IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 June 2023, accepted 25 June 2023, date of publication 28 June 2023, date of current version 6 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3290895

RESEARCH ARTICLE

Quantum Artificial Hummingbird Algorithm for Feature Selection of Social IoT

MOHAMED ABD ELAZIZ[1,2,3,4,5], ABDELGHANI DAHOU[6], MOHAMMED AZMI AL-BETAR[3,7], SHAKER EL-SAPPAGH[2,8], DIEGO OLIVA[9], (Senior Member, IEEE), AND AHMAD O. ASEERI[10], (Member, IEEE)

[1]Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
[2]Faculty of Computer Science and Engineering, Galala University, Suez 435611, Egypt
[3]Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates
[4]Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon
[5]MEU Research Unit, Middle East University, Amman 11831, Jordan
[6]Mathematics and Computer Science Department, University of Ahmed Draia, Adrar 01000, Algeria
[7]Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Irbid 19117, Jordan
[8]Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
[9]Depto. de Innovación Basada en la Información y el Conocimiento, Universidad de Guadalajara, CUCEI, Guadalajara 44430, Mexico
[10]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Corresponding authors: Mohamed Abd Elaziz (abd_el_aziz_m@yahoo.com) and Ahmad O. Aseeri (a.aseeri@psau.edu.sa)

This work was supported by the Deputyship for Research and Innovation, Ministry of Education, Saudi Arabia, under Project IF-PSAU-2022/01/19574.

ABSTRACT The Internet of Things (IoT) benefits from social networking platforms in establishing and enhancing social-oriented services, information, and autonomous social relationships. Social IoT (SIoT) systems can boost the user experience in the real world in several applications, including healthcare, transportation, and entertainment. However, the collected data from various interconnected SIoT systems is massive, demanding robust and efficient processing algorithms, feature extraction, selection, and inference. This work presents an enhanced Artificial Hummingbird algorithm (AHA) for feature selection (FS). The enhanced version of AHA is performed using the advantages of Quantum-based optimization. The main aim of using Quantum is to improve the population's exploration ability while discovering feasible regions. Extensive experiments utilizing eighteen UCI datasets were conducted to validate the developed FS method, QAHA. The QAHA is compared with other FS methods, and the experimental established its efficiency. Moreover, a set of four datasets from SIoT are used to evaluate the applicability of QAHA to the real-world setting. The results using these datasets indicate the high performance of QAHA to increase the accuracy by decreasing the number of features. In the case of UCI datasets, the average accuracy of the developed QAHA is 93% among the eighteen datasets. Whereas, In the case of the SIoT datasets, the developed QAHA has an accuracy of nearly 90.7%, 98.7%, 92.2%, and 84.6% for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively.

INDEX TERMS Artificial hummingbird algorithm, feature selection, swarm optimization, quantum-based optimization, social Internet of Things.

I. INTRODUCTION

In recent years, the Internet of Things (IoT) platforms integrated a paradigm to enhance the exchanged information between millions of objects called the Social Internet of Things (SIoT). The SIoT paradigm intelligently connects social networks and IoT to create social communications between humans and objects. The development of the SIoT ecosystem needs robust algorithms to build a robust system in terms of architecture, system components, shared information, feature extraction and selection, scalability, relation management, web services, and parameters [1]. However, with the massive amounts of data from different sources such as social network platforms, smartphones, and

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

sensors, the collected information should be processed and analyzed quickly to provide quick reports, diagnoses, and analyses. In addition, the transferred information across independent objects can differ in operating platforms, data type, communication protocols, and object type. For instance, the SIoT ecosystem can benefit from autonomous communication between objects to create a social relation, such as managing interconnected objects in smart city infrastructure and providing better services, including transportation (shortest path), event monitoring on social media platforms, crisis monitoring, urban resources management, and healthcare [2].

Regardless of the existing data processing algorithms, the high volume of data needs practical tools to process and analyze it. The high dimensionality of the data sets could affect the performance of machine learning (ML) tools. More specifically, the classification is affected since the computational effort increases and the accuracy decreases. These demerits occur due to the data acquired containing redundant information and noise that demerits the methods applied to it. To overcome such problems, in the related literature, it is possible to find dimensionality reduction algorithms; these kinds of methods are divided into Feature Selection (FS) and Feature Extraction (FE). In general terms, FE helps to obtain the information that composes a data set. For example, from a set of images (from the same source), it is possible to apply different operators that extract the most relevant features from the objects and store them in a table. Meanwhile, FS is used over the data set to remove the noise, irrelevant, and redundant information that permits an increase in classifiers' performance. It is performed by selecting a subset of the data with the most relevant information from the original data set. The use of FS has been extended to different applications such as network intrusion detection [3], cardiovascular disease prediction [4], COVID-19 classification [5], and Alzheimer diagnosis [6] to mention some.

The FS methodologies are classified into the filter, wrapper, and embedded methods. The wrapper-based approaches use ML tools as classifiers to verify if the selected features are substantially representative. However, complex data sets with significant information require more computational resources. In contrast, the filter methods do not use ML mechanisms and only take the information from the data set to select the most representative information. To perform this task filter, FS approaches the intrinsic information of the data set that is analyzed by using metrics such as the correlation or distances to determine the subset of features. The embedded methods have integrated the FS as part of the learning process. This kind of approach combines the benefits of filter and wrapper methodologies. The embedded methods train a single ML model and select the features based on their importance but also according to the model. To improve the performance of FS methods, they have used different techniques, such as meta-heuristic algorithms.

Meta-heuristic algorithms (MA) are optimization tools that permit the exploration of a bounded search space with different search agents simultaneously. MA algorithms are divided into two prominent families; one is for the evolutionary-based methods as the genetic algorithms (GA) or the differential evolution (DE). In contrast, the second family is for the swarm-based approaches as the particle swarm optimization (PSO). In general, MA uses different rules that permit to use of operators that perform the exploration of the search space and the exploitation of the most prominent regions. In the related literature, it is possible to find modern MA that employs sophisticated rules and operators that permit finding the optimal solution with high performance. Some examples are atomic the orbital search (AOS) [7], the African vultures optimization algorithm (AVOA) [8], the golden eagle optimizer (GEO) [9], the honey badger algorithm (HBA) and the artificial hummingbird algorithm (AHA) [10] to mention some. Besides, the MA has been improved using hybrid methods or operators that permit better performance. For example, in [11], a multi-population whale optimization algorithm is proposed by clustering; here, the proposed method is used for solving optimization benchmark problems. The electromagnetic field optimization algorithm was also improved by using mutation operators for image thresholding [12]. In the same way, the reptile search algorithm has also been improved by using mutation techniques [13]. Another interesting approach includes the modification of the moth-flame optimizer with an adaptation mechanism for mechanical engineering problems [14].

Regarding the use of MA for FS, it is possible to find many implementations that include hybrid and improved approaches. For example, the binary version of the moth-flame optimization proposed in [15] for FS using medical data sets. The fractional-order comprehensive learning marine predators algorithm (MPA) is an improved approach for FS [16]. The chaotic maps are used in [17] to improve the gaining sharing knowledge-based optimization algorithm for FS. Also, the standard (MPA) is modified to create different binary versions for FS [18]. The information related to the use of MA for FS is vast, and it is growing with the number of MAs published every year. A recent review related to MA for solving the FS problem was presented in [19].

On the other hand, since MA needs to be adapted to solve problems from different fields of application [20], sometimes it takes effort to achieve the optimal solution. Even if the MA is good, some problems have multi-modal search spaces that affect the performance of the search process. In this way, MA can fail in suboptimal solutions. In the related literature, they have been proposed different methodologies that permit improvement of the performance of MA. One of these approaches takes advantage of modern physics, and using the motion in subatomic levels to define the movement of particles creates quantum behavior. The quantum behavior has been used to improve different MA as the quantum PSO (QPSO) [21], the quantum GA (QGA) [22], the quantum Henry gas solubility optimization (QHGSO) algorithm [23], the quantum simulated annealing (QSA) [24], the quantum salp swarm algorithm (QSSA) [25], the quantum cuckoo search algorithm (QCSA) [26] and the quantum marine predator's algorithm (QMPA) [27] to mention some. An interesting study of quantum MA can be found in [28]; besides, an analysis of the role of MA in quantum computers is presented in [29]. However, such methods are proposed for specific problems and suffer from some limitations such as premature convergence. This motivated us to propose an alternative FS approach based on a quantum version of the artificial hummingbird algorithm, QAHA, for solving the FS problem. The AHA algorithm has recently been introduced and applied to solve different problems for renewable energies [30]. It has also been used for the parameter identification of Li-Ion batteries in electric vehicles [31]. In [32], a modified version of AHA has been presented as FS using the random Opposition technique to handle the waste classification problem. This FS method has established its performance in comparison with other MH techniques.

Only a few applications can be found since the AHA was recently proposed. For that reason, it is necessary to test and improve this approach.

In this article, we introduce an improved version of the AHA using quantum-based optimization. This method is called quantum AHA (QAHA). To verify the performance of the proposed QAHA, we used the K-nearest neighbors (KNN) classifier to analyze 18 data sets with different characteristics taken from the UCI repository. As a case study, the QAHA is also applied for classifying data sets of Social IoT. Different metrics and comparisons with similar approaches validate the performance of the QAHA. In this context, the novelty of this proposal is integrating the quantum theory with the AHA and its application for SIoT data classification.

The main contribution of this work can be summarized as follows:

- Propose a modified version of the Artificial Hummingbird algorithm as FS for Social IoT.
- Using the strength of Quantum-based optimization technique to improve the performance of AHA, which aims to enhance its exploration ability.
- Evaluate the efficiency of developed QAHA using a set of UCI datasets. In addition, a set of four social databases related to IoT were also selected to validate the QAHA algorithm.
- Compare the results with the state-of-the-art FS methods.

The remainder paper is organized as follows: Section II-A presents the preliminaries of the AHA. Section III introduces the proposed approach. Section IV shows the experiments and comparisons. Finally, some conclusions are discussed in Section V.

II. BACKGROUND

A. ARTIFICIAL HUMMINGBIRD ALGORITHM

The artificial Hummingbird algorithm (AHA) is a newly established swarm intelligence algorithm by Zhao et al. [10]. AHA is inspired by the intelligent behaviors of the smallest-sized bird in the world called Hummingbirds. From a biological perspective, the hummingbird has considered the most intelligent animal since its body size is very small compared to its intelligence level. Therefore, this intelligent behavior is modeled as an optimization algorithm.

As a high-level description of the basic steps of AHA is given in Figure 1. Where the algorithm starts with the population and visit table initialization, determines the flight behavior, and chooses the foraging strategy. Lastly, as an iterative algorithm, the algorithm will check whether to terminate after each iteration.

1) STEP 1: POPULATION AND VISIT TABLE CONSTRUCTION

As with any other population-based algorithm, AHA c starts by developing its population. Which consists of n hummingbirds placed on n food sources. For hummingbirds' positions (x) initialization, Eq. (1) will be utilized where x_i refers to the position of the i^{th} food source assigned to the i^{th} hummingbird. In addition, the terms LB and UB are the lower and upper bounds of the optimization problem. Furthermore, to establish the initial random population, r was utilized, which refers to a random vector of numbers between 0 and 1. It is worth mentioning that LB, UB, r, and x_i are vectors of d dimension, according to the problem on hand.

$$x_i = LB + r \cdot (UB - LB) \qquad , i = 1, 2, \ldots, N \tag{1}$$

where $r \in [0, 1]$ is random value. LB and UB represent the limits of the search space. N is the size of the population.

In addition to population initialization, some of the hummingbirds' characteristics are their intelligence and strong memory. In nature, each hummingbird will be responsible for one food source, from which all information about that food source can be gained (e.g., nectar-refilling rate). Furthermore, it can move/migrate and get food from other sources. However, in terms of memorization, hummingbirds memorize two pieces of information. The first is related to the food source it is assigned to. Furthermore, the second relates to visiting all other food sources last time. Simulating this in AHA, the memories of the birds were presented using a visiting table, which represents the food sources visiting level. The higher the level, the longer the time since the last visit.

As illustrated in Table 1, the visit table contains n rows and n columns (here, n=4), where the rows represent the number of birds and the columns are the number of food sources. The table will be filled according to Eq. (2), where two cases exist. In the first case, $i = j$ refers to the hummingbird and its assigned food source. In this case, no visit level will be assigned, and the value of this cell will be null. On the other

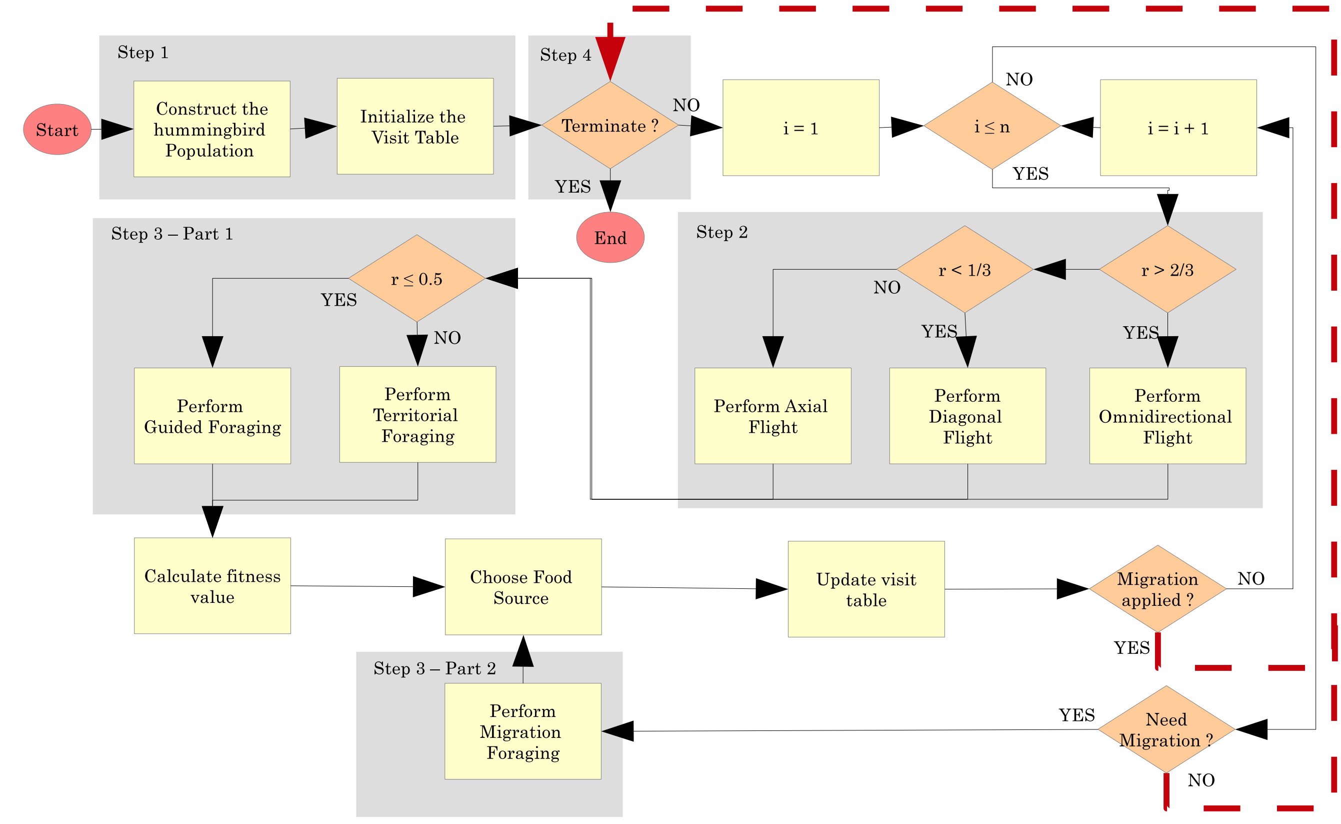


FIGURE 1. Artificial hummingbird algorithm flowchart.

hand, when $i \neq j$, all the food sources will be assigned to 0, indicating a recent visit (because we are still at the initial step). In other words, for each hummingbird, all the food sources level will be zero, except the food source it is responsible for will be null.

$$VT_{ij} = \begin{cases} null, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \qquad i = 1, 2, \ldots, n; j = 1, 2, \ldots, n \tag{2}$$

2) STEP 2: DETERMINE THE FLIGHT BEHAVIOUR

Generally, the flying behavior of the birds is known to be omnidirectional. However, for hummingbirds, three flying manners are conducted;

1) Axial flight,
2) Diagonal flight, and
3) Omnidirectional flight

Referring to Figure 2, the first flight type (A) is the Axial flight, where the birds fly along axes. Next (B) is the Diagonal flight; in this type, the birds fly from one corner of the rectangle to the opposite corner (assuming a rectangle is formed between two axes). Lastly, the Omnidirectional flight (C), in which all the axes drive the bird's movement.

That was the theory of the flight types. However, programming-wise, a random number r will be generated in the range of (0,1) and utilized for flight selection according to the conditional Eq. (3).

TABLE 1. Initial visit table.

		Food Sources			
		x_1	x_2	x_3	x_4
Hummingbird	x_1		0	0	0
	x_2	0		0	0
	x_3	0	0		0
	x_4	0	0	0	

$$Flighttype = \begin{cases} Diagonalflight, & \text{if } r < \frac{1}{3} \\ Omnidirectionalflight, & \text{if } r > \frac{2}{3} \\ Axialflight, & \text{Otherwise} \end{cases} \tag{3}$$

According to the chosen flight type, one of the Eqs. (4)-(6) will be executed. The aim is to define the

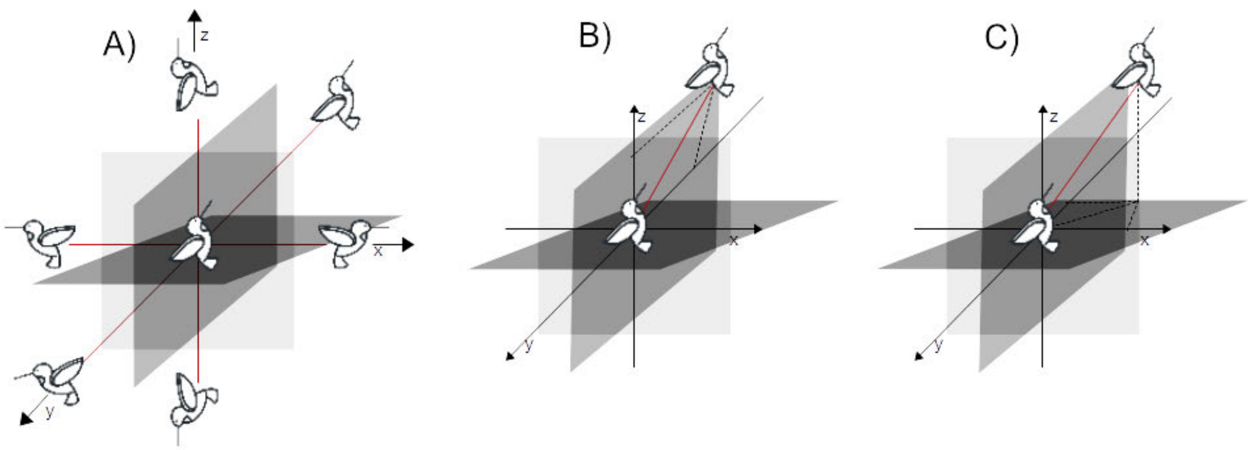


FIGURE 2. The three flying behaviours of hummingbirds.

direction switch vector D. This vector consists of ones and zeros, in which 1 indicates a move concerning the i^{th} dimension. With regards to the equations, and starting with the Axial flight direction generation Eq. (4), $randi([1, d])$ is used to a random number in the range of 1 and d will be generated. This number refers to the dimension that the hummingbird will fly along. Accordingly, the chosen dimension will be set to 1 and all others to 0.

$$D^{(i)} = \begin{cases} 1, & \text{if } i = randi([1, d]) \\ 0, & \text{Otherwise} \end{cases} \quad i = 1, \ldots, d \tag{4}$$

Next is the Diagonal flight Eq. (5), since the diagonal flight depends on more than one axes (more specifically, 2 to d-1 axes). Therefore, a permutation random vector generator function, "P," generates a distinct dimension. Notably, "P" was provided by "j", where "j" refers to the number of permutation axes. In which "j" is an integer between 1 and K, and "K" is defined to be $K \in [2, \lceil r_1 \cdot (d-2) \rceil + 1]$. Moreover, "$r_1$" in the K equation is a random number greater than 0 and less than or equal to 1.

$$D^{(i)} = \begin{cases} 1, & \text{if } i = P(j), j \in [1, k], P = randperm(k) \\ 0, & \text{Otherwise} \end{cases} \quad i = 1, \ldots, d \tag{5}$$

The third flight type, Omnidirectional, presents a movement for all dimensions. Therefore, the direction switch vector will be set to 1 for all dimensions.

$$D^{(i)} = 1, i = 1, \ldots, d \tag{6}$$

3) STEP 3: DETERMINE THE FORAGING BEHAVIOUR

According to the availability and the level of nectar-refilling, the hummingbirds utilize the previously mentioned flights to reach better food sources. The foraging of hummingbirds is of three types

1) Guided foraging,
2) Territorial foraging, and
3) Migration foraging.

For easier visualization, the three types of foraging are demonstrated in Figure 3. In which the flowers in the figure represent the hummingbirds' food sources. Furthermore, we will consider one bird for the subsequent sections' explanation. However, for implementation, all hummingbirds will follow the same procedures.

4) GUIDED FORAGING

The first foraging procedure is guided foraging; as the name states, it is guided in behaviors, which means the bird utilizes its predefined knowledge to guide it to the food source. The hummingbird knowledge and as mentioned earlier, is accumulated in its memory or the visit table as an AHA notation. Therefore, the bird will return to the visit table and pick to move to the food source with the highest level (the longest not visited food source). If two or more food sources have the same visit level, the decision will be according to their nectar-refilling rate, where the bird will visit the food source with the highest level and highest nectar-refilling rate. This foraging procedure was formulated as illustrated in Eq. (7). Where $v_i(t+1)$ is the position of the new food source for the next iteration $(t+1)$, $x_{i,tar}(t)$ and $x_i(t)$ are the target food source, and the current bird position at iteration t, respectively. In addition, D in the equation refers to the direction switch vector established by the flight type (Eq. (4)-(6)). Lastly, the guiding factor a is subjected to a normal distribution $N(0, 1)$.

$$v_i(t+1) = x_{i,tar}(t) + a \cdot D \cdot (x_i(t) - x_{i,tar}(t)) \tag{7}$$

5) TERRITORIAL FORAGING

After visiting the known food sources and eating their nectar, the bird will start searching for better food sources in their neighbor regions (Territorial). Thus, Eq. (8) will be utilized to allocate the new food source position $v_i(t+1)$ based on $x_i(t)$, which is the current bird position at iteration t,

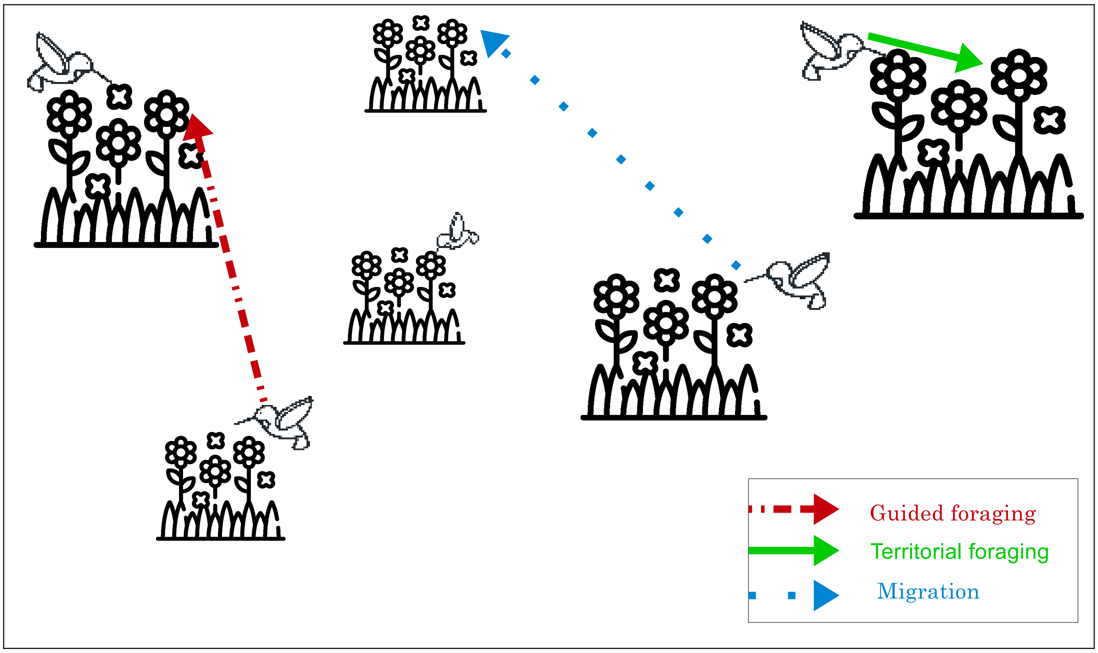


FIGURE 3. The three hummingbirds' foraging behaviours.

And the direction switch vector D. In addition, this foraging is controlled by a territorial factor b, formed as shown in Eq. (9) [10].

$$v_i(t+1) = x_i(t) + b \cdot D \cdot x_i(t) \tag{8}$$
$$b \sim N(0, 1) \tag{9}$$

For both the guided foraging as well as the territorial foraging, the hummingbird will not fly to $v_i(t+1)$ unless its nectar-refilling rate is better than what is in the current position $x_i(t)$ (food source standing on). The nectar-refilling rate is simulated as the objective function of the optimization problem $f(\cdot)$. Therefore, the position of the hummingbird at iteration $x_i(t+1)$ is decided according to Eq. (10). In which $f(x_i(t))$ is the nectar-refilling rate at the current hummingbird position, and $f(v_i(t+1))$ is the nectar-refilling rate for the new food source. Furthermore, programming-wise, the foraging behavior is selected according to a random number r ($r \in [0, 1]$), such that if $r < 0.5$, then the guided foraging will take place. Otherwise, the territorial foraging.

Moreover, after deciding the food source to fly to, the visit table must be updated. The chosen food source level in the table will be set to 1, and all other food sources' levels will be incremented by 1.

$$x_i(t+1) = \begin{cases} x_i(t), & f(x_i(t)) \leq f(v_i(t+1)) \\ v_i(t+1), & f(x_i(t)) > f(v_i(t+1)) \end{cases} \tag{10}$$

6) MIGRATION FORAGING

After investigating all possible known and neighbor food sources, the last alternative is to migrate. The migration of hummingbird decision occurs after discovering the lack of food. In which a migration coefficient M will be defined, and in case the number of iterations reaches M, the migration will occur (M suggested by the author to be 2n [10]). As a result, the hummingbird position will be updated to move to a random unknown area, which is the same as developing the hummingbird position at the initial population. Thus, the new hummingbird position $x_{wor}(t+1)$ is calculated using Eq. (11). The "wor" subscript in $x_{wor}(t+1)$ is used to indicate the worst situation happening to the bird, which causes it to migrate. One more time, after reaching the food source, the visit table will be updated.

$$x_{wor}(t+1) = LB + r \cdot (UB - LB) \tag{11}$$

7) STEP 4: TERMINATION

Generally speaking, different stopping criteria can be utilized to terminate the iterative process of the optimization algorithm. Such as [33]

1) defining the maximum number of iterations,
2) defining a target objective value, etc

In the case of AHA, we utilized the maximum number of iterations *Max_iterations* as a stopping criterion. In steps 2 & 3 will be iteratively repeated until the number of iterations reaches *Max_iterations*.

B. QUANTUM-BASED OPTIMIZATION

This section introduces the basic information for quantum-based optimization (QBO). In QBO, the binary number represents the features that will be selected or eliminated, corresponding to 1 or 0, respectively. Each feature in QBO is represented by a quantum bit (Q-bit (q)), where q denotes the superposition of binary values (i.e., '1' and '0') and is

TABLE 2. The value of $\Delta\theta$ [34].

X_{ij}	X_{bj}	$f(X_{ij}) \geq f(X_{bj})$	$\Delta\theta$
0	0	False	0
0	1	False	0.01 π
1	0	False	-0.01π
1	1	False	0
0	0	True	0
0	1	True	0
1	0	True	0
1	1	True	0

represented by a complex number as:

$$q = \alpha + i\beta = e^{i\theta}, \; |\alpha|^2 + |\beta|^2 = 1 \tag{12}$$

where the possibility of the value of the Q-bit being '0' and '1' is given by α and β, respectively. The parameter θ denotes the angle of q and it is updated using $arctan(\alpha/\beta)$.

The process of finding the change in the value of q is the main objective of QBO, and it is determined by calculating $\Delta\theta$ as:

$$q(t+1) = q(t) \times R(\Delta\theta) = [\alpha(t)\beta(t)] \times R(\Delta\theta) \tag{13}$$

where $R(\Delta\theta)$ stands for the rotational matrix associated with a change of $\Delta\theta$ in the quantum angle, and it is defined as:

$$R(\Delta\theta) = \begin{bmatrix} cos(\Delta\theta) & -sin(\Delta\theta) \\ sin(\Delta\theta) & cos(\Delta\theta) \end{bmatrix} \tag{14}$$

Following [34], the value of $\Delta\theta$ is predefined according to best solution (i.e., X_b) and it is given as in Table 2. In this table, X_{ij} stands for the bit j of the binary solution of X_i. Whereas, X_{bj} refers to the jth bit of X_b at iteration t. In addition, the angle vector has eight values according to the experimental test conducted on [35].

III. PROPOSED METHOD

The fundamental purpose of using QBO is to enhance the ability to strike a better balance between exploration and exploitation while looking for a feasible solution. The developed FS method, QAHA, begins by dividing the data into training and testing sets of 70% and 30%, respectively. The fitness values for each individual are then computed using random values. The individual with the smallest fitness value is then allocated as the best agent. After that, employing AHA exploitation, the solution is updated. The updating of an individual is conducted again until the stop criteria are reached. Following that, based on the best solution, the dimension of the testing set is reduced, and the implemented QAHA as FS is assessed using multiple measures (as in Figure 4). The following sections go through the QAHA in detail.

A. FIRST STAGE

The initial agents representing the population are created at this stage. Each solution contains D Q-bits (D is the number of features). Therefore, the solution X_i can be formulated as in Eq. (15).

$$X_i = [q_{i1}\,|q_{i2}|\ldots|\,q_{iD}] = [\theta_{i1}\,|\theta_{i2}|\ldots|\theta_{iD}],\, i = 1, 2, \ldots, N \tag{15}$$

In this equation, X_i refers to a set of superpositions of probabilities of those features that are either selected or not.

B. SECOND STAGE

This part of the QAHA primarily aims to update the agents until they reach the stop criteria. This is accomplished via a series of steps, the first of which is to obtain the binary of each X_i using the equation:

$$BX_{ij} = \begin{cases} 1 & if\ |\beta|^2 > \ rand \\ 0 & otherwise \end{cases} \tag{16}$$

where rand$\in$[0, 1] is random value and β is represented in Eq. (12). The next step is to train the KNN classifier with five neighbors as the model hyper-parameters using the training features that correspond to the ones in BX_{ij} and compute the fitness value, which is defined as:

$$Fit_i = \rho \times \gamma + (1-\rho) \times \left(\frac{|BX_{ij}|}{D}\right) \tag{17}$$

In Eq. (17), $|BX_{ij}|$ denotes the total number of selected features, and γ is the error classification using the KNN classifier (i.e., relevant features). $\rho\in[0, 1]$ is the factor that equalizes the fitness value of two sections. The key reason for choosing KNN is that it is simple, efficient, and has one parameter. Furthermore, it has outperformed the majority of other classifiers in various applications because it saves the data from the training set.

The step after that is to find the best agent X_b with the smallest Fit_b. Then using the operators of AHA as discussed in Eqs. (2)-(11).

C. THIRD STAGE

The testing set is reduced at this stage by only selecting features corresponding to those in the binary version of X_b. The trained classifier (KNN) is then applied to the decreased dimension of the testing set and predicts the output of the testing set. The next step is to assess the output's quality using various indicators. The steps of the QAHA algorithm are described in Algorithm 1.

The computational complexity of the QAHA depends on the initial population, size of population N, the fitness evaluation (N_{Fit}), and t_{max} maximum number of iterations.

$$\begin{aligned} O(QAHA) = O(T \times C \times N + T \times N \\ \times D + T \times D/2) + O(N \times D) \end{aligned} \tag{18}$$

Therefore, the complexity of QAHA is given as:

$$\begin{aligned} O(QAHA) = O(T \times N_{Fit} \\ \times N + T \times N \times D + T \times D/2) \end{aligned} \tag{19}$$

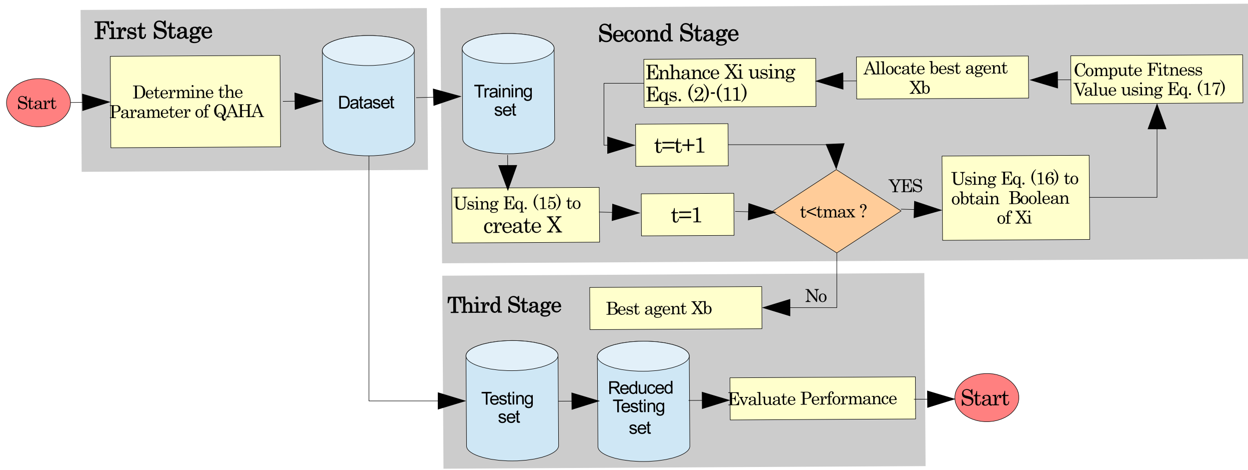


FIGURE 4. Steps of QAHA as FS.

Algorithm 1 The QAHA Method

1: Input: a dataset with D features, as well as population size (N), iterations (*tmax*), and QAHA parameters.
2: **First Stage**
3: Divide the data into two sets (i.e., testing and training)
4: Using Eq. (15), create the population X. **Second Stage**
5: Set $t = 1$
6:**while** ($t < tmax$) **do**
7: Using Eq. (16) to obtain the Boolean form of X_i.
8: Compute fitness value of X_i using training set as in Eq. (17).
9: Allocate the best agent X_b.
10: Enhance X using Eqs.(2)-(11).
11: $t = t + 1$
Third Stage
12: Reduce the testing set according to selected features using X_b.
13: Evaluate the quality using different metrics.

IV. EXPERIMENTAL RESULTS

The Results Section discusses the performance of eight different feature selection optimization techniques, including AHA, LSHDE [36], whale optimization algorithm (WOA) [37], Self-adaptive differential evolution (SaDE) [38], Teaching Learning Based Optimization (TLBO) [39], grey-wolf optimization algorithm (GWO) [40], L-SHADE with Semi Parameter Adaptation (LSPACMA) [41], and Genetic algorithm (GA) [42]. These popular techniques are compared with the proposed QAHA algorithm. In addition, the parameter of each algorithm is used as in the original work. The common the parameters such as the size of the population and t_{max} is 20 and 50, respectively. Each FS algorithm was conducted 25 times for fair comparison and statistical analysis. To measure the superiority of the proposed algorithm, the performance of these algorithms is compared using two experimental series. The details of each one are given in the following subsections.

A. EXPERIMENTS BASED ON UCI DATASETS

In this experiment, we study the efficiency of the developed QAHA in tackling the benchmark of UCI datasets.

1) UCI DATASETS DESCRIPTION

In this experiment, different UCI datasets are used to assess QAHA. To validate the developed QAHA, eighteen UCI datasets [43] are used. Table 3 contains the descriptions of these datasets. These datasets were gathered from various fields with varying features and instances, as shown in Table 3.

In addition, the efficiency of the developed and other competitive methods is evaluated using performance metrics. For example, Accuracy, Sensitivity, Specificity, and Fitness value. These measures are defined in Eqs. (20)-(23).

- Average of the Fit_b value is computed as:

$$Average\ of\ \ Fit_b = \frac{1}{N_r}\sum_{i=1}^{N_r} Fit_b^i \tag{20}$$

where N_r refers to number of runs and Fit_b stands for the best fitness value at the i^{th} run.

- Accuracy measure is defined as:

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{21}$$

- Sensitivity is formulated as:

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{22}$$

- Specificity is defined as:

$$Specificity = \frac{TN}{(TN + FP)} \tag{23}$$

where T_P, T_N, F_P, and F_N stand for true positives, true negatives, false positives, and false negatives, respectively.

TABLE 3. Description of UCI datasets.

Set	Number of features	Number of instances	Number of classes	Data category
Breastcancer (S1)	9	699	2	Biology
BreastEW(S2)	30	569	2	Biology
CongressEW(S3)	16	435	2	Politics
Exactly(S4)	13	1000	2	Biology
Exactly2(S5)	13	1000	2	Biology
HeartEW(S6)	13	270	2	Biology
IonosphereEW(S7)	34	351	2	Electromagnetic
KrvskpEW(S8)	36	3196	2	Game
Lymphography(S9)	18	148	2	Biology
M-of-n(S10)	13	1000	2	Biology
PenglungEW(S11)	325	73	2	Biology
SonarEW(S12)	60	208	2	Biology
SpectEW(S13)	22	267	2	Biology
Tic-tac-toc(S14)	9	958	2	Game
Vote(S15)	16	300	2	Politics
WaveformEW(S16)	40	5000	3	Physics
WINEEW(S17)	13	178	3	Chemistry
Zoo(S18)	16	101	6	Artificial

2) RESULTS AND DISCUSSION

Table 4 shows the accuracy metric of the different FS optimizers measured on the 18 datasets. As can be noticed, QAHA achieves the best results with 10 out of 18 datasets. It achieves 100% with five datasets, i.e., S1, S4, S11, S17, and S18. QAHA achieved accuracies of 0.9522, 0.8889, 0.9714, 0.9117, 0.9985, 0.8306, 0.8122, and 0.7650 for the S2, S6, S7, S9, S10, S13, S14, and S16 datasets, respectively. HA has yet to achieve the best results with any 18 datasets compared to other optimizers. These results show the role of quantum in significantly improving the performance of the AHA algorithm (P-value <0.0001). SHADE, WOA, SADE, and BGWO have yet to achieve the best results with any datasets compared to other FS techniques. On the other hand, TLBO achieves the best results with seven datasets. However, there is no significant difference between the proposed QAHA and the TLBO regarding these seven results. For example, both algorithms achieve 100% accuracy with the S18, S4, and S17 datasets. With the S2 dataset, TLBO achieved enhanced accuracy by 0.0051 than QAHA. For the S7 dataset, TLBO beats QAHA by 0.0145 accuracies. Regarding the S9 and S10, TLBO has better accuracies by 0.022 and 0.0005, respectively. LSPACMA and GA achieved the best results in one (S14, accuracy = 0.8750) and two (S13, 0.8580; S18, 1.0) datasets, respectively. To conclude, the proposed QAHA optimizer significantly achieved better results than the other eight algorithms (P-value < 0.01), as shown in Figure 5.

Table 5 shows the sensitivity metric of the different FS optimizers measured on the 18 datasets. These results are consistent with the accuracy metric in Table 4. The proposed QAHA algorithm achieves the best sensitivity with 12 datasets compared to the other optimizers. Both QAHA and AHA algorithms achieve 100% sensitivity with the S11 dataset. Note that AHA achieved high results with this dataset only. Extending AHA with the quantum causes the QAHA to achieve significantly improved results (P-Value <0.0001). In addition, the proposed QAHA algorithm performs significantly better than the other seven optimizers (P-value < 0.01). For example, LSHADE beats QAHA with two datasets, i.e., S14 and S18. WOA beats the proposed algorithm with only one dataset, S18. SADE achieved better results than QAHA with the S16 dataset only. TLBO achieves the best results with six datasets, but TLBO and QAHA achieve the same results with three datasets, i.e., S10, S11, and S17. TLBO has not achieved significantly better results than QAHA with the other datasets. For example, with the S4 dataset, TLBO and QAHA achieve a sensitivity of 1.0 and 0.9978, respectively. BGWO and LSPACMA achieve better sensitivities than QAHA with the S18 dataset alone. Finally, GA beats QAHA with only two datasets of, S18 and S7. These results are summarized in Figure 6.

Table 6 compares the specificity for the tested optimizers. The proposed QAHA optimizer achieves the best specificity with all datasets except S4, S6, S7, S14, and S18. QAHA has statistically significantly improved the performance of AHA (P-value<0.001), which achieves the best specificity with the S11 dataset. Regarding the S1, S8, S13, S9, S12, and S16 datasets, the QAHA optimizer achieves the highest specificity of 0.9965, 0.9740, 0.8164, 0.9882, 0.9921, and 0.7747, respectively. For S2 and S3, again, QAHA achieves the highest specificity scores of 0.9824 and 0.9818, respectively. TLBO algorithm achieves the best results with the S4 dataset (specificity of 1.0). QAHA, lshade, SaDE, BGWO, and LSPACMA get the highest specificity of 1.0 with the S5 dataset. TLBO scores a specificity of 0.9024 with the S6 dataset. For S7, both SADE and GA achieve the best results. Both QAHA and TLBO achieve the best specificity of 1.0 with the dataset of S10. All optimizers achieve a specificity of 1.0 with the S11 dataset except LSHADE and SADE. The LSHADE algorithm achieves a specificity of 0.9646 with the S14 dataset. Regarding the S17 dataset,

TABLE 4. Accuracy of FS algorithms using testing set.

	AHA	QAHA	lshade	WOA	SaDE	TLBO	bGWO	LSPACMA	SGA
S1	0.8839	**1.0000**	0.9286	0.9476	0.9643	0.9729	0.9567	0.9429	0.9462
S2	0.9096	0.9522	0.8684	0.9433	0.9123	**0.9573**	0.9415	0.9035	0.9351
S3	0.9368	**0.9770**	0.9540	0.9448	0.9195	0.9655	0.9157	0.9655	0.9609
S4	0.6523	**1.0000**	0.7375	0.8960	0.6400	**1.0000**	0.8987	0.6700	0.8667
S5	0.6478	**0.7950**	0.7250	0.7703	0.7450	0.7507	0.7900	0.7300	0.7130
S6	0.7454	0.8889	0.7593	0.7988	0.7500	0.8877	0.8272	0.7593	0.8753
S7	0.8704	0.9714	0.9296	0.9174	0.9789	**0.9859**	0.9380	0.9437	0.9577
S8	0.6891	**0.9698**	0.5852	0.9507	0.5250	0.9547	0.9577	0.5594	0.9616
S9	0.7341	0.9117	0.8000	0.8821	0.8073	**0.9337**	0.8756	0.8333	0.8844
S10	0.6723	0.9985	0.7450	0.9497	0.7325	**0.9990**	0.9627	0.7100	0.9477
S11	0.8462	**1.0000**	0.7333	0.9686	0.6667	0.9511	0.9822	0.8846	0.8667
S12	0.8738	**0.9940**	0.6905	0.9825	0.6905	0.9794	0.9381	0.5357	0.9937
S13	0.7093	0.8306	0.8148	0.7593	0.7500	0.8531	0.7716	0.8519	**0.8580**
S14	0.6302	0.8122	0.7188	0.7809	0.7630	0.8354	0.7819	**0.8750**	0.8250
S15	0.8650	**0.9767**	0.9667	0.9622	0.9500	0.9711	0.9700	0.9083	0.9600
S16	0.6387	0.7650	0.5370	0.7283	0.5580	0.7530	0.7186	0.5170	0.7533
S17	0.8931	**1.0000**	0.8333	0.9759	0.9167	**1.0000**	0.9833	0.9861	0.9833
S18	0.9713	**1.0000**	0.6667	0.9968	**1.0000**	**1.0000**	0.9841	0.8333	**1.0000**
AVG	0.78±0.12	**0.93**±0.08	0.77±0.12	0.89±0.09	0.79±0.15	0.93±0.08	0.89±0.08	0.80±0.15	0.90±0.08

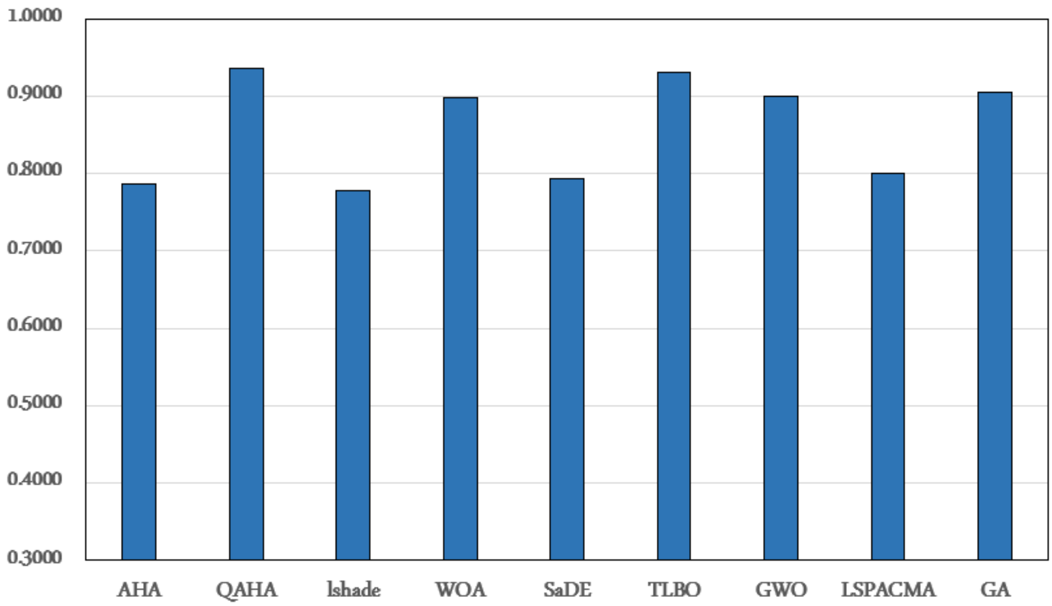


FIGURE 5. Comparison of average accuracy of different FS optimizers.

both QAHA, lshade, and WOA achieve a specificity of 1.0. QAHA, SaDE, TLBO, LSPACMA, and GA achieve a specificity of 1.0. For the S15 dataset, QAHA lshade WOA achieves a specificity of 1.0. All optimizers achieve a specificity of 1.0 with the S18 dataset except AHA, QAHA, and SADE. Figure 7 illustrates the specificity of different optimizers. As can be noticed, the proposed QAHA optimizer achieves the best average specificity over the 18 datasets, following the TLBO algorithm with a small difference.

To measure how the optimization algorithms balance the number of selected features and the model performance, we analyze the behavior of the fit function for all optimizers. Please note that the lowest values are preferred because we aim to minimize the objective function. Table 7 shows the average fitness values for all optimizers with the 18 datasets. Table 8 shows the fitness values' variations to show the fitness values' stability. Table 9 shows the best fitness values for each optimizer. Table 10 shows the worst fitness values for all optimizers. These results are collected from 30 runs of the models. In Table 7, we notice that the proposed QAHA optimizer achieves the lowest average fitness values for 15 datasets and does not achieve the lowest average with S9, S13, and S14. There is a significant difference between the results of QAHA and other optimizers (P-value <0.01). AHA achieves the lowest average fitness with the S9 dataset. Ishade, SaDE, and TLBO optimizers do not achieve the best

TABLE 5. Sensitivity of FS algorithms.

	AHA	QAHA	lshade	WOA	SaDE	TLBO	BGWO	LSPACMA	SGA
S1	0.9042	**0.9965**	0.9792	0.9420	0.9674	0.9773	0.9747	0.9770	0.9709
S2	0.9541	**0.9824**	0.9306	0.9634	0.9638	0.9787	0.9479	0.9225	0.9771
S3	0.9356	**0.9818**	0.9362	0.9450	0.9310	0.9815	0.9155	0.9623	0.9490
S4	0.8109	0.9978	0.9470	0.9372	0.6761	**1.0000**	0.9642	0.8042	0.9012
S5	0.7314	**1.0000**	**1.0000**	0.9862	**1.0000**	0.9933	**1.0000**	**1.0000**	0.8393
S6	0.7655	0.8983	0.6552	0.8938	0.7609	**0.9024**	0.8076	0.7833	0.8923
S7	0.9804	0.9915	0.9767	0.9829	**1.0000**	0.9970	0.9841	0.9800	**1.0000**
S8	0.6899	**0.9740**	0.5269	0.9642	0.9040	0.9622	0.9625	0.9096	0.9613
S9	0.7313	**0.9882**	0.8438	0.7897	0.8750	0.8704	0.8706	0.4444	0.9375
S10	0.6042	**1.0000**	0.8049	0.9289	0.4359	**1.0000**	0.9559	0.8333	0.9093
S11	**1.0000**	**1.0000**	0.6190	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
S12	0.8417	**0.9921**	0.7895	0.9792	0.6458	0.9795	0.9014	0.8434	0.9900
S13	0.1900	**0.8164**	**1.0000**	0.0762	0.3333	0.6952	0.2278	**1.0000**	0.7926
S14	0.7756	0.8915	**0.9646**	0.8407	0.8654	0.8767	0.8249	0.8833	0.8780
S15	0.9658	**1.0000**	**1.0000**	**1.0000**	0.9167	0.9704	0.9800	0.9524	0.9872
S16	0.5996	**0.7747**	0.6667	0.7025	0.7000	0.7221	0.6762	0.5841	0.7215
S17	0.9393	**1.0000**	0.9091	0.9704	**1.0000**	**1.0000**	0.9958	**1.0000**	**1.0000**
S18	0.9773	0.9308	**1.0000**	**1.0000**	0.9615	**1.0000**	**1.0000**	**1.0000**	**1.0000**
AVG	0.79±0.20	**0.95**±0.07	0.73±0.36	0.84±0.29	0.77±0.27	0.93±0.09	0.88±0.19	0.78±0.32	0.92±0.08

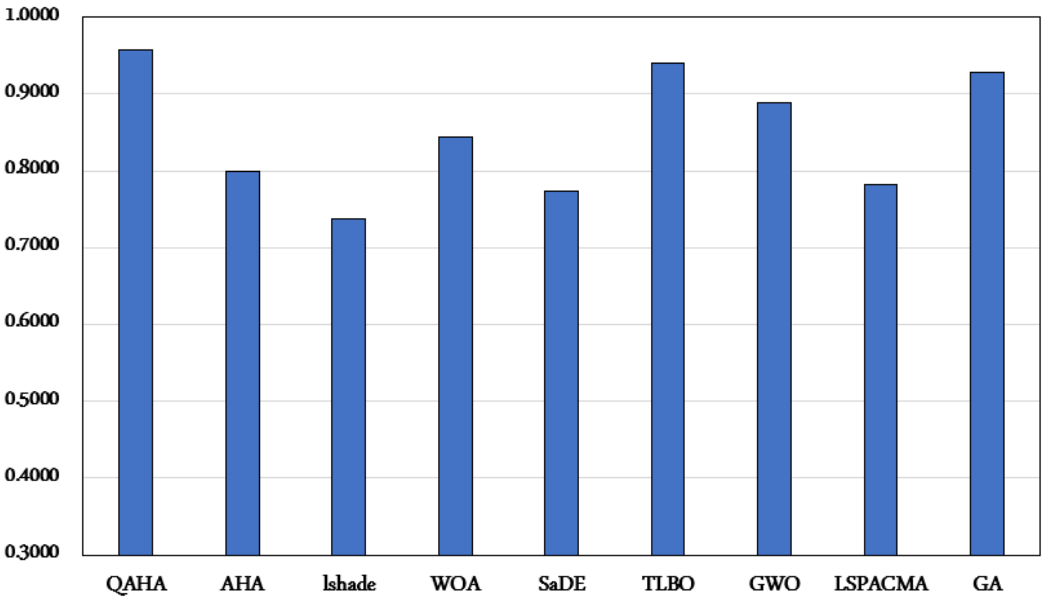


FIGURE 6. Comparison of average sensitivity of different FS optimizers.

results with any dataset. WOA optimizer has the best results with the two datasets of S11 and S15. GWO optimizer has the best result with the one dataset of S5. LSPACMA optimizer achieves the best result with S13 and GA optimizer with S14.

In Table 8, QAHA achieves 0 STD with only one dataset (S5). On the other hand, SaDE achieves the lowest STD with four datasets (S1, S4, S8, and S16), and LSPACMA achieves the lowest STD with seven datasets (i.e., S2, S3, S4, S9, S13, S14, and S18). However, as shown in Figure 8, the proposed QAHA achieves the lowest average STD of fitness value compared to the other optimizers. AHA, QAHA, lshade, WOA, SaDE, TLBO, GWO, LSPACMA, and GA achieve an average STD of 0.01319, 0.00924, 0.01879, 0.02477, 0.01278, 0.01845, 0.02148, 0.02126, and 0.01846, respectively. We conclude from these results that QAHA is the most stable optimizer because it achieves the lowest average STD with the 18 datasets. Table 9 and Table 10 show the upper and lower fitness values for each optimizer with the 18 datasets. QAHA achieved the best results with seven datasets S1 (0.0333), S3 (0.0269), S5 (0.1922), S7 (0.0548), S15 (0.0338), S16 (0.2654), and S17(0.0385). AHA, Ishade, SaDE, and GA did not achieve the best results with any datasets. WOA optimizer has the best results with four datasets of S2 (0.0491), S4 (0.0462), S9 (0.0471), and S11 (0.0031). TLBO optimizer has the best performance with S14 (0.2120); GWO optimizer with S4 (0.0462), S15 (0.0338), and S17 (0.0385); LSPACMA with one dataset of S13 (0.1370). However, as shown in Figure 8, the proposed QAHA achieves the lowest average value for the best fitness compared to the other optimizers. AHA, QAHA, lshade, WOA, SaDE, TLBO, GWO, LSPACMA, and GA achieve an average value for best fitness of 0.0988, 0.0882, 0.2129,

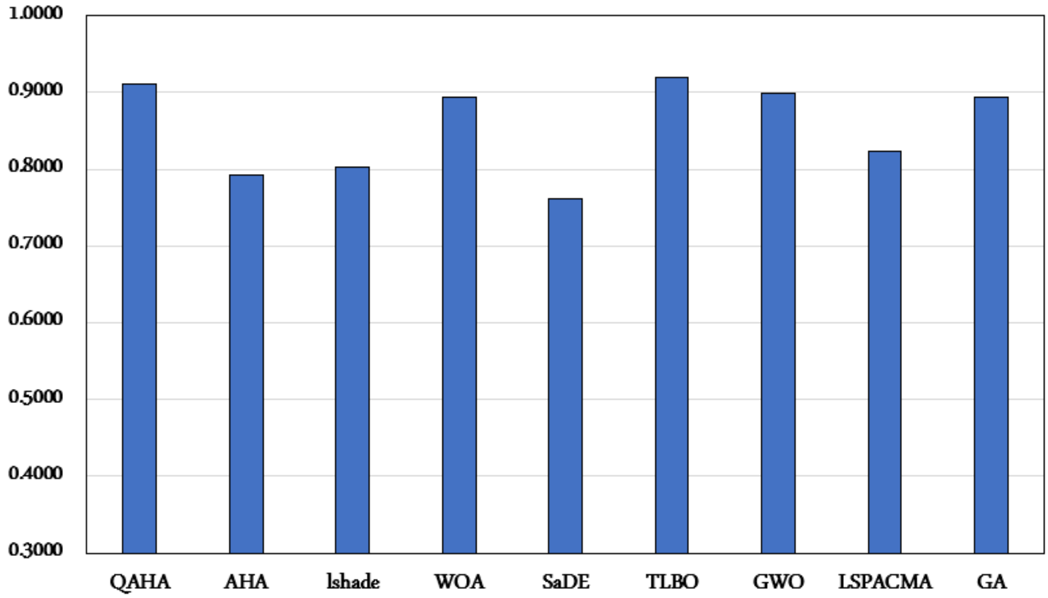


FIGURE 7. Comparison of average specificity of different FS optimizers.

TABLE 6. Specificity of FS methods.

	AHA	QAHA	lshade_iter	WOA_FS	SaDE	TLBO	bGWO	LSPACMA	SGA
S1	0.8536	**0.9938**	0.8182	0.9564	0.9583	0.9638	0.9185	0.8868	0.9057
S2	0.8305	**0.9318**	0.7619	0.9192	0.8333	0.9244	0.9278	0.8721	0.8752
S3	0.9386	0.9688	0.9750	0.9444	0.8966	0.9394	0.9161	0.9706	**0.9778**
S4	0.7014	0.9926	0.9184	0.8022	0.7517	**1.0000**	0.7072	0.8333	0.7949
S5	0.7232	**0.8429**	0.7462	0.7567	0.7434	0.6667	0.8340	0.7854	0.7160
S6	0.7220	0.8521	**0.8800**	0.6606	0.7419	0.8718	0.8632	0.7292	0.8595
S7	0.6680	0.8729	0.8571	0.8375	0.9500	**0.9679**	0.8533	0.8571	0.9063
S8	0.6881	**0.9657**	0.6487	0.9358	0.1388	0.9462	0.9521	0.1549	0.9619
S9	0.7893	0.8308	0.7857	0.9149	0.6429	0.8667	0.8205	**1.0000**	0.8810
S10	0.7097	0.9976	0.7034	0.9621	0.9221	**0.9984**	0.9664	0.6406	0.9693
S11	0.9962	**1.0000**	0.9692	0.9714	0.9231	0.9944	0.9952	0.8571	**1.0000**
S12	0.8979	0.9957	0.6087	0.9846	0.7500	0.9793	0.9825	0.9167	**0.9970**
S13	0.8273	0.8600	**1.0000**	0.9983	0.9103	0.9083	0.9270	**1.0000**	0.8711
S14	0.3932	0.6858	0.2385	0.6667	0.5484	0.7424	0.7053	**0.8611**	0.7284
S15	0.8183	0.9622	0.9375	0.9433	**0.9722**	0.9714	0.9650	0.8846	0.9392
S16	0.7840	0.8715	**1.0000**	0.8533	0.7000	0.8625	0.8539	0.5605	0.8650
S17	0.9636	**1.0000**	0.9200	0.9827	0.9444	**1.0000**	0.9933	**1.0000**	0.9719
S18	0.9600	0.7688	0.6667	0.9952	0.3750	0.9524	**1.0000**	**1.0000**	0.8667
Avg	0.79±0.14	**0.91**±0.092	0.80±0.18	0.89±0.11	0.76±0.22	**0.91**±0.09	0.89±0.09	0.82±0.21	0.89±0.08

0.0967, 0.2203, 0.1298, 0.0998, 0.2163, and 0.1298. On the other hand, for the average of the worst fitness values shown in Table 10, our optimizer achieves the lowest values, which again means that our optimizer is much more stable than other optimizers. QAHA achieves the best values in Table 10 with 11 datasets, i.e., S1 (0.0462), S3 (0.0457), S4 (0.1187), S5 (0.1922), S10 (0.0705), S11 (0.0135), S12 (0.0700), S14 (0.2418), S16 (0.2998), S17 (0.0615), and S18 (0.0250). Compared to QAHA, the AHA optimizer does not achieve the best results with any dataset. SaDE and GA have no best results with any datasets. Other optimizers, including Ishade, WOA, TLBO, GWO, and LSPACMA, achieve the best performance with two (S6 and S15), one (S2), one (S11), One (S5), and one (S13), respectively. QAHA achieves the lowest average for the worst fitness values over the 18 datasets. AHA, QAHA, lshade, WOA, SaDE, TLBO, GWO, LSPACMA, and GA achieve an average value for worst fitness of 0.1493, 0.1200, 0.2191, 0.1718, 0.2384, 0.1924, 0.1666, 0.2463, and 0.1916, respectively. Figure 8 summarizes the average fitness values for all optimizers. The figure highlighted the significant difference in performance between QAHA and other optimizers (P-value < 0.05).

In addition, we use the Friedman test as a non-parametric test to check the difference between the results of the developed method and others is significant or not. Table 11 shows the mean rank of the results obtained using the

TABLE 7. Average of fitness value of FS methods.

Mean	AHA	QAHA	lshade	WOA	SaDE	TLBO	GWO	LSPACMA	GA
S1	0.0797	**0.0372**	0.0833	0.0738	0.0917	0.0925	0.0679	0.1067	0.1018
S2	0.0885	**0.0684**	0.1254	0.0699	0.1114	0.0941	0.0809	0.1342	0.1280
S3	0.0582	**0.0310**	0.0655	0.0738	0.1063	0.0902	0.1075	0.0575	0.1018
S4	0.0684	**0.0572**	0.2504	0.1598	0.3853	0.2588	0.1415	0.2977	0.1923
S5	0.2576	**0.1922**	0.2258	0.2170	0.2285	0.3158	**0.1998**	0.2223	0.3306
S6	0.1845	**0.1842**	0.2019	0.2160	0.2417	0.2478	0.2038	0.2519	0.1958
S7	0.0873	**0.0796**	0.1160	0.0993	0.1148	0.1573	0.0817	0.1561	0.1206
S8	0.0925	**0.0836**	0.3904	0.0971	0.3658	0.1132	0.0955	0.3584	0.1148
S9	**0.1195**	0.1395	0.2567	0.1287	0.2516	0.1889	0.1564	0.2167	0.1818
S10	0.0607	**0.0529**	0.2118	0.1176	0.2706	0.1998	0.0998	0.3197	0.1179
S11	0.1468	**0.0090**	0.3200	**0.0408**	0.3500	0.0418	0.0489	0.2474	0.2022
S12	0.0704	**0.0490**	0.2833	0.0673	0.3333	0.1471	0.0965	0.3917	0.0890
S13	0.1845	0.1968	0.1630	0.2336	0.2417	0.2027	0.2353	**0.1370**	0.2047
S14	0.2496	0.2351	0.2635	0.2572	0.2992	0.2659	0.2548	0.3208	**0.2279**
S15	0.0820	**0.0404**	0.0567	**0.0457**	0.0850	0.0881	0.0533	0.1142	0.1052
S16	0.2871	**0.2848**	0.3574	0.2996	0.4094	0.3137	0.3026	0.4381	0.3075
S17	0.0536	**0.0492**	0.1833	0.0699	0.1583	0.0956	0.0571	0.1597	0.0878
S18	0.0334	**0.0194**	0.3333	0.0533	0.0833	0.0515	0.0660	0.2333	0.0563
AVG	0.12±0.08	0.10±0.08	0.22±0.10	0.13±0.08	0.23±0.11	0.16±0.09	0.13±0.08	0.23±0.11	0.16±0.08

TABLE 8. STD of fitness value of FS methods.

STD	AHA	QAHA	lshade	WOA	SaDE	TLBO	GWO	LSPACMA	GA
S1	0.0066	0.0055	0.0075	0.0113	**0.0000**	0.0120	0.0061	0.0141	0.0120
S2	0.0083	0.0113	0.0068	0.0115	0.0236	0.0067	0.0114	**0.0000**	0.0067
S3	0.0091	0.0062	0.0089	0.0153	0.0236	0.0119	0.0196	**0.0000**	0.0119
S4	0.0226	0.0159	0.0489	0.0963	**0.0000**	0.0823	0.0810	**0.0000**	0.0823
S5	0.0142	**0.0000**	0.0153	0.0262	0.0236	0.0164	0.0049	0.0000	0.0164
S6	0.0204	0.0139	0.0234	0.0225	0.0092	0.0105	0.0319	0.0340	0.0105
S7	0.0128	0.0159	0.0172	0.0177	0.0070	0.0091	0.0094	0.0096	0.0091
S8	0.0083	0.0062	0.0116	0.0150	**0.0000**	0.0106	0.0141	0.0468	0.0106
S9	0.0252	0.0174	0.0482	0.0597	0.0096	0.0271	0.0255	**0.0000**	0.0271
S10	0.0156	0.0066	0.0529	0.0410	0.0211	0.0425	0.0416	0.0087	0.0425
S11	0.0181	0.0028	0.0018	0.0314	0.0236	0.0023	0.0266	0.0852	0.0023
S12	0.0126	0.0128	0.0140	0.0128	0.0000	0.0106	0.0202	0.0825	0.0106
S13	0.0158	0.0147	0.0123	0.0072	0.0196	0.0119	0.0177	**0.0000**	0.0119
S14	0.0094	0.0070	0.0209	0.0180	0.0026	0.0231	0.0177	**0.0000**	0.0231
S15	0.0126	0.0080	0.0122	0.0168	0.0236	0.0232	0.0201	0.0318	0.0232
S16	0.0118	0.0088	0.0090	0.0174	**0.0000**	0.0085	0.0115	0.0129	0.0085
S17	0.0086	0.0080	0.0146	0.0139	0.0196	0.0161	0.0117	0.0570	0.0161
S18	0.0051	0.0053	0.0127	0.0119	0.0236	0.0075	0.0157	**0.0000**	0.0075
AVG	0.01±0.01	0.01±0.00	0.02±0.01	0.02±0.02	0.01±0.01	0.02±0.02	0.02±0.02	0.02±0.03	0.02±0.02

Friedman test for the Sensitivity, accuracy, Specificity, and fitness value measures. From these results, it can be noticed that P-value (last column) is less than 0.05, and this indicates the difference is significant. Whereas the mean rank of the developed method is higher according to the Sensitivity, accuracy, and Specificity, as well as the smallest according to the average, STD, Best, and worst fitness value (Fit). This means the proposed QAHA can provide better prediction performance than other methods to handle UCI datasets.

B. EXPERIMENTS BASED ON SOCIAL IoT DATASETS

Within this experiment, we study the efficiency of the developed QAHA in tackling the Social IoT datasets.

1) DATASETS DESCRIPTION

We consider the following four datasets described in this section to evaluate and validate the proposed QAHA algorithm. The datasets are GPS trajectories, hepatitis, GAS sensors (home activity monitoring), and MovementAAL (Indoor User Movement Prediction from RSS) [44]. Each dataset consists of features and a set of labels where all datasets are tackled as binary and multi-class classification. The description of the datasets is given below, while Table 12 summarizes the number of features and classes in each dataset. In our experiment, we split each dataset into training and testing sets with a ratio of 77% and 33%, respectively, in case of the absence of a predefined split.

TABLE 9. Best fitness of FS methods.

Best	AHA	QAHA	lshade	WOA	**SaDE**	TLBO	GWO	LSPACMA	GA
S1	0.0719	**0.0333**	0.0833	0.0590	0.0917	0.0830	0.0573	0.0967	0.0830
S2	0.0716	0.0528	0.1254	**0.0491**	0.0947	0.1107	0.0607	0.1342	0.1107
S3	0.0478	**0.0269**	0.0655	0.0560	0.0897	0.0769	0.0664	0.0575	0.0769
S4	0.0462	0.0462	0.2048	**0.0462**	0.3853	0.0615	**0.0462**	0.2977	0.0615
S5	0.2434	**0.1922**	0.2258	0.2102	0.2118	0.3032	0.1967	0.2223	0.3032
S6	0.1538	0.1705	0.2019	0.1731	0.2352	0.1692	0.1628	0.2278	0.1692
S7	0.0695	**0.0548**	0.1160	0.0742	0.1099	0.1048	0.0674	0.1493	0.1048
S8	0.0725	0.0712	0.3866	**0.0660**	0.3658	0.1003	0.0683	0.3254	0.1003
S9	0.0611	0.1156	0.2500	**0.0471**	0.2448	0.1322	0.1065	0.2167	0.1322
S10	0.0462	0.0462	0.2118	0.0615	0.2557	0.0615	0.0538	0.3135	0.0615
S11	0.0763	0.0037	0.3200	**0.0031**	0.3333	0.1982	0.0203	0.1872	0.1982
S12	0.0417	0.0233	0.2833	0.0481	0.3333	0.0750	0.0648	0.3333	0.0750
S13	0.1561	0.1742	0.1630	0.2182	0.2278	0.1773	0.1939	**0.1370**	0.1773
S14	0.2448	0.2260	0.2635	0.2354	0.2974	**0.2120**	0.2307	0.3208	0.2120
S15	0.0463	**0.0338**	0.0567	0.0363	0.0683	0.0625	**0.0338**	0.0917	0.0625
S16	0.2667	**0.2654**	0.3574	0.2730	0.4094	0.2951	0.2847	0.4290	0.2951
S17	0.0385	**0.0385**	0.1833	0.0462	0.1444	0.0692	**0.0385**	0.1194	0.0692
S18	0.0250	0.0125	0.3333	0.0375	0.0667	0.0438	0.0438	0.2333	0.0438
AVG	0.10±0.08	0.09±0.08	0.21±0.10	0.10±0.08	0.22±0.11	0.13±0.08	0.10±0.08	0.22±0.01	0.13±0.08

TABLE 10. Worst fitness value of FS methods.

Worst	AHA	QAHA	lshade	WOA	**SaDE**	TLBO	GWO	LSPACMA	GA
S1	0.0959	**0.0462**	0.0833	0.0976	0.0917	0.1023	0.0818	0.1167	0.1228
S2	0.1040	0.0944	0.1254	**0.0856**	0.1281	0.1065	0.1011	0.1342	0.1365
S3	0.0748	**0.0457**	0.0655	0.1037	0.1230	0.1017	0.1431	0.0575	0.1330
S4	0.1502	**0.1187**	0.2960	0.2822	0.3853	0.2845	0.2372	0.2977	0.3096
S5	0.2892	**0.1922**	0.2258	0.3117	0.2452	0.3469	**0.2121**	0.2223	0.3559
S6	0.2205	0.2115	**0.2019**	0.2513	0.2481	0.2795	0.2692	0.2759	0.2090
S7	0.1086	0.1134	0.1160	0.1279	0.1197	0.1894	0.0996	0.1629	0.1389
S8	0.1046	0.0976	0.3942	0.1160	0.3658	0.1399	0.1215	0.3915	0.1340
S9	0.1542	0.1678	0.2633	0.2467	0.2583	0.2600	0.2052	0.2167	0.2278
S10	0.1052	**0.0705**	0.2118	0.1836	0.2855	0.2582	0.1785	0.3258	0.2106
S11	0.1640	**0.0135**	0.3200	0.0852	0.3667	**0.0449**	0.0905	0.3077	0.2065
S12	0.0881	**0.0700**	0.2833	0.0867	0.3333	0.1705	0.1243	0.4500	0.1081
S13	0.2318	0.2258	0.1630	0.2379	0.2556	0.2273	0.2652	**0.1370**	0.2227
S14	0.2729	**0.2418**	0.2635	0.2917	0.3010	0.3040	0.2924	0.3208	0.2793
S15	0.1075	0.0650	**0.0567**	0.0950	0.1017	0.1075	0.0975	0.1367	0.1438
S16	0.3095	**0.2998**	0.3574	0.3229	0.4094	0.3348	0.3215	0.4472	0.3215
S17	0.0635	**0.0615**	0.1833	0.0865	0.1722	0.1192	0.0712	0.2000	0.1192
S18	0.0438	**0.0250**	0.3333	0.0804	0.1000	0.0866	0.0866	0.2333	0.0688
AVG	0.15±0.08	0.12±0.08	0.22±0.10	0.17±0.09	0.24±0.11	0.19±0.09	0.17±0.08	0.25±0.11	0.19±0.08

TABLE 11. Mean rank for QAHA and other methods using Friedman test for UCI datasets.

	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	GA	P-value
Sensitivity	2.8333	7.75	3.8056	4.8611	3.8333	6.7778	5.0278	4	6.1111	1.36E-08
Accuracy	2.0556	8.25	2.8889	5.2222	3.3889	7.7222	5.5833	3.8889	6	1.29E-15
Specificity	2.7778	6.8056	3.9444	5.1111	3.4444	6.4444	5.6944	4.9167	5.8611	1.56E-05
Average of Fit	3.0556	1.3333	6.3333	3.6667	7.3889	6.1111	4.0556	7.3333	5.7222	1.22E-14
STD of Fit	4.2222	3.2778	5.5	6.6944	4.5278	5	5.9444	4.8333	5	0.0149
Best of Fit	3.0556	2.0833	7.1111	3.3056	7.8056	5.5556	3.2222	7.3056	5.5556	5.90E-16
worst of Fit	3.4167	1.5556	5.2778	4.5556	6.8333	6.3056	4.2778	6.9444	5.8333	6.08E-10

- **GPS trajectories**: A collection of 163 trajectories of vehicle movements, including car and bus. The features presented in the dataset include trajectory id, GPS coordinates (latitude, longitude), and trajectory duration. The labels are the vehicle type (car or bus).
- **Hepatitis**: A collection of patients with hepatitis C that consists of 155 instances, two classes (die and live), and 19 features. The features include categorical features such as sex, steroid, fatigue, malaise, and numerical features such as age, BILIRUBIN, ALBUMIN, and

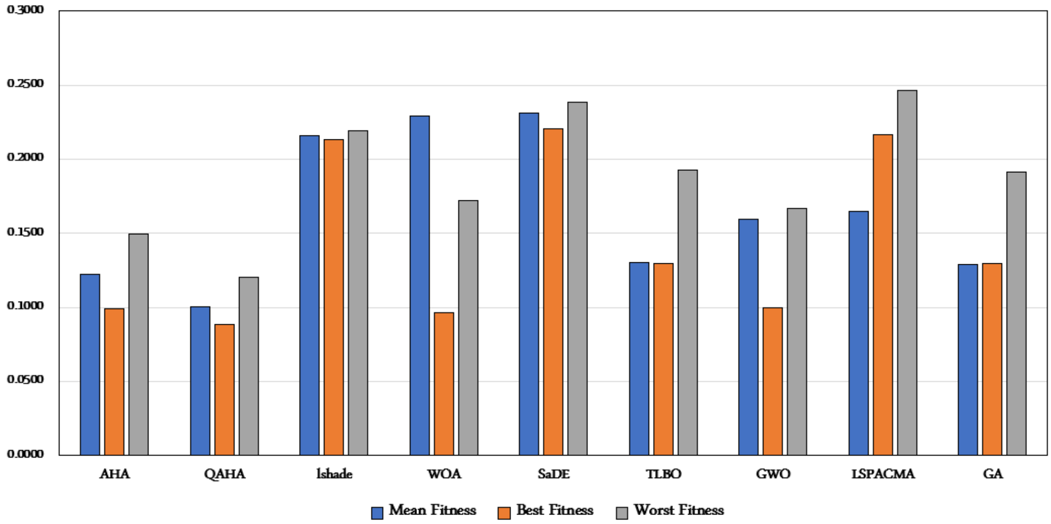


FIGURE 8. Comparison of fitness values of different FS optimizers.

others. The aim is to predict if the patient dies or lives.

- **GAS sensors**: A collection of sensors that measure the concentration of various gases in the air using 8 MOX gas sensors, humidity, and temperature. The data consists of 100 time-series records with a total of 11 features. The data was collected in the home environment, and the labels are the background, wine, and banana activity. The aim is to predict the activity and differentiate between the background, wine, and banana.
- **MovementAAL**: A collection of sensors that measure user movement in an office area covered by a Wireless Sensor Network (WSN). The dataset was created as a real-life benchmark in Ambient Assisted Living. The features were collected based on the captured radio signal strength (RSS) from the WSN nodes and the user using five sensors and four anchors. The task is formulated as a binary classification to predict the user movement pattern, including the location changing or preserving movements in an office environment consisting of two rooms.

2) RESULTS AND DISCUSSION

This section discusses the performance of a deep learning model based on a convolutional neural network and a Transformer-based model used to learn different feature representations from the raw input data. The objective of using the DL model [45] is to learn meaningful patterns and data representations from the input data rather than relying on the raw features. For instance, the transformer-based models can learn and generate contextual text representations from the input text, enhancing overall performance. Thus, various feature sizes were extracted from different datasets with a variation of feature number. The extracted features are fed to different FS optimization algorithms in the machine learning pipeline to assess FS approaches' performance on reducing the extracted features' dimensionality and improving overall performance. Table 13 shows the number of features generated using the DL model.

The comparison results between the proposed method and other methods are given in Table 14. We discover that the proposed QAHA achieves the best accuracy with all datasets, i.e., 0.907, 0.987, 0.922, and 0.846 for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively. QAHA outperformed the baseline model in three datasets. The enhanced accuracies are 5.55%, 0.35%, and 7.70% for Trajectory, GAS sensors, and MovementAAL, respectively. Compared with the AHA, QAHA enhances the accuracy of the four datasets (i.e., 5.56%, 0.12%, 2.60%, and 6.73% for Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). The proposed optimizer has better accuracy than the LASHDE optimizer with three datasets (2.85%, 2.81%, and 6.73% for Trajectory, GAS sensors, and MovementAAL datasets, respectively). Compared with the WOA optimizer, QAHA has considerably improved accuracy for all datasets (i.e., 11.11%, 6.33%, 3.90%, and 12.50 % for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). By comparing QAHA with SaDE optimizer, we find that our optimized beats SaDE with the four datasets (i.e., 9.26%, 0.35%, 2.60%, 8.65% for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). In the same way, QAHA beats TLBO with the four datasets (i.e., 11.11%, 6.33%, 2.60%, and 6.73%, for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). Compared with bGWO, QAHA achieves superior accuracies with the four datasets (i.e., 11.11%, 0.59%, 2.60%, and 6.73% for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). Finally, QAHA beats both LSPACMA and SGA, where for LSPACMA, the enhanced accuracy with all datasets are 9.26%, 0.71%, 2.60%, and 7.69% for the Trajectory, GAS sensors, Hepatitis, and MovementAAL

TABLE 12. Characteristics of the datasets used in our experiments.

Dataset	Data Type	#Features	#Instances	#Classes	Task
GPS trajectories	Numerical	6	163	2	Predict vehicle type Car or Bus
Hepatitis	Numerical	19	155	2	Healthcare
GAS sensors	Numerical	11	919438	3	HAR
MovementAAL	Numerical	4	13197	2	HAR

TABLE 13. Characteristics of the datasets used in our experiments.

Dataset	Number of Features	Number of Instances	Number of Classes
GPS trajectories	769	163	2
Hepatitis	2433	232	2
GAS sensors	129	919438	3
MovementAAL	513	13197	2

datasets, respectively; for GAS, QAHA improves the accuracy as follows: 5.56%, 0.24%, 1.30%, and 5.77% for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively. Figure 9 summarizes the results of different optimizers and highlights the level of improvement for the QAHA.

Table 15 reports the specificity of the DL models based on different feature selection optimization techniques. QAHA achieves specificity of 0.84, 0.9967, 0.9431, and 0.9259 for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets. QAHA beat all other optimizers with the last two datasets. For the Trajectory dataset, SaDE achieves the best specificity of 0.88; for the GAS sensors dataset, GA and AHA achieve the highest specificity of 0.9983. Note that the difference between QAHA and AHA with the GAS sensors dataset is 0.1658% which is insignificant. As discussed for the accuracy metric, on average, QAHA improves the specificity by 2.00% compared with all other optimizers for the Hepatitis dataset and by 5% for the MovementAAL dataset. Figure 10 shows the specificity results of all optimizers with the four datasets.

Table 16 reports the sensitivity metric of the nine optimizers with the four datasets. This table shows that the QAHA achieves the best sensitivity results compared to other algorithms for the four datasets (i.e., 0.897, 0.996, 0.921, and 0.740 for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). Compared to the AHA algorithm, QAHA has an improved average performance of 4.78% with the four datasets. Compared with the other optimizers, QAHA achieves an improved average sensitivity of 8.72%, 6.00%, 6.97%, 5.50%, 7.07%, 6.70%, and 3.12% compared with QAHA, LASHDE, WOA, SaDE, TLBO, bGWO, LSPACMA, and SGA, respectively for the four datasets. Figure 11 shows the sensitivity results of all optimizers with the four datasets.

QAHA achieves the best results compared to other optimizers. These results are consistent with the four datasets. The main reason for these results is that the QAHA selected the most informative features. The algorithm can identify the less relevant features. These features could be noisy or correlated with other features. As a result, removing these features makes the final dataset more clean and consistent. Compared to other optimizers and the baseline model, QAHA selected the least number of features with all datasets, see Table 17. QAHA selected 159, 45, 436, and 70 from the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets. QAHA significantly reduces the number of selected features. This causes the resulting models to be faster and easier to train, more accurate, and lighter to deploy. Compared with the original feature sets, QAHA removes large percentages of irrelevant features from each dataset (removed percentages from the original feature set are 79.32%, 65.12%, 82.08%, and 86.35% for the Trajectory, GAS sensors, Hepatitis, and MovementAAL datasets, respectively). Compared with the other optimizers, QAHA reduces the selected features from the Trajectory dataset by large percentages (i.e., 11.44%, 38.75%, 18.34%, 49.54%, 61.25%, 24.71%, 16.91%, and 69.96% for AHA, QAHA, LASHDE, WOA, SaDE, TLBO, bGWO, LSPACMA, and SGA, respectively. For the GAS sensors dataset, QAHA reduces the selected features by the following percentages: 25.58%, 24.03%, 37.21%, 37.98%, 37.21%, 5.43%, 34.88%, and 30.23% for AHA, QAHA, LASHDE, WOA, SaDE, TLBO, bGWO, LSPACMA, and SGA, respectively. With the Hepatitis dataset, 34.65%, 36.46%, 11.96%, 35.27%, 65.84%, 2.79%, 1.44%, and 44.10% have been reduced compared with the AHA, QAHA, LASHDE, WOA, SaDE, TLBO, bGWO, LSPACMA, and GA optimizers, respectively. For the MovementAAL datasets, QAHA reduced the percentage of selected features by 38.60%, 37.43%, 17.93%, 24.37%, 67.25%, 18.91%, 16.57%, and 9.36% for the AHA, QAHA, LASHDE, WOA, SaDE, TLBO, bGWO, LSPACMA, and GA optimizers, respectively. Figure 12 compares optimization techniques regarding the selected feature sets.

Table 18 reports the values of the fitness function for different optimization techniques with different datasets. The lower the fitness values, the better the algorithm because

TABLE 14. Accuracy of the SIoT datasets with different optimizers.

	Baseline	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	SGA
GPS trajectories	0.8519	0.8519	**0.9074**	0.8789	0.7963	0.8148	0.7963	0.7963	0.8148	0.8519
GAS sensors	0.9836	0.9859	**0.9871**	0.9590	0.9237	0.9835	0.9237	0.9812	0.9800	0.9847
Hepatitis	0.9221	0.8961	**0.9221**	**0.9221**	0.8831	0.8961	0.8961	0.8961	0.8961	0.9091
MovementAAL	0.7692	0.7788	**0.8462**	0.7788	0.7212	0.7596	0.7788	0.7788	0.7692	0.7885
AVG	0.86±0.08	0.88±0.09	0.92±0.06	0.88±0.08	0.83±0.09	0.86±0.10	0.85±0.07	0.86±0.09	0.87±0.08	0.88±0.08

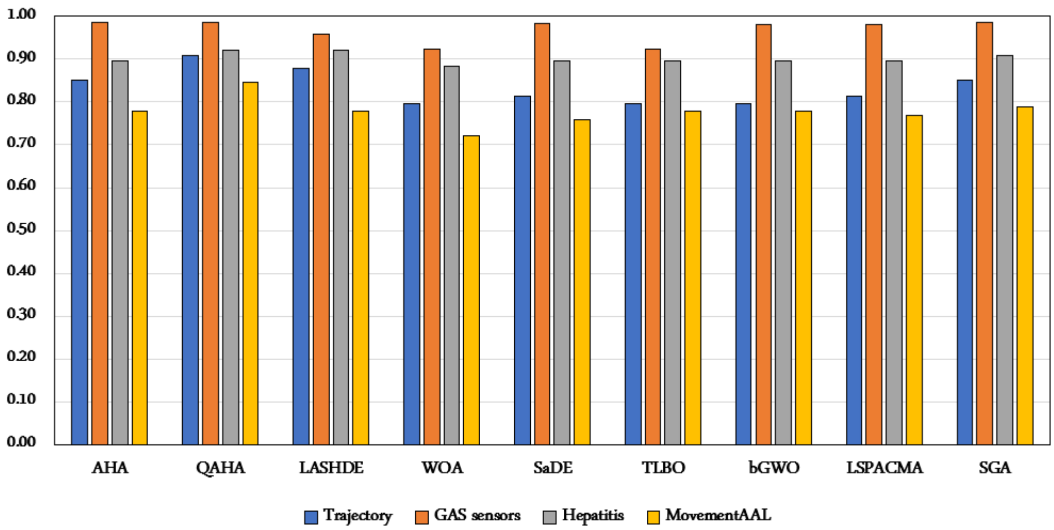


FIGURE 9. Comparison of accuracy metric of different FS optimizers for the SIoT datasets.

TABLE 15. Specificity of different optimizers for the SIoT datasets.

	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	SGA
GPS trajectories	0.8400	0.8400	0.8400	0.8400	**0.8800**	0.8400	0.8400	0.8400	0.8400
GAS sensors	**0.9983**	0.9967	0.9967	0.9967	0.9967	0.9967	0.9967	0.9967	**0.9983**
Hepatitis	0.9231	**0.9431**	0.9231	0.9231	0.9231	0.9231	0.9231	0.9231	0.9231
MovementAAL	0.9074	**0.9259**	0.8148	0.8519	0.8889	0.8889	0.8889	0.8889	0.8889
AVG	0.92±0.06	0.93±0.06	0.89±0.07	0.90±0.06	0.92±0.05	0.91±0.06	0.91±0.06	0.91±0.06	0.91±0.06

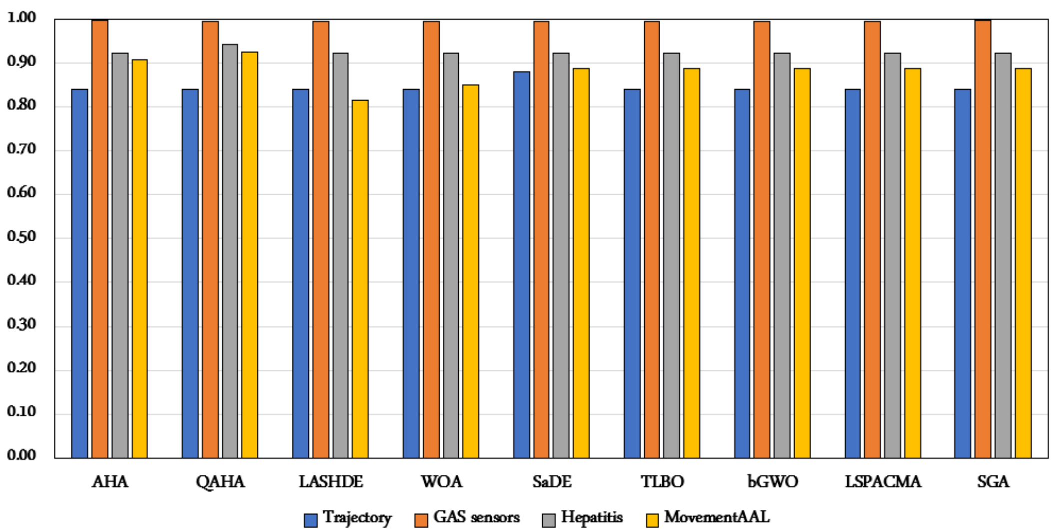


FIGURE 10. Comparison of Specificity metric of different FS optimizers for the SIoT datasets.

we need to minimize the objective function. As seen in Table 18, QAHA achieves the lowest fitness values with three of the four datasets (i.e., 0.08372,0.01582, and 0.07021 for Trajectory, GAS sensors, and Hepatitis, respectively). For the MovementAAL dataset, SaDE achieves the lowest fitness value of 0.1772. Figure 13 illustrates the comparison of

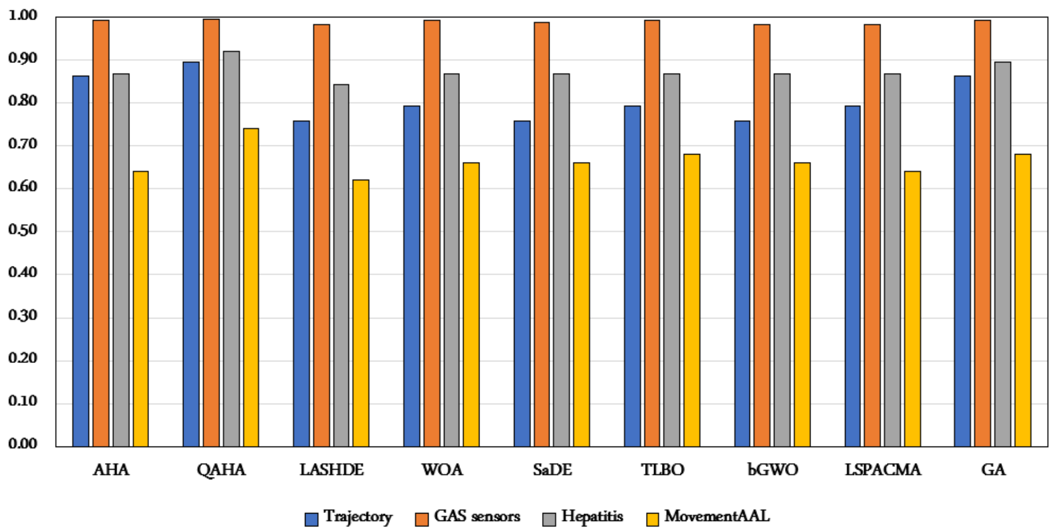


FIGURE 11. Comparison of Sensitivity metric of different FS optimizers for the SIoT datasets.

TABLE 16. Sensitivity of different optimizers for the SIoT datasets.

	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	GA
GPS trajectories	0.8621	**0.8966**	0.7586	0.7931	0.7586	0.7931	0.7586	0.7931	0.8621
GAS sensors	0.9919	**0.9960**	0.9839	0.9919	0.9879	0.9919	0.9839	0.9839	0.9919
Hepatitis	0.8684	**0.9211**	0.8421	0.8684	0.8684	0.8684	0.8684	0.8684	0.8947
MovementAAL	0.6400	**0.7400**	0.6200	0.6600	0.6600	0.6800	0.6600	0.6400	0.6800
AVG	0.84±0.15	0.89±0.11	0.80±0.15	0.83±0.14	0.82±0.14	0.83±0.13	0.82±0.14	0.82±0.14	0.86±0.13

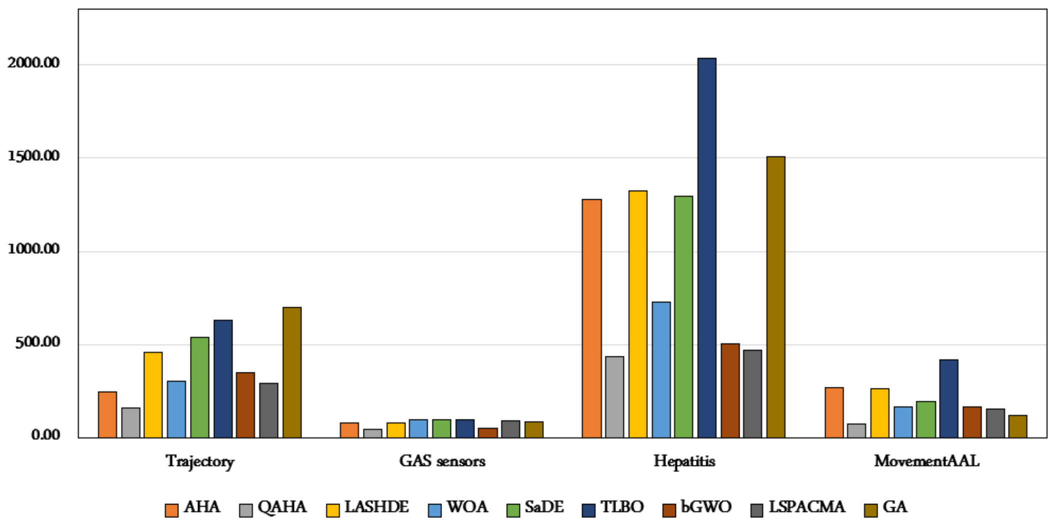


FIGURE 12. Comparison of the number of selected features by each optimizer.

TABLE 17. Number of selected features for SIoT.

	Baseline	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	GA
GPS trajectories	769	247	**159**	457	300	540	630	349	289	697
GAS sensors	129	78	**45**	76	93	94	93	52	90	84
Hepatitis	2433	1279	**436**	1323	727	1294	2038	504	471	1509
MovementAAL	513	268	**70**	262	162	195	415	167	155	118
AVG	550±520	468±547	177±179	529±551	320±284	530±543	794±858	268±199	251±168	602±667

fitness values for different optimizers with the four datasets, and Figure 14 shows the distribution of the fitness values using a box plot. These figures show that QAHA has mainly achieved minimal fitness values compared with other

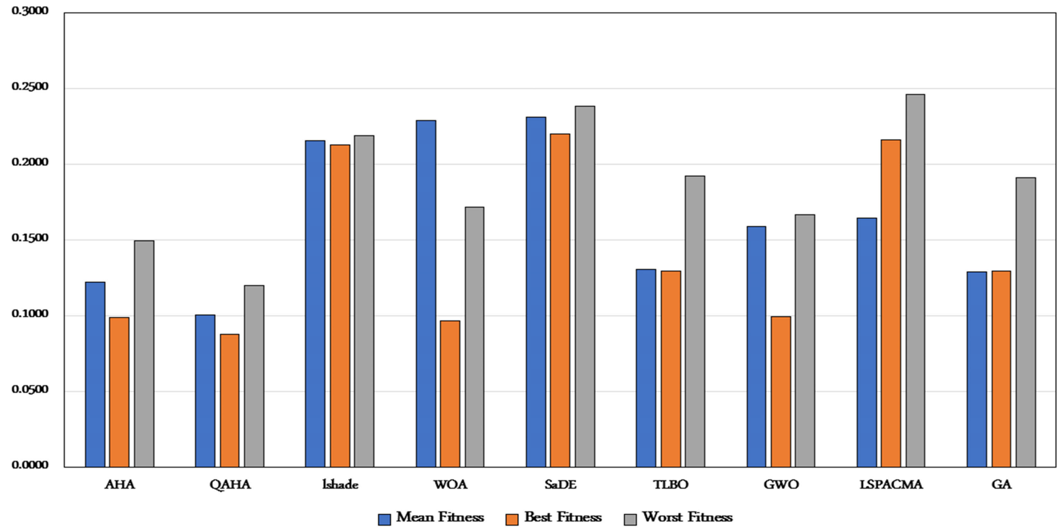


FIGURE 13. Comparison of the number of selected features by each optimizer.

TABLE 18. Fitness value of the optimization algorithms with the SIoT datasets.

	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	GA
GPS trajectories	0.1335	**0.0837**	0.2238	0.1680	0.1001	0.2654	0.2027	0.1749	0.1340
GAS sensors	0.0182	**0.0158**	0.0575	0.0406	0.0171	0.0885	0.0341	0.0469	0.0169
Hepatitis	0.0703	**0.0702**	0.1391	0.1111	0.0703	0.1773	0.1142	0.1129	0.0842
MovementAAL	0.2088	0.2371	0.2392	0.2285	**0.1772**	0.2801	0.2317	0.2184	0.1939
AVG	0.11±0.08	0.10±0.09	0.16±0.08	0.14±0.08	0.09±0.07	0.20±0.09	0.15±0.09	0.14±0.07	0.11±0.08

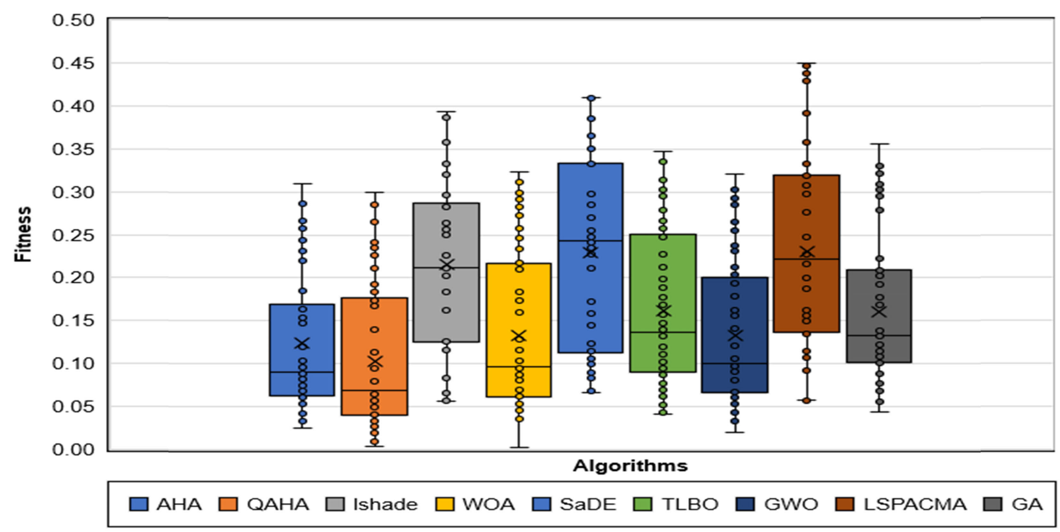


FIGURE 14. Comparison of the fitness by each optimizer.

optimizers. We note from the previous discussions of the results using different datasets that the QAHA consistently outperforms the other eight optimizers and the baseline models. In other words, QAHA beats the eight optimizers with 22 datasets from different domains. We can conclude that QAHA is a stable feature selection optimizer that always selects the best and most informative list of features that carry the majority of variance in different datasets.

Moreover, Table 19 illustrates the mean rank of the developed methods and other methods using the Friedman test. These results show a significant difference between the QAHA and other methods according to Accuracy, No. Features, Sensitivity, and Fitness value. In addition, according to the p-value obtained using the Specificity results, there is no significant difference between QAHA and other methods. However, the mean rank of QAHA is still the

TABLE 19. Mean rank for QAHA and other methods using Friedman test for SIoT datasets.

	AHA	QAHA	LASHDE	WOA	SaDE	TLBO	bGWO	LSPACMA	GA	P-value
Accuracy	6.75	9.625	6.75	1.375	4.125	3.5	4.375	4.125	7.875	0.0036
Specificity	6.375	6.625	3.5	3.75	5.625	4.5	4.5	4.5	5.625	0.2158
No. Features	4.75	1	5.75	4.875	7	8.375	3.75	3.5	6	0.0116
Sensitivity	5.25	9	1.5	5.25	3.875	5.875	3.375	3.5	7.375	0.0014
Fitness	3.125	2.5	8	5.25	2.125	9	6.25	5.75	3	9.31E-04

higher one, which indicates it is applicable to use QAHA in real-world applications, including social IoT.

These results indicate the high ability of the developed QAHA to balance exploration and exploitation, leading to improved output performance. Therefore, the error classification and the number of selected features are reduced.

V. CONCLUSION

Regarding shared information on SIoT systems, the data types and sources can differ based on the object generating the data, environment, and application. This study covers different types of data generated in an IoT environment related to SIoT, including features extracted from interconnected objects over the Internet, social media data, sensor data, transportation data, and healthcare data. We introduced an enhanced version of the Artificial Hummingbird algorithm (AHA) using the quantum concept as a feature selection technique. The main objective of using the quantum is to enhance the exploration of agents while determining the optimal subset of features. To validate the efficiency of the developed QAHA, a set of experimental series has been conducted using benchmark and real social IoT datasets. In addition, the results of QAHA have been compared with well-known FS methods, including LASHDE, WOA, SaDE, TLBO, GWO, LSPACMA, and GA. The comparison results have shown the high performance of QAHA overall in the compared methods among the tested data. For 18 UCI datasets, the accuracy of 93%, Specificity of 91%, Sensitivity of 95%, Number of selected features of 468, and Fitness value is 0.10. The proposed QAHA has an accuracy of 88%, Specificity of 93%, Sensitivity of 89%, Number of selected features of 468, and a Fitness value is 0.10 for the SIoT dataset. With this achievement of QAHA, however, it still suffers from time complexity and also, the mechanism that can be used to determine the initial population. Since this initial value for the solutions has the largest influence on the performance of QAHA towards the optimal subset of features.

The presented QAHA model can be extended as a multi-objective FS and applied to applications such as image segmentation in future work. It can also be used as a task scheduler in the Internet of Things. Moreover, the optimization algorithms can be very beneficial in big data applications related frameworks that benefit from IoT and social networks such as marketing, telecommunication, and finance in terms of reducing data noise, resource, and bandwidth consumption, and improving the overall efficiency in low-resource environments.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through project number (IF-PSAU-2022/01/19574).

REFERENCES

[1] M. M. Rad, A. M. Rahmani, A. Sahafi, and N. N. Qader, ''Social Internet of Things: Vision, challenges, and trends,'' *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, pp. 1–40, Dec. 2020.
[2] S. Rho and Y. Chen, ''Social Internet of Things: Applications, architectures and protocols,'' *Future Gener. Comput. Syst.*, vol. 81, pp. 667–668, May 2018.
[3] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, ''Supervised feature selection techniques in network intrusion detection: A critical review,'' *Eng. Appl. Artif. Intell.*, vol. 101, May 2021, Art. no. 104216.
[4] P. Ghosh, S. Azam, M. Jonkman, A. Karim, F. M. J. M. Shamrat, E. Ignatious, S. Shultana, A. R. Beeravolu, and F. De Boer, ''Efficient prediction of cardiovascular disease using machine learning algorithms with relief and LASSO feature selection techniques,'' *IEEE Access*, vol. 9, pp. 19304–19326, 2021.
[5] J. Too and S. Mirjalili, ''A hyper learning binary dragonfly algorithm for feature selection: A COVID-19 case study,'' *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106553.
[6] Y. Zhang, S. Wang, K. Xia, Y. Jiang, and P. Qian, ''Alzheimer's disease multiclass diagnosis via multimodal neuroimaging embedding feature selection and fusion,'' *Inf. Fusion*, vol. 66, pp. 170–183, Feb. 2021.
[7] M. Azizi, ''Atomic orbital search: A novel metaheuristic algorithm,'' *Appl. Math. Model.*, vol. 93, pp. 657–683, May 2021.
[8] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, ''African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems,'' *Comput. Ind. Eng.*, vol. 158, Aug. 2021, Art. no. 107408.
[9] A. Mohammadi-Balani, M. D. Nayeri, A. Azar, and M. Taghizadeh-Yazdi, ''Golden eagle optimizer: A nature-inspired metaheuristic algorithm,'' *Comput. Ind. Eng.*, vol. 152, Feb. 2021, Art. no. 107050.
[10] W. Zhao, L. Wang, and S. Mirjalili, ''Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications,'' *Comput. Methods Appl. Mech. Eng.*, vol. 388, Jan. 2022, Art. no. 114194.
[11] M. A. Navarro, D. Oliva, A. Ramos-Michel, D. Zaldívar, B. Morales-Castañeda, M. Pérez-Cisneros, A. Valdivia, and H. Chen, ''An improved multi-population whale optimization algorithm,'' *Int. J. Mach. Learn. Cybern.*, vol. 13, pp. 2447–2478, Apr. 2022.
[12] I. Aranguren, A. Valdivia, M. Pérez-Cisneros, D. Oliva, and V. Osuna-Enciso, ''Digital image thresholding by using a lateral inhibition 2D histogram and a mutated electromagnetic field optimization,'' *Multimedia Tools Appl.*, vol. 81, no. 7, pp. 10023–10049, Mar. 2022.
[13] K. H. Almotairi and L. Abualigah, ''Improved reptile search algorithm with novel mean transition mechanism for constrained industrial engineering problems,'' *Neural Comput. Appl.*, vol. 34, pp. 17257–17277, May 2022.
[14] M. H. Nadimi-Shahraki, A. Fatahi, H. Zamani, S. Mirjalili, and L. Abualigah, ''An improved moth-flame optimization algorithm with adaptation mechanism to solve numerical and mechanical engineering problems,'' *Entropy*, vol. 23, no. 12, p. 1637, Dec. 2021.
[15] M. H. Nadimi-Shahraki, M. Banaie-Dezfouli, H. Zamani, S. Taghian, and S. Mirjalili, ''B-MFO: A binary moth-flame optimization for feature selection from medical datasets,'' *Computers*, vol. 10, no. 11, p. 136, Oct. 2021.
[16] D. Yousri, M. A. Elaziz, D. Oliva, A. Abraham, M. A. Alotaibi, and M. A. Hossain, ''Fractional-order comprehensive learning marine predators algorithm for global optimization and feature selection,'' *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107603.

[17] P. Agrawal, T. Ganesh, and A. W. Mohamed, "Chaotic gaining sharing knowledge-based optimization algorithm: An improved metaheuristic algorithm for feature selection," *Soft Comput.*, vol. 25, no. 14, pp. 9505–9528, Jul. 2021.

[18] D. S. A. Elminaam, A. Nabil, S. A. Ibraheem, and E. H. Houssein, "An efficient marine predators algorithm for feature selection," *IEEE Access*, vol. 9, pp. 60136–60153, 2021.

[19] M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1103–1127, May 2021.

[20] M.-C. Yuen, S.-C. Ng, and M.-F. Leung, "A competitive mechanism multi-objective particle swarm optimization algorithm and its application to signalized traffic problem," *Cybern. Syst.*, vol. 52, no. 1, pp. 73–104, Jan. 2021.

[21] S. Yang, M. Wang, and L. Jiao, "A quantum particle swarm optimization," in *Proc. Congr. Evol. Comput.*, vol. 1, 2004, pp. 320–324.

[22] A. SaiToh, R. Rahimi, and M. Nakahara, "A quantum genetic algorithm with quantum crossover and mutation operations," *Quantum Inf. Process.*, vol. 13, no. 3, pp. 737–755, Mar. 2014.

[23] D. Mohammadi, M. A. Elaziz, R. Moghdani, E. Demir, and S. Mirjalili, "Quantum Henry gas solubility optimization algorithm for global optimization," *Eng. Comput.*, vol. 38, pp. 2329–2348, Mar. 2021.

[24] V. Kumar, G. Bass, C. Tomlin, and J. Dulny, "Quantum annealing for combinatorial clustering," *Quantum Inf. Process.*, vol. 17, no. 2, pp. 1–14, Feb. 2018.

[25] R. Chen, C. Dong, Y. Ye, Z. Chen, and Y. Liu, "QSSA: Quantum evolutionary salp swarm algorithm for mechanical design," *IEEE Access*, vol. 7, pp. 145582–145595, 2019.

[26] J. García and C. Maureira, "A KNN quantum cuckoo search algorithm applied to the multidimensional knapsack problem," *Appl. Soft Comput.*, vol. 102, Apr. 2021, Art. no. 107077.

[27] M. Abd Elaziz, D. Mohammadi, D. Oliva, and K. Salimifard, "Quantum marine predators algorithm for addressing multilevel image segmentation," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107598.

[28] K. B. Prakash, "Quantum meta-heuristics and applications," in *Cognitive Engineering for Next Generation Computing: A Practical Analytical Approach*. 2021, pp. 265–297.

[29] Z. A. Dahi and E. Alba, "Metaheuristics on quantum computers: Inspiration, simulation and real execution," *Future Gener. Comput. Syst.*, vol. 130, pp. 164–180, May 2022.

[30] M. S. Abid, H. J. Apon, K. A. Morshed, and A. Ahmed, "Optimal planning of multiple renewable energy-integrated distribution system with uncertainties using artificial hummingbird algorithm," *IEEE Access*, vol. 10, pp. 40716–40730, 2022.

[31] M. A. Hamida, R. A. El-Sehiemy, A. R. Ginidi, E. Elattar, and A. M. Shaheen, "Parameter identification and state of charge estimation of Li-ion batteries used in electric vehicles using artificial hummingbird optimizer," *J. Energy Storage*, vol. 51, Jul. 2022, Art. no. 104535.

[32] M. Ali, P. P. F. Rajeena, and D. S. A. Elminaam, "A feature selection based on improved artificial hummingbird algorithm using random opposition-based learning for solving waste classification problem," *Mathematics*, vol. 10, no. 15, p. 2675, Jul. 2022.

[33] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, "On stopping criteria for genetic algorithms," in *Advances in Artificial Intelligence—SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29–Ocotber 1, 2004. Proceedings 17* Berlin, Germany: Springer, 2004, pp. 405–413.

[34] K. Srikanth, L. K. Panwar, B. Panigrahi, E. Herrera-Viedma, A. K. Sangaiah, and G.-G. Wang, "Meta-heuristic framework: Quantum inspired binary grey wolf optimizer for unit commitment problem," *Comput. Electr. Eng.*, vol. 70, pp. 243–260, Aug. 2018.

[35] A. Narayanan, "Quantum computing for beginners," in *Proc. Congr. Evol. Comput.*, vol. 3, 1999, pp. 2231–2238.

[36] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.

[37] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[38] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1785–1791.

[39] M. Allam and M. Nandhini, "Optimal feature selection using binary teaching learning based optimization algorithm," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 34, no. 2, pp. 329–341, Feb. 2022.

[40] R. A. Ibrahim, M. A. Elaziz, and S. Lu, "Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Syst. Appl.*, vol. 108, pp. 1–27, Oct. 2018.

[41] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 145–152.

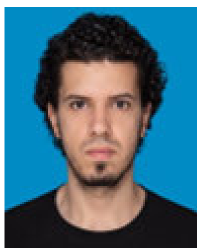
[42] L. S. Oliveira, N. Benahmed, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature subset selection using genetic algorithms for handwritten digit recognition," in *Proc. XIV Brazilian Symp. Comput. Graph. Image Process.*, 2001, pp. 362–369.

[43] A. Frank. (2010). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[44] A. Ahmad, M. Khan, A. Paul, S. Din, M. M. Rathore, G. Jeon, and G. S. Choi, "Toward modeling and optimization of features selection in big data based social Internet of Things," *Future Gener. Comput. Syst.*, vol. 82, pp. 715–726, May 2018.

[45] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

MOHAMED ABD ELAZIZ received the B.S. and M.S. degrees in computer science and the Ph.D. degree in mathematics and computer science from Zagazig University, Egypt, in 2008, 2011, and 2014, respectively. From 2008 to 2011, he was an Assistant Lecturer with the Department of Computer Science. He is an Associate Professor with Zagazig University. He is the author of more than 426 articles. He is one of the 2% influential scholars, which depicts the 100,000 top scientists in the world. His research interests include metaheuristic techniques, security IoT, cloud computing, machine learning, signal processing, image processing, and evolutionary algorithms.

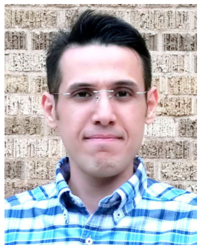


ABDELGHANI DAHOU received the B.S. and M.S. degrees in computer science and intelligent systems from the University of Ahmad Draia, Adrar, Algeria, in 2012 and 2014, respectively, and the Ph.D. degree in computer science from the Wuhan University of Technology, Wuhan, Hubei, China, in 2019. He is a Lecturer with the Faculty of Science and Technology, University of Ahmad Draia. His research interests include deep learning, signal processing, data mining, neuro-evolution, image processing, and natural language processing.

MOHAMMED AZMI AL-BETAR received the Ph.D. degree in artificial intelligence from Universiti Sains Malaysia, in 2010. He was a Postdoctoral Research Fellow with Universiti Sains Malaysia, for three years, where he was invited as a Visiting Researcher (two times). He is currently the Head of the Evolutionary Computation Research Group (ECRG), Artificial Intelligence Research Center (AIRC), Ajman University. He has been a full-time Faculty Member and a Coordinator of the Master of Science in Artificial Intelligence (M.Sc.-AI) Program, Ajman University, since 2020. Furthermore, he has more than 15 years of teaching experience in higher education institutions. He has taught several courses in the computer science and artificial intelligence fields. Before joining Ajman University, he was the Deputy Dean of academic affairs, the Deputy Dean of scientific research for quality assurance and development, and the Head of the Information Technology Department, Al-Balqa Applied University, Jordan. He has published more than 180 scientific articles in high-quality and well-reputed journals and conferences. Accordingly, he ranked in Stanford University's study of the world's top 2% of scientists in 2020/2021/2022.

SHAKER EL-SAPPAGH received the bachelor's degree in computer science from the Information Systems Department, Faculty of Computers and Information, Cairo University, Egypt, in 1997, the master's degree from Cairo University, in 2007, and the Ph.D. degree in computer science from the Information Systems Department, Faculty of Computers and Information, Mansura University, Mansura, Egypt, in 2015. In 2003, he joined the Department of Information Systems, Faculty of Computers and Information, Minia University, Egypt, as a Teaching Assistant. Since June 2016, he has been with the Department of Information Systems, Faculty of Computers and Information, Benha University, as an Assistant Professor. He worked as a Research Professor at the UWB Wireless Communications Research Center in the Department of Information and Communication Engineering at Inha University, South Korea for three years from 2018 to 2020. He worked as a Research Professor at the Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain for one year in 2021. Now, he has been an Associate Professor at Galala University, Egypt, since 2021. He has also been a Senior Researcher at the College of Computing and Informatics, Sungkyunkwan University, South Korea, since 2021. His publications in clinical decision support systems and semantic intelligence. His current research interests include machine learning, medical informatics, (fuzzy) ontology engineering, distributed and hybrid clinical decision support systems, semantic data modeling, fuzzy expert systems, and cloud computing. He is a reviewer of many journals. He is a reviewer in many journals, and he is very interested in the diseases' diagnosis and treatment research.

DIEGO OLIVA (Senior Member, IEEE) received the B.S. degree in electronics and computer engineering from the Industrial Technical Education Center (CETI), Guadalajara, Mexico, in 2007, the M.Sc. degree in electronic engineering and computer sciences from the Universidad de Guadalajara, Mexico, in 2010, and the Ph.D. degree in informatics from Universidad Complutense de Madrid, in 2015. Currently, he is an Associate Professor with Universidad de Guadalajara. Since 2020, he has been a Visiting Professor with Tomsk Polytechnic University, Russia. He is the coauthor of more than 100 articles in international journals and five books. His research interests include evolutionary and swarm algorithms, hybridization of evolutionary and swarm algorithms, and computational intelligence. He has the distinction of the National Researcher Rank 2 by the Mexican Council of Science and Technology. He is part of the editorial board of IEEE ACCESS, *PLOS One*, *Mathematical Problems in Engineering*, IEEE LATIN AMERICA TRANSACTIONS, and *Engineering Applications of Artificial Intelligence*.



AHMAD O. ASEERI (Member, IEEE) received the bachelor's degree in computing from King Saud University, Saudi Arabia, the master's degree in computer science from the University of Wisconsin–Madison, USA, and the Ph.D. degree in computer science from Texas Tech University, USA. He is currently an Assistant Professor with the Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Saudi Arabia. He received the Full Scholarship and the Government Scholarship for the master's and Ph.D. studies, respectively. His main research interests include artificial intelligence (AI), main focus in the area of deep learning, with application to neural network-based risk analysis in physical unclonable functions for resource constraint IoTs, natural language processing (NLP), and computer vision; data mining, with application to clustering techniques, including bisecting K-means clustering (BKM), limited-iteration bisecting K-means (LIBKM), and memory-aware clustering algorithms; and applied deep learning for medical applications.