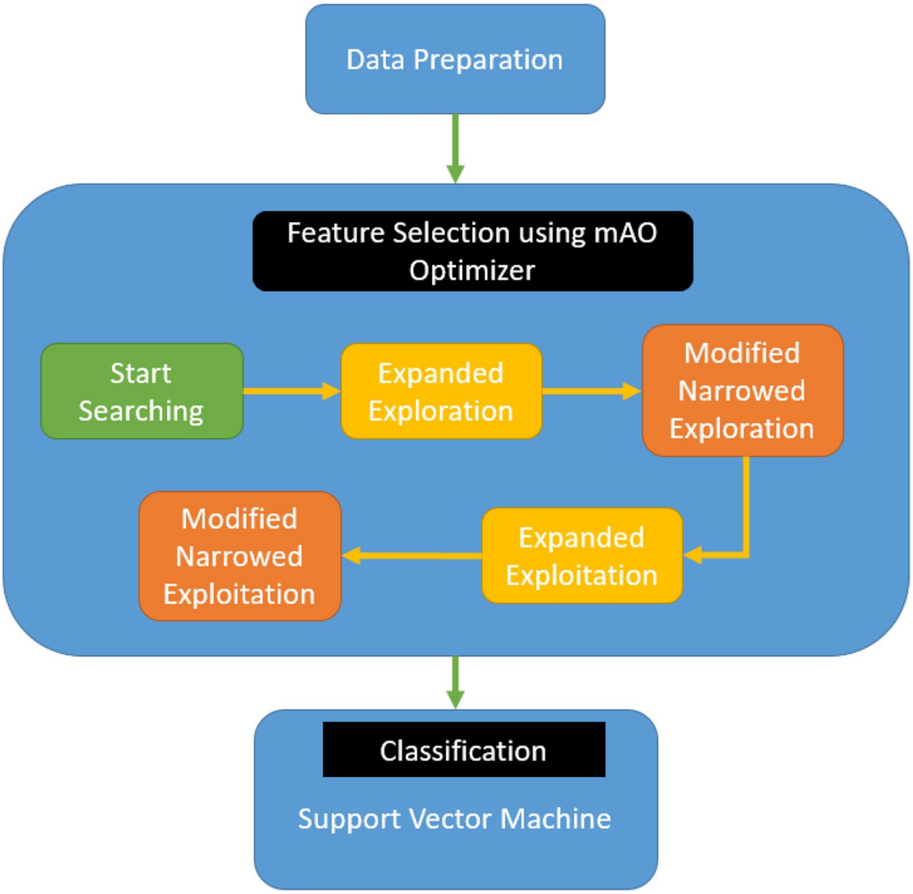


Fig. 2 The proposed methodology

Dataset because of the wide range of values of the features that cause significant faults in the classification procedure, so it is necessary to make the range of feature values from 0 to 1. The next step is to obtain the optimal set of features using mAO, represented in four methods: expanded exploration, modified narrowed exploration, expanded exploitation, and finally, modified narrowed exploitation, at which the optimal solution is obtained. The last step of the proposed method is to classify the obtained subset of features through SVM to normal and attack classes and compare the results of the testing data to the training data to calculate the evaluation measures: accuracy, detection rate, number of features, false alarm, and execution time.

3.2.1 The main steps of the proposed mAO

Expanded exploration The first step of the optimization process is represented mathematically as in Eq. (12).

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + \left(X_M(t) - X_{best}(t) * rand\right) \tag{12}$$

where, $X_1(t+1)$ is the solution of the next run of t, found by (X1). $X_{best}(t)$ is the best-found solution up to *tth* iteration. The formula $\left(1 - \frac{t}{T}\right)$ is utilized to control the wide search by the number of iterations. $X_{best}(t)$ is the locations mean value of the current solutions at *tth* iteration and calculated using Eq. (13). Rand is a random value between 0 and 1. t and T is the current run and the maximum iteration number, respectively.

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t), \forall j = 1, 2, \dots.., Dim \tag{13}$$

where *Dim* is the size of the problem and N is the number of nominee solution.

Modified narrowed exploration The second step of the optimization process is mathematical presented in Eq. (14).

$$X_2(t+1) = X_{best}(t) \times Cauchy(D) + X_R(t) + (y - x) * rand \tag{14}$$

where $X_2(t+1)$ is the solution to the next run. D is the dimension space, and Cauchy (D) is the Cauchy distribution function, calculated using Eq. (15). $X_R(t)$, is a random solution taken in the range of [1 N] at the ith iteration.

$$Cauchy(D) = \mu + s * (\tan(\pi * (rand - 0.5))) \tag{15}$$

where median $\mu = 0$ and scale $s = 1$; *rand* is a normally distributed stochastic number.

Expanded exploitation The third step starts to converge the search space for a solution, and the mathematical representation is presented in Eq. (16).

$$X_3(t+1) = \left(X_{best}(t) - X_M(t)\right) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta \tag{16}$$

where $X_3(t+1)$ the solution of the next run of t. and $X_{best}(t)$ is the estimated location of the prey until i^{th} iteration, and $X_M(t)$ is the average value of the current solution at t^{th} iteration. Rand is a random number that varies from 0 to 1. α and δ are the exploitation tuning restrictions equal to (0.1). *LB* refers to the lower bound, and *UB* refers to the upper bound.

Modified narrowed exploitation The last step of the optimization process narrows the search space to find the final optimal solution, which is represented mathematically in Eq. (17).

$$\begin{gathered} X_4(t+1) = QF \times X_{best}(t) - \left(G_1 \times X(t) \times rand\right) \\ -G_2 \times Cauchy(D) + rand \times G_1 \end{gathered} \tag{17}$$

$X_4(t+1)$ is the best solution. *QF* is the Quality Function, which is utilized to balance the search methods and calculated using Eq. (18). *G1* indicates the movement of AO to follow the prey, which is calculated using Eq. (19). *G2* presents decreasing values from 2 to 0, which refers to the incline flying of the AO that follows prey during the hunt from position (1) to position (t), which is calculated using Eq. (20). $X(t)$ is the current solution at the *tth* iteration.

$$QF(t) = t^{\frac{2\times rand - 1}{(1-T)^2}} \tag{18}$$

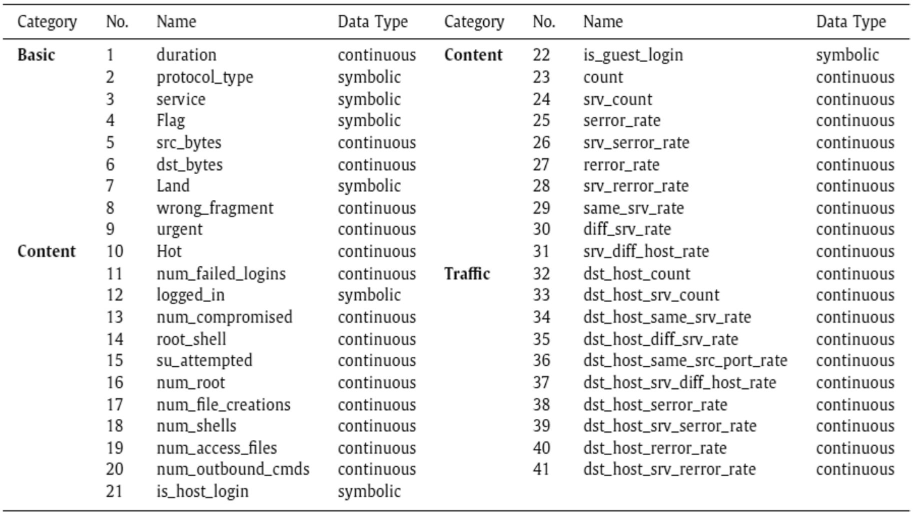
$$G_1 = 2 \times rand - 1 \tag{19}$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{20}$$

$QF(t)$ is the quality function value at the *tth* iteration, and *rand* is a random number between 0 & 1. t and T is the current run and the maximum number of iteration, respectively. Cauchy (D) is the Cauchy distribution function. The main procedure of the proposed algorithm is presented in Algorithm 2.

```
Initialization phase:
Initialize the population X of the mAO.
Initialize the parameters of the AO (i.e., α, δ, etc).
WHILE (The end condition is not met) do
  Calculate the fitness function values.
  Xbest(t)= Determine the best obtained solution according to the fitness values.
  for (i = 1,2…, N) do
    Update the mean value of the current solution XM(t).
    Update the x, y, G1, G2, Cauchy (D) , etc.
      if t⩽ (23) ∗T then
      if rand⩽0.5 then
        ▹ Step 1: Expanded exploration (X1)
        Update the current solution using Eq. (13).
        if Fitness (X1(t + 1)) < Fitness(X(t)) then
        X(t) = (X1(t + 1))
        if Fitness (X1(t + 1)) < Fitness (Xbest(t)) then
          Xbest(t) =X1(t + 1)
        end if
      end if
    else
        ▹ Step 2: Modified Narrowed exploration (X2)
        Update the current solution using Eq. (15).
        if Fitness (X2(t + 1)) < Fitness(X(t)) then
          X(t) = (X2(t + 1))
          if Fitness (X2(t + 1)) < Fitness (Xbest(t)) then
            Xbest(t) =X2(t + 1)
          end if
        end if
      end if
    else
      if rand⩽0.5 then
        ▹ Step 3: Expanded exploitation (X3)
        Update the current solution using Eq. (17).
        if Fitness (X3(t + 1)) < Fitness(X(t)) then
          X(t) = (X3(t + 1))
          if Fitness (X3(t + 1)) < Fitness (Xbest(t)) then
        Xbest(t) =X3(t + 1)
          end if
        end if
    else
      ▹ Step 4: Modified Narrowed exploitation (X4)
      Update the current solution using Eq. (18).
      if Fitness (X4(t + 1)) < Fitness(X(t)) then
        X(t) = (X4(t + 1))
          if Fitness (X4(t + 1)) < Fitness (Xbest(t)) then
            Xbest(t) =X4(t + 1)
          end if
        end if
      end if
        end if
  end for
end while
return the best solution (Xbest).
```

Algorithm 2 Pseudo-code of the Modified Aquila Optimizer



Category	No.	Name	Data Type	Category	No.	Name	Data Type
Basic	1	duration	continuous	**Content**	22	is_guest_login	symbolic
	2	protocol_type	symbolic		23	count	continuous
	3	service	symbolic		24	srv_count	continuous
	4	Flag	symbolic		25	serror_rate	continuous
	5	src_bytes	continuous		26	srv_serror_rate	continuous
	6	dst_bytes	continuous		27	rerror_rate	continuous
	7	Land	symbolic		28	srv_rerror_rate	continuous
	8	wrong_fragment	continuous		29	same_srv_rate	continuous
	9	urgent	continuous		30	diff_srv_rate	continuous
Content	10	Hot	continuous		31	srv_diff_host_rate	continuous
	11	num_failed_logins	continuous	**Traffic**	32	dst_host_count	continuous
	12	logged_in	symbolic		33	dst_host_srv_count	continuous
	13	num_compromised	continuous		34	dst_host_same_srv_rate	continuous
	14	root_shell	continuous		35	dst_host_diff_srv_rate	continuous
	15	su_attempted	continuous		36	dst_host_same_src_port_rate	continuous
	16	num_root	continuous		37	dst_host_srv_diff_host_rate	continuous
	17	num_file_creations	continuous		38	dst_host_serror_rate	continuous
	18	num_shells	continuous		39	dst_host_srv_serror_rate	continuous
	19	num_access_files	continuous		40	dst_host_rerror_rate	continuous
	20	num_outbound_cmds	continuous		41	dst_host_srv_rerror_rate	continuous
	21	is_host_login	symbolic				

Fig. 3 Features of the KDD dataset

Table 2 Type of attacks

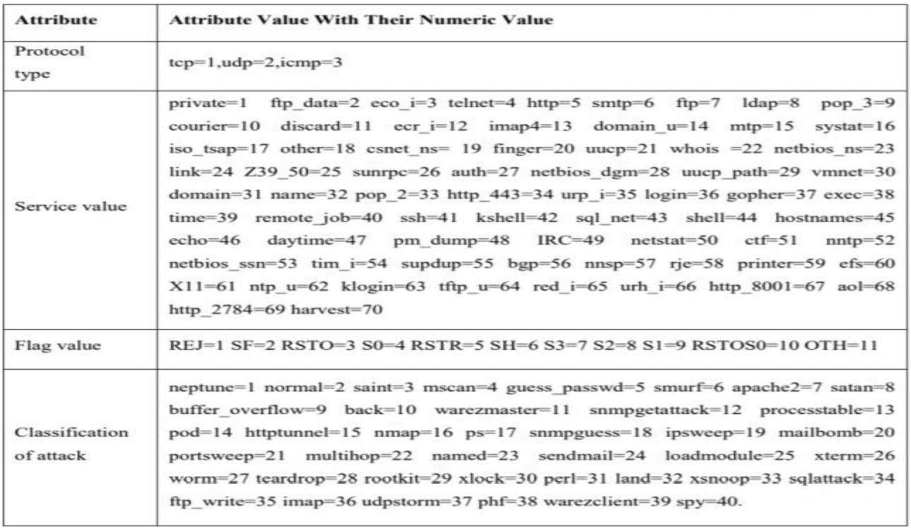
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
DoS	Back, Land, Neptune, Pod, Smurf, teardrop, Mailbomb, Processtable, Udpstorm, Apache2, Worm
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Xlock, xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow,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

3.3 NSL-KDD dataset

In this paper, KDD is a famously used online dataset for experimental testing of the proposed methods. The KDD dataset has 41 features in each set, as shown in Fig. 3. These sets are labeled as an attack or normal data, and each feature is categorized into three attribute types (Nominal, Binary, and Numeric) [65]."

The basic types of attacks in the KDD dataset are shown in Table 2:

- **Denial of Service Attacks (DoS)** happens when the user is restricted from accessing a service by an attack.



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. 4 Transform Methodology

- **User to Root Attacks (U2R)**: This happens when the attacker accesses the root system computer unauthorizedly.
- **Remote to Local attacks (R2L)**: happen when the attacker has unauthorized access from the root machine
- **Probing attacks** occur when the attacker probes the network to collect information about the system to avoid the security system.

3.3.1 Preparing dataset

Several steps have to be done before using the Dataset for selecting the optimal features. The first step is to convert the categorical features to numeric features to enable SVM to deal with them because machine learning algorithms do not process non-numeric features. The conversion mechanism is applied in Fig. 4, representing the feature values with their numeric values used to transform categorical data into numeric.

The second step is Normalization, which is needed because of the feature values range that leads to major errors in classification. Normalization means unifying the range of values of features from 0 to 1. Due to the non-uniformly distributed features of the KDD dataset, the Max–Min Normalization method is used as in Eq. (21).

$$X' = (Original\ value - Min\_Value)/(Maxvalue - Min\_Value) \tag{21}$$

where X' is the normalized value. The last step is to choose the most relevant features to be processed by SVM, which is done by the proposed (mAO) method.

4 Experiments and results

In this chapter, the efficiency of the suggested method using mAO is examined using KDD 99 as a test dataset, and the results are evaluated against the original AO method, Modified GWO-based IDS (mGWO), and PSO-SVM as in (Safaldin, Otair & Abualigah 2020) and AOA SVM Intrusion detection (AS_IDS) as in (Faten Q., 2021). We have taken a systematic approach to select the parameters, ensuring that they align with the best-known configurations in the field. Our parameter choices are grounded in the extensive research and experimentation reported in the literature. The simulation of the test was accomplished using a Personal computer with a Core i7 2.4 GHz CPU, 8 GB RAM, and MATLAB 2020.

4.1 Evaluation measures

The following evaluation measures are used to assess the results obtained by the proposed method. Various criteria are used to evaluate the performance of the proposed method, such as; accuracy, detection rate, false alarm rate, number of features, and execution time.

- **Accuracy:** the percentage of data correctly classified as a true positive (TP) and true negative (TN).
- **Detection Rate (DR):** "is the ratio of true positive to the total non-self-samples found by Dataset, where TP and FN are the tallies of true positive and false negative.

$$DR = TP/(TP + FN) \tag{22}$$

- **False Alarm Rate (FAR):** is the ratio of false-positive to the total self-samples recognized by the Dataset, where FP and TN are the tallies of false-positive and true negative, as shown below.

$$FAR = FP/(FP + TN) \tag{23}$$

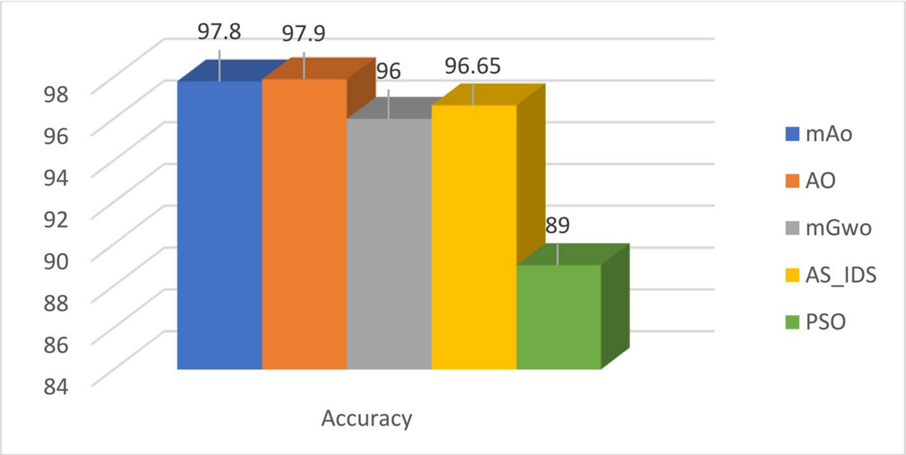


Fig. 5 Accuracy values of compared methods

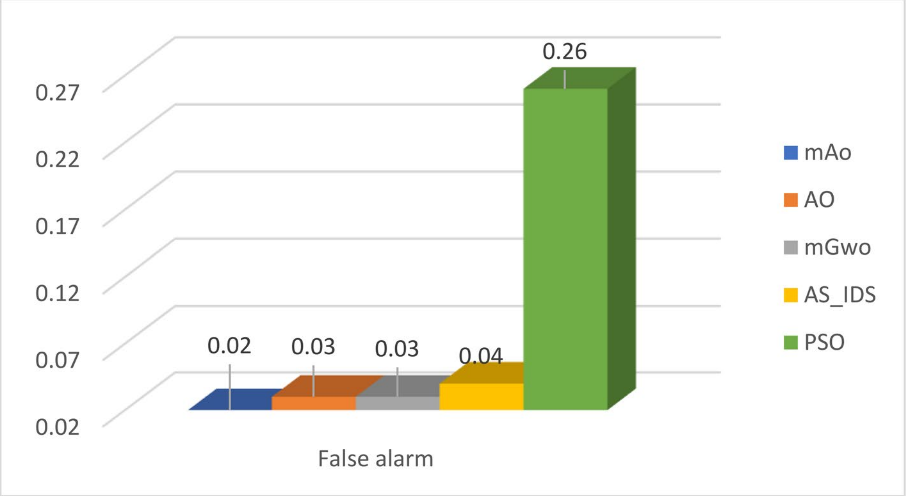


Fig. 6 False alarm values of compared methods

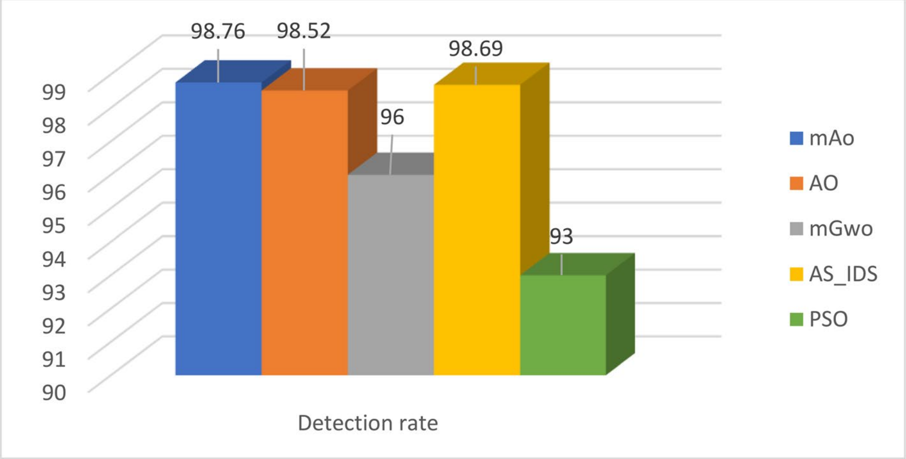


Fig. 7 Detection rate values of compared methods

- **Number of Features**: this is the number of selected features used to classify normal and abnormal activities.
- **Execution time:** is the time used to complete the process of classification.

4.2 Results and discussion

In this section, the test results of the proposed method are analyzed and compared to the results obtained by the AO algorithm, mGWO, and AS_IDS as they have remarkable results regarding the evaluation metrics, including accuracy, detection rate, and the number of features, execution time and false alarm rate.

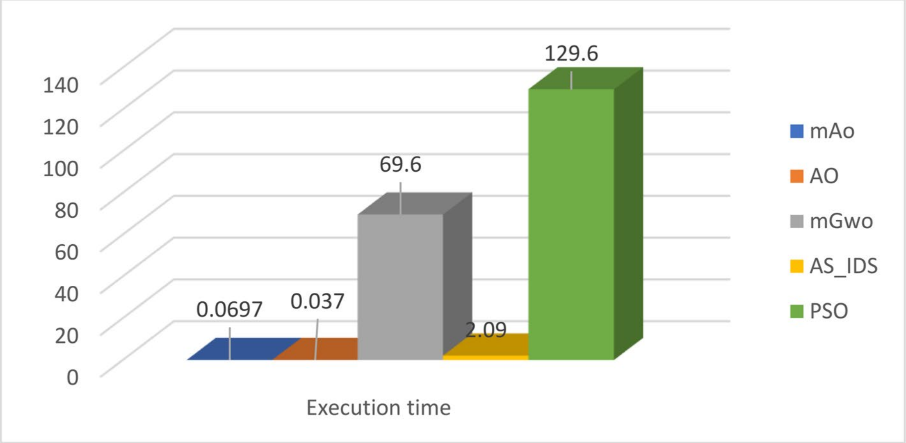


Fig. 8 Execution time (hours) values of compared methods

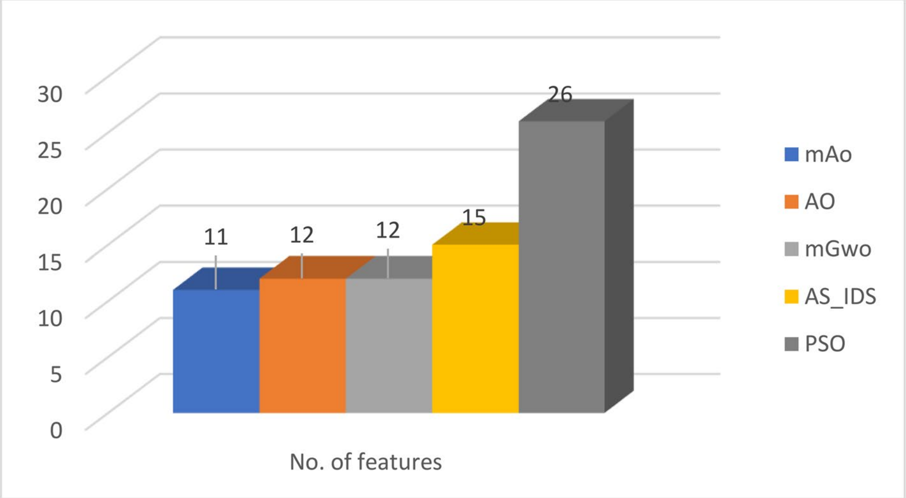


Fig. 9 Number of features of compared methods

Table 3 The results of the proposed technique based on the evaluation metrics

Method	Accuracy	Detection Rate	False Alarm	Process Time (hours)	Feature Number
mGWO	96%	96%	0.03	69.6	12
AO	97.9%	98.52%	0.03	0.037	12
mAO	97.8%	98.76%	0.02	0.0697	11
AS_IDS	96.65%	98.69%	0.04	2.09	15
PSO	89%	93%	0.26	129.6	26
Enhancement	AO 0.1%	mAO 0.24%	mAO 33%	AO 88%	mAO 8.3%

As in Fig. 5, the mAO method and AO have very close results with 97.8% and 97.9%, respectively, which are higher than the accuracy of mGWO, AS_IDS and PSO with 96%, 96.65%, and 89%, respectively. This indicates that the modification of mAO did not affect the performance of the original Aquila regarding accuracy. On the other hand, mAO outperforms the other compared method by a significant difference.

Considering the false alarm results, Fig. 6 shows that the proposed method (mAO) has least value of 0.02, which is less than AO and mGWO, with values of 0.03 for both methods, a value of 0.04 for the AS_IDS method, and 0.26 for PSO method. This indicates that mAO method experiences much fewer errors when detecting suspicious data and confirms the superiority of mAO.

As shown in Fig. 7, the detection rate of the proposed method mAO is 98.76%, and it is higher than AO, mGWO, PSO, and AS_IDS, with 98.52%, 96%, 93%, and 98.69%, respectively. This indicates the superiority of the mAO method and reflects the high accuracy in detecting suspicious data among network traffic. Also, this result proves that mAO method is superior to other meta-heuristic algorithms.

Figure 8 shows that the proposed method mAO has a relatively higher execution time than AO, with values of 0.0697 and 0.037 h, respectively. Although it is considered an excellent result compared to other methods, mGWO, PSO and AS_IDS result with a value of 69.6, 129.6, and 2.09 h, respectively. This indicates that the mAO method is

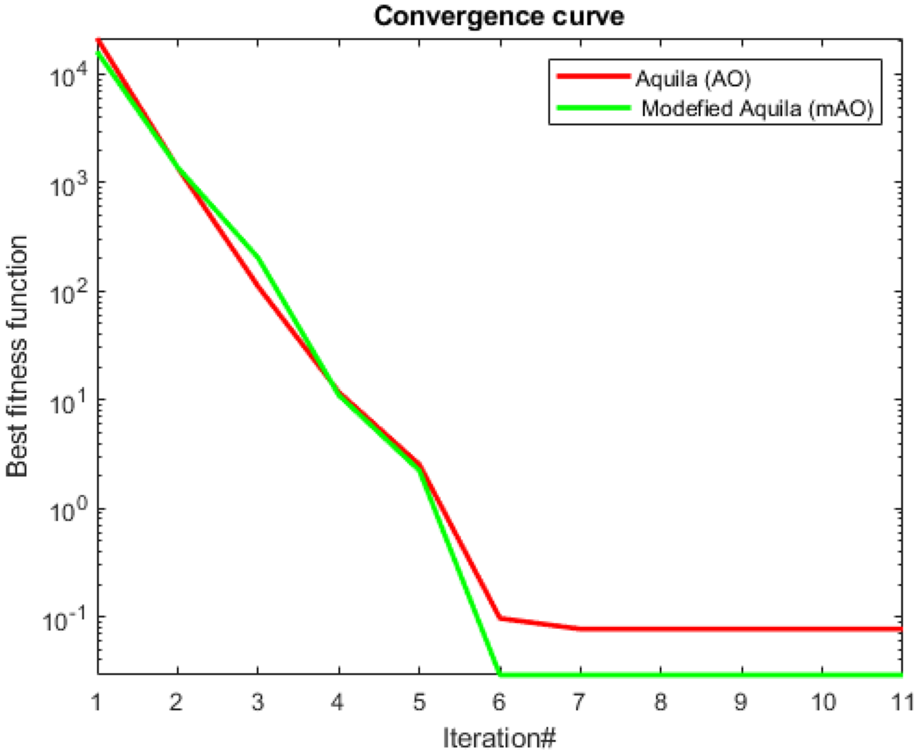


Fig. 10 Convergence rate of mAO and AO

much better, outperforms other methods, and consumes less time, leading to a more efficient and secure IDS.

As shown in Fig. 9, the proposed mAO method has resulted in 11 features less than AO and mGWO results of 12 features, 15 for AS_IDS and 26 for PSO. This shows that mAO has selected a small number of features compared to the original data of 41 features, leading to faster IDS performance.

In summary, mAO method is superior to mGWO, AS_IDS, PSO, and AO methods in terms of detection rate and false alarm and has fewer features than the original data. However, the accuracy value of mAO is slightly less than the AO method, which is insignificant regarding the high accuracy achieved by mAO method. This means that mAO is very effective for feature selection optimization than other well-known methods i.e., AO and mGWO as demonstrated in Table 3.

The enhancement percentage is calculated by comparing the results of mAO with the best result among mGwo and AO using the following Equation:

$$EnhancmentPercentage = (|OldValue - NewValue|)/OldVlaue$$

Old value: is the best value between AO and mGWO for example, regarding detection rate, the best value is found by mAO. Based on that, the enhancement done is calculated as follows:

$$\begin{aligned}\mathrm{Theenhancement} &= ((\mathrm{mAOdetectionrate} - \mathrm{AOdetectionrate})/\mathrm{mAOdetectionrate}) * 100\% \\ &= ((98.76 - 98.52)/98.76) * 100\% = 0.24\%\end{aligned}$$

Also, it is noticed that the convergence rate of mAO is better than AO, which means that mAO reaches a better fitness value than AO within the same number of iterations, as shown in Fig. 10. The convergence rate is compared between AO and mAO only because other methods' results are obtained from the literature.

5 Conclusion and future work

IDS systems predominantly suffer from low detection rates and low accuracy. This paper suggested a method using (mAO) for feature selection to enhance the performance of the SVM intrusion detection system. The suggested method was evaluated using KDD'99 datasets. The results indicated that the proposed method had accomplished excellent results considering the accuracy, detection rate, false alarm rate, number of features, and execution time. The obtained outcomes of mAO are compared with mGWO, AS_IDS, and AO algorithms. The comparison indicated that the suggested method is better than the compared methods, considering the number of features, detection rate, and false alarms with 11, 98.76%, and 0.02, respectively. Also, the suggested method is better than the AS_IDS and mGWO regarding accuracy and execution time by 97.8% and 0.0697, respectively. Overall, the suggested (mAO) method has competitive results regarding the evaluation metrics, especially when considering the selection of the least number of features, which significantly reduces the execution time and reflects the efficiency and security of IDS. The mAO method can be verified with various datasets and different classifiers such as decision trees. Also, the proposed method can be used to solve different real-world optimization problems such as image segmentation, clustering, and cloud computing task scheduling.

Funding No funding.

Data availability Data is available from the authors upon reasonable request.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Kamilaris A, Pitsillides A (2016) Mobile phone computing and the Internet of things: A survey. IEEE Internet Things J 3(6):885–898
2. Khan MA, Hussain S (2020) Energy efficient direction-based topology control algorithm for WSN. Wirel Sens Netw 12(3):37–47
3. Ndunagu JN et al (2022) Development of a Wireless Sensor Network and IoT-based Smart Irrigation System. Appl Environ Soil Sci 2022
4. Chang J-Y, Shen T-H (2016) An efficient tree-based power saving scheme for wireless sensor networks with mobile sink. IEEE Sens J 16(20):7545–7557
5. Jondhale SR, Maheswar R, Lloret J (2022) Fundamentals of Wireless Sensor Networks. Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks. Springer, pp 1–19
6. Ullo SL, Sinha GR (2020) Advances in smart environment monitoring systems using IoT and sensors. Sensors 20(11):3113
7. Nguyen LT et al (2008) An energy efficient routing scheme for mobile wireless sensor networks. in 2008 IEEE International Symposium on Wireless Communication Systems. IEEE.
8. Balid W, Tafish H, Refai HH (2017) Intelligent vehicle counting and classification sensor for real-time traffic surveillance. IEEE Trans Intell Transp Syst 19(6):1784–1794
9. Du X, Chen H-H (2008) Security in wireless sensor networks. IEEE Wirel Commun 15(4):60–66
10. Sert OC et al (2022) Temptracker: a service oriented temporal natural language processing based tool for document data characterization and social network analysis. Int Arab J Inf Technol 19(3):342–352
11. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52(12):2292–2330
12. Sharma H, Haque A, Blaabjerg F (2021) Machine learning in wireless sensor networks for smart cities: a survey. Electronics 10(9):1012
13. Al-Fuhaidi B et al (2020) An efficient deployment model for maximizing coverage of heterogeneous wireless sensor network based on harmony search algorithm. J Sens 2020
14. Sun Z et al (2017) An intrusion detection model for wireless sensor networks with an improved V-detector algorithm. IEEE Sens J 18(5):1971–1984
15. Latif S et al (2021) Intrusion detection framework for the Internet of things using a dense random neural network. IEEE Trans Industr Inf 18(9):6435–6444
16. Abdel-Basset M et al (2021) Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks. IEEE Internet Things J 8(15):12251–12265
17. Salim MM, Singh SK, Park JH (2021) Securing Smart Cities using LSTM algorithm and lightweight containers against botnet attacks. Appl Soft Comput 113:107859
18. Singh SK et al (2021) DeepBlockScheme: A deep learning-based blockchain driven scheme for secure smart city. HCIS 11(12):1–13
19. Huang X (2021) Network intrusion detection based on an improved long-short-term memory model in combination with multiple spatiotemporal structures. Wirel Commun Mob Comput 2021
20. Mohammadi M et al (2021) A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. J Netw Comput Appl 178:102983
21. Makkar A, Park JH (2022) SecureCPS: Cognitive inspired framework for detection of cyber attacks in cyber–physical systems. Inf Process Manage 59(3):102914

22. Karami A (2018) An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities. Expert Syst Appl 108:36–60
23. Mohammadi S et al (2019) Cyber intrusion detection by combined feature selection algorithm. J Inf Secur Appl 44:80–88
24. Abualigah L et al (2023) Revolutionizing sustainable supply chain management: A review of metaheuristics. Eng Appl Artif Intell 126:106839
25. Safaldin M, Otair M, Abualigah L (2021) Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. J Ambient Intell Humaniz Comput 12(2):1559–1576
26. Otair M et al (2022) An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. Wireless Netw 28(2):721–744
27. Al-Shourbaji I et al (2023) Artificial Ecosystem-Based Optimization with Dwarf Mongoose Optimization for Feature Selection and Global Optimization Problems. Int J Comput Intell Syst 16(1):1–24
28. Chen H et al (2023) Hybrid slime mold and arithmetic optimization algorithm with random center learning and restart mutation. Biomimetics 8(5):396
29. Huh J-H (2018) Implementation of lightweight intrusion detection model for security of smart green house and vertical farm. Int J Distrib Sens Netw 14(4):1550147718767630
30. Houssein EH et al (2022) An efficient equilibrium optimizer with support vector regression for stock market prediction. Neural Comput Appl 34(4):3165–3200
31. Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer
32. Mostafa RR et al (2022) Boosting chameleon swarm algorithm with consumption AEO operator for global optimization and feature selection. Knowl-Based Syst 246:108743
33. Wu D et al (2022) Enhance teaching-learning-based optimization for tsallis-entropy-based feature selection classification approach. Processes 10(2):360
34. Abualigah L, Diabat A (2022) Chaotic binary group search optimizer for feature selection. Expert Syst Appl 192:116368
35. BaturŞahin C, Abualigah L (2021) A novel deep learning-based feature selection model for improving the static analysis of vulnerability detection. Neural Comput Appl 33(20):14049–14067
36. Panousopoulou A, Azkune M, Tsakalides P (2016) Feature selection for performance characterization in multi-hop wireless sensor networks. Ad Hoc Netw 49:70–89
37. Zhang Y (2012) Support vector machine classification algorithm and its application. in International conference on information computing and applications. Springer.
38. Nadimi-Shahraki MH et al (2021) Mtv-mfo: Multi-trial vector-based moth-flame optimization algorithm. Symmetry 13(12):2388
39. Nadimi-Shahraki MH et al (2021) An improved moth-flame optimization algorithm with adaptation mechanism to solve numerical and mechanical engineering problems. Entropy 23(12):1637
40. Abualigah L et al (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250
41. Agushaka JO, Ezugwu AE, Abualigah L (2022) Dwarf mongoose optimization algorithm. Comput Methods Appl Mech Eng 391:114570
42. Abualigah L et al (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
43. Oyelade ON et al (2022) Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. IEEE Access 10:16150–16177
44. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
45. Zamani H, Nadimi-Shahraki MH, Gandomi AH (2022) Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. Comput Methods Appl Mech Eng 392:114616
46. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
47. Abualigah L et al (2022) Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. Expert Syst Appl 191:116158
48. Nadimi-Shahraki MH et al (2022) GGWO: Gaze cues learning-based grey wolf optimizer and its applications for solving engineering problems. J Comput Sci 61:101636
49. Nadimi-Shahraki MH et al (2021) Migration-based moth-flame optimization algorithm. Processes 9(12):2276
50. Liu G et al (2022) An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. Sensors 22(4):1407
51. Ifzarne S, Hafidi I, Idrissi N (2021) Secure data collection for wireless sensor network. Emerging Trends in ICT for Sustainable Development. Springer, pp 241–248
52. Ifzarne S et al. (2021) Anomaly detection using machine learning techniques in wireless sensor networks. in Journal of Physics: Conference Series. IOP Publishing.

53. Alazzam H, Sharieh A, Sabri KE (2020) A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst Appl 148:113249
54. Lv L et al (2020) A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine. Knowl-Based Syst 195:105648
55. Khalvati L, Keshtgary M, Rikhtegar N (2018) Intrusion detection based on a novel hybrid learning approach. J AI Data Min 6(1):157–162
56. Javeed D et al (2023) An Explainable and Resilient Intrusion Detection System for Industry 5.0. IEEE Transactions on Consumer Electronics
57. Xun Y et al. (2023) Side Channel Analysis: A Novel Intrusion Detection System Based on Vehicle Voltage Signals. IEEE Transactions on Vehicular Technology
58. Javeed D et al (2023) An Intelligent Intrusion Detection System for Smart Consumer Electronics Network. IEEE Transactions on Consumer Electronics
59. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381
60. Farahani G (2020) *Feature selection based on cross-correlation for the intrusion detection system.* Security and Communication Networks, 2020
61. Ekinci S et al (2022) An effective control design approach based on novel enhanced aquila optimizer for automatic voltage regulator. Artif Intell Rev 1–32.
62. Ewees AA et al (2022) A cox proportional-hazards model based on an improved aquila optimizer with whale optimization algorithm operators. Mathematics 10(8):1273
63. Wang S et al (2021) An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. Processes 9(9):1551
64. Salgotra R, Singh U (2017) Application of mutation operators to flower pollination algorithm. Expert Syst Appl 79:112–129
65. Tavallaee M et al (2009) A detailed analysis of the KDD CUP 99 data set. in 2009 IEEE symposium on computational intelligence for security and defense applications. Ieee

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Laith Abualigah[1,2,3,4,5,6,7] · Saba Hussein Ahmed[6] · Mohammad H. Almomani[8] · Raed Abu Zitar[9] · Anas Ratib Alsoud[3] · Belal Abuhaija[10] · Essam Said Hanandeh[11] · Heming Jia[12] · Diaa Salama Abd Elminaam[13,14] · Mohamed Abd Elaziz[15,16,17]

✉ Laith Abualigah
Aligah.2020@gmail.com

✉ Belal Abuhaija
babuhaij@kean.edu

[1] Computer Science Department, Al Al-Bayt University, Mafraq 25113, Jordan

[2] Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon

[3] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan

[4] MEU Research Unit, Middle East University, Amman 11831, Jordan

[5] Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

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

[7] School of Engineering and Technology, Sunway University Malaysia, Petaling Jaya 27500, Malaysia

[8] Department of Mathematics, Facility of Science, The Hashemite University, P.O box 330127, Zarqa 13133, Jordan

[9] Sorbonne Center of Artificial Intelligence, Sorbonne University-Abu Dhabi, Abu Dhabi, UAE

[10] Department of Computer Science, Wenzhou-Kean University, Wenzhou, China

[11] Department of Computer Information System, Zarqa University, P.O. Box 13132, Zarqa, Jordan

[12] School of Information Engineering, Sanming University, Sanming 365004, China

[13] Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt

[14] Computer Science Department, Faculty of Computer Science, Misr International University, Cairo 12585, Egypt

[15] Faculty of Computer Science & Engineering, Galala University, Suze 435611, Egypt

[16] Artificial Intelligence Research Center (AIRC), Ajman University, 346 Ajman,
United Arab Emirates

[17] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt