



# An Improved Binary Quantum-based Avian Navigation Optimizer Algorithm to Select Effective Feature Subset from Medical Data: A COVID-19 Case Study

Ali Fatahi<sup>1,2</sup> · Mohammad H. Nadimi-Shahraki<sup>1,2</sup>  · Hoda Zamani<sup>1,2</sup>

Received: 26 May 2023 / Revised: 10 August 2023 / Accepted: 21 August 2023 / Published online: 25 September 2023  
© Jilin University 2023

## Abstract

Feature Subset Selection (FSS) is an NP-hard problem to remove redundant and irrelevant features particularly from medical data, and it can be effectively addressed by metaheuristic algorithms. However, existing binary versions of metaheuristic algorithms have issues with convergence and lack an effective binarization method, resulting in suboptimal solutions that hinder diagnosis and prediction accuracy. This paper aims to propose an Improved Binary Quantum-based Avian Navigation Optimizer Algorithm (IBQANA) for FSS in medical data preprocessing to address the suboptimal solutions arising from binary versions of metaheuristic algorithms. The proposed IBQANA's contributions include the Hybrid Binary Operator (HBO) and the Distance-based Binary Search Strategy (DBSS). HBO is designed to convert continuous values into binary solutions, even for values outside the [0, 1] range, ensuring accurate binary mapping. On the other hand, DBSS is a two-phase search strategy that enhances the performance of inferior search agents and accelerates convergence. By combining exploration and exploitation phases based on an adaptive probability function, DBSS effectively avoids local optima. The effectiveness of applying HBO is compared with five transfer function families and thresholding on 12 medical datasets, with feature numbers ranging from 8 to 10,509. IBQANA's effectiveness is evaluated regarding the accuracy, fitness, and selected features and compared with seven binary metaheuristic algorithms. Furthermore, IBQANA is utilized to detect COVID-19. The results reveal that the proposed IBQANA outperforms all comparative algorithms on COVID-19 and 11 other medical datasets. The proposed method presents a promising solution to the FSS problem in medical data preprocessing.

**Keywords** Feature subset selection · Optimization · Binary metaheuristic algorithms · Bioinspired · Machine learning · Medical datasets

## 1 Introduction

During the past years, with the increasing volume of data, the role of data mining techniques for extracting patterns and ultimately discovering hidden knowledge has become more prominent. Knowledge discovery involves searching a large amount of data to identify hidden patterns and extract relevant insights [1]. Applying data mining techniques to medical data has increasing benefits for medical diagnosticians

and can minimize the potential errors of inexperienced physicians [2]. An accurate and reliable diagnosis in the early stages of the disease can significantly impact the patient's life and expenses. Since medical datasets are often collected from multiple sources and for different purposes, besides relevant features, they may contain many irrelevant and redundant ones. Relevant features include essential information and can be used to discover knowledge and hidden patterns. Redundant features comprise necessary information already offered by another feature; as a result, these features don't provide the prediction model any extra useful data and only cause the computational cost for the classification algorithm. Features that do not provide useful information are irrelevant, and their presence causes a decline in the precision of prediction models' classification ability. As a result, the existence of irrelevant and redundant features not only imposes an additional computational cost to data

✉ Mohammad H. Nadimi-Shahraki  
nadimi@iaun.ac.ir; nadimi@ieee.org

<sup>1</sup> Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran

<sup>2</sup> Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran

mining algorithms but also degrades the accuracy of the predictions [3].

Feature Subset Selection (FSS) is a common way to overcome the mentioned challenges by selecting a set of  $M$  effective and relevant features from an original set of  $N$  features where  $M < N$ . Generally, there are three main types of FSS techniques: filter-based methods, wrapper-based methods, and embedded-based methods [4, 5]. Without utilizing machine learning techniques, filter-based methods evaluate the relevance of features based on criteria including consistency, correlation, similarity, distance, information, variance, and statistical criteria to select features [6]. Filter-based feature selection using ant colony optimization [7] and filter-based binary particle swarm optimization [8] are two prominent FSS algorithms that employ the filter-based method. Although filter-based FSS techniques are generally less computationally expensive than other methods due to their independence from classification algorithms, they usually do not yield adequate performance. The wrapper-based method, popularized by Kohavi and John [9], evaluates the selected feature subset's quality using a classification algorithm. Despite outperforming the filter-based method in accuracy, the frequent usage of the classification algorithm makes it computationally more expensive and requires more execution time than the filter-based methodology. [10]. Embedded-based methods are another category of FSS algorithms, performing selection during training and specific to certain learning machines [11]. They are more efficient by avoiding retraining predictors for each subset but are limited to specific machines and can be intricate. Examples include FSS based on support vector machine [12, 13] and embedded-based genetic programming [14].

As the wrapper-based approach yields better results [15], three search strategies have been used to generate numerous FSS algorithms: sequential, exponential, and random search. The features are progressively added or removed in the sequential search strategy [16, 17], but this search strategy is prone to trap in local optima solutions [18]. The exponential search strategy makes sure to find the optimal features subset by evaluating all possible feature subsets [19]; however, they are computationally costly and inapplicable for many real-world datasets [18]. The random search strategy begins with a random feature subset and then proceeds with a sequential search strategy. Metaheuristic algorithms, a prominent class of random search methods, use stochastic techniques to solve optimization problems. They explore the search space, promote population diversity, and bypass local optimum solutions by embedding randomness into their search procedures. Additionally, they locally search promising areas to enhance solution quality [20]. Particle Swarm Optimization (PSO) [21], Genetic Algorithm (GA) [22], Differential Evolution (DE) [23], Ant Colony Optimization (ACO) [24], Artificial Bee Colony (ABC) [25], Grey

Wolf Optimizer (GWO) [26], Moth-Flame Optimization (MFO) [27], Whale Optimization Algorithm (WOA) [28], and Harris Hawks Optimization (HHO) [29] are some well-known metaheuristic algorithms proposed to tackle different optimization problems such as task scheduling [30–32], optimal power flow [33, 34], and engineering design [35–40]. Researchers continue developing new metaheuristic algorithms to cover some flaws of previous algorithms, such as low convergence rate due to randomization in their methods [41, 42] and the lack of mechanisms to maintain population diversity and the imbalance between search strategies [43]. For instance, the Marine Predators Algorithm (MPA) [44], Colony Predation Algorithm (CPA) [45], Aquila Optimizer (AO) [46], Starling Murmuration Optimizer (SMO) [47], dwarf mongoose optimization algorithm [48], and Quantum-based Avian Navigation Optimizer Algorithm (QANA) [49] are among the recent metaheuristic algorithms proposed for continuous problem-solving [50, 51]. Among these, QANA is a scalable population-based metaheuristic algorithm inspired by migratory bird navigation.

Since the FSS is a binary problem that deals with selecting or discarding features, researchers have developed different binary metaheuristic algorithms to solve various real-world optimization problems [52]. For instance, the Binary Particle Swarm Optimization (BPSO) [53], Binary Gravitational Search Algorithm (BGSa) [54], Binary Differential Evolution (BDE) [55], Binary Bat Algorithm (BBA) [56], Binary Grey Wolf Optimization (bGWO) [57], Binary Dragonfly Algorithm (BDA) [58], Binary Salp Swarm Algorithm (BSSA) [59], and Binary Artificial Bee Colony (BABc) [60] are some well-known binary metaheuristic algorithms developed by applying different binarization methods including, transfer functions and crossover operator. Although metaheuristic algorithms have their benefits, the No-Free-Lunch (NFL) theorem [61] asserts that no individual algorithm can solve all problems [62, 63]. As a result, in FSS domain, new binary metaheuristic algorithms, such as Binary Arithmetic Optimization Algorithm (BAOA) [64], Binary Simulated Annealing-Based Dynamic Step Shuffled Frog Leaping Algorithm (BDSSRLFLA) [65], Binary Coronavirus Disease Optimization Algorithm (BCOVIDOA) [66], binary enhanced gaussian bare-bones grasshopper optimization [67], and Binary Quantum-based Avian Navigation Optimizer Algorithm (BQANA) [68] are still developing.

This paper aims to propose IBQANA, an improved binary quantum-based avian navigation optimizer algorithm, to select an effective feature subset from different medical datasets. First, we propose a binarization method by introducing a novel Hybrid Binary Operator (HBO) to map continuous values produced by the QANA into binary solutions. This method combines a threshold approach with Boolean operators, addressing the limitations of existing binarization techniques. Consequently,

the binary version of QANA, called BQANA-HBO, is developed, enabling more efficient and accurate binary solutions. Then, an improved version of BQANA-HBO, named IBQANA is developed to tackle the issues of local optima entrapment and low convergence rate. This improvement is achieved by introducing the Distance-Based Binary Search Strategy (DBSS), consisting of exploration and exploitation phases. In the exploration phase, which mostly happen in the early iterations, inferior search agents are updated by crossing over their positions with the farthest search agent's position in the archive. Conversely, in the final iterations, the exploitation phase is employed to update the inferiors' position by crossing over with the best position. It is expected that these contributions enhance the performance and effectiveness of IBQANA for FSS problems.

The efficacy of the proposed IBQANA was tested on 12 medical datasets with different numbers of features. First, the effectiveness of BQANA-HBO developed by the introduced binarization method is investigated and compared with BQANA developed using variable thresholding and five representatives of 20 binary versions of QANA developed by incorporating five distinct transfer function categories, including S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic transfer functions. The performance evaluation of IBQANA includes assessing its effectiveness by considering accuracy, fitness, and the selected features quantity. To determine its competitiveness, IBQANA is compared with seven well-known algorithms commonly used in the literature for FSS, including Binary Differential Evolution (BDE) [55], Binary Bat Algorithm (BBA) [56], V-shaped Binary Particle Swarm Optimization (VPSO) [69], hybrid Binary Particle Swarm Optimization and Gravitational Search Algorithm (BPSOGSA) [70], Binary Dragonfly Algorithm (BDA) [58], Quadratic Binary Harris Hawk Optimization (QBHHO) [71], and Improving Whale Optimization Algorithm for FSS with a Time-Varying transfer function (BWOA-TV) [72]. In addition, the proposed IBQANA is also compared with two other algorithms, namely the Binary Quantum-based Avian Navigation Optimizer Algorithm based on the thresholding approach (BQANA) [68] and the introduced BQANA-HBO. Moreover, a case study was conducted where IBQANA was employed as a diagnostic tool for Coronavirus Disease 2019 (COVID-19). To demonstrate the superiority of the proposed algorithm, the findings were statistically analyzed using the Friedman test. Based on the experimental and statistical results, it was discovered that the proposed IBQANA performs better than other comparative algorithms in identifying relevant features from COVID-19 and 11 other medical datasets.

## 2 Related Works

Due to the exponential growth in the number of potential solutions when dealing with datasets containing  $N$  features, evaluating all  $2^N$  possible solutions becomes an NP-hard problem. As a result, researchers have extensively addressed the problem of FSS by turning to metaheuristic algorithms, which have proven successful in approximating solutions. In this section, we review several prominent metaheuristic algorithms used in the wrapper approach, listed in the order of their appearance in the literature.

Kennedy and Eberhart [53] proposed a Binary version of the Particle Swarm Optimization (BPSO) algorithm using a sigmoid transfer function to tackle discrete optimization problems [73]. Sigmoid and its other three versions [69] are known as S-shaped transfer functions, frequently used in the literature. Marandi et al. [74] proposed Boolean Particle Swarm Optimization (BPSO) to solve a dual-band dual-polarized planar antenna design problem. This algorithm uses three Boolean operators, including and ( $\wedge$ ), or ( $\vee$ ), and xor ( $\oplus$ ), to generate binary solutions. Binary Differential Evolution (BDE), proposed by Gong et al. [55], utilizes discrete mutation and crossover operators to solve binary problems. Rashdi et al. [54] proposed the V-shaped transfer function and developed the Binary Gravitational Search Algorithm (BGSA) accordingly. Nakamura et al. [56] introduced the Binary Bat Algorithm (BBA) by mapping continuous search agents' velocity into binary solutions utilizing the S-shaped transfer function. Mirjalili et al. [69] proposed the VPSO algorithm as another binary version of PSO by applying four V-shaped transfer function family variants. The comparison between applying S-shaped and V-shaped transfer functions to the PSO algorithm indicates that the V-shaped transfer function provides superior results compared to the S-shaped transfer function. Mirjalili et al. [70] proposed a hybridization of binary PSO and GSA algorithms (BPSOGSA) to solve FSS problems by applying a V-shaped transfer function. Mirjalili [58] proposed a Dragonfly Algorithm (DA) inspired by the natural swarming patterns of dragonflies. Its binary version, named BDA was also developed using V-shaped transfer function.

Aslan et al. [75] introduced Jayax, a binary form of the Jaya algorithm [76] that generates binary solutions using the xor ( $\oplus$ ) Boolean operator. The results reported in this research show that the Jayax predominantly provides more optimal solutions than the binary form of the Jaya algorithm based on the transfer function. Jordehi [77] proposed three versions of the quadratic transfer function to develop the quadratic binary PSO for solving scheduling shiftable appliance problems in smart homes. The reported results show that the binary PSO developed using the S-shaped transfer function can generate better results than the V-shaped

transfer function. Furthermore, the introduced quadratic transfer function outperforms all other binary versions of PSO developed using different transfer functions. Too et al. [71] proposed the Quadratic Binary Harris Hawk Optimization (QBHHO) by introducing a new variant of the quadratic transfer function. The findings demonstrate that the introduced transfer function provides superior results for most investigated FSS problems. Sayed et al. [78] developed a Chaotic Crow Search Algorithm (CCSA) to tackle the drawbacks of the crow search algorithm in solving FSS problems, including entrapment in local optima and low convergence speed. CCSA employs ten chaotic maps to alleviate the mentioned flaws and boost its performance. The reported findings demonstrate that the sine map improves the CSA's performance in the FSS field. The authors in [79] proposed an effective binary version of PSO using the U-shaped transfer function (UBPSO). The results demonstrate that the UBPSO algorithm could generate superior binary solutions compared to the S-shaped and V-shaped binary PSO.

The Whale Optimization Algorithm (WOA) proposed by Mirjalili and Lewis [28] inspired by the hunting behavior of humpback whales is a well-known metaheuristic algorithm for finding the optimal solution with high speed using Simple but powerful search mechanisms [80]. To address the FSS problem, Mohammadzadeh and Gharehchopogh [81] introduced a hybridization of WOA and flower pollination algorithms based on opposition-based learning for email spam detection. In another study, Kahya et al. [72] developed a binary version of WOA, named BWOA-TV, incorporating a time-varying transfer function. Turkoglu et al. [82] proposed a binary artificial algae algorithm for FSS, inspired by the behavior of algae in nature. Piri et al. [83] developed a Discrete Artificial Gorilla Troop Optimization (DAGTO) algorithm to solve FSS problems in the healthcare sector. The authors implemented four variants of the DAGTO for different numbers and types of objective functions. Abualigah and Diabat [84] proposed a chaotic binary group search optimizer to solve the FSS problem by combining chaotic maps and a binary group search optimizer. Shaddeli et al. [85] introduced an improved African vulture optimization algorithm to alleviate convergence to local optima when solving discrete problems by hybridizing it with SCA and applying four strategies. Moreover, the authors used S-shaped and V-shaped transfer functions to map continuous solutions into binary ones.

Helmi et al. [86] proposed three binary versions of MPA using S-shaped and V-shaped transfer functions to address the problem of FSS in human activity recognition (HAR). HAR refers to identifying a person's actions based on measurements obtained from various mechanisms, including cameras, interior sensors, radars, wireless signals, and other sources [87]. A new method for diagnosing brain tumors was put out by Ren et al. [88], employing a step-by-step process based on a deep

learning-based Water Strider Algorithm (WSA). The proposed method involves feature extraction, FSS, and classification steps, using the WSA. Specifically, the WSA is employed to select the most relevant features that contribute to the classification of brain tumors. A Binary QANA (BQANA) algorithm was presented by Nadimi-Shahraki et al. [68] to solve the FSS problem in the medical data field using two different approaches: the first employs various transfer functions to convert the canonical QANA to binary, while the second maps continuous solutions to binary by setting a variable threshold for each dimension. The results revealed that BQANA developed using the threshold method generates better solutions than transfer functions. The results also demonstrated that  $S_4$ ,  $V_1$ ,  $U_4$ ,  $Z_3$ , and  $Q_3$  transfer functions are the representatives of their groups. In a similar study, Nadimi-Shahraki et al. [89] employed transfer functions and variable threshold approaches to develop Binary Starling Murmuration Optimizer (BMSO) by mapping continuous solutions of the Starling Murmuration Optimizer (SMO). Like the previous research, the results show that the variable threshold approach provides superior results.

### 3 Quantum-based Avian Navigation Optimizer Algorithm

The Quantum-based Avian Navigation Optimizer Algorithm (QANA) proposed by Zamani et al. [49] is a recent DE algorithm inspired by migrating birds' remarkable accuracy in long-distance aerial navigation. QANA utilizes multiple operators such as population partitioning, a qubit-crossover, two mutation strategies, and self-adaptive quantum orientation to achieve competitive results in continuous search spaces. Moreover, the V-echelon communication topology facilitates information sharing among search agents. Therefore, as seen in Algorithm 1, the QANA pseudocode includes four basic steps: initialization and multi-flock construction, movement strategy, fitness evaluation, and updating positions.

#### 3.1 Initialization and Multi-flock Construction

To distribute the entire population of birds represented by matrix  $A$  in Eq. (1), the population is divided into  $k$  different geographical areas using random centers. This partitioning creates  $k$  flocks, each consisting of  $n$  search factors, where  $n = N/k$ . As a result, the position of each bird in the  $D$ -dimensional problem space is denoted by vector  $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ .

$$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} \quad (1)$$



After the flock construction, in subsequent iterations, the search agents simultaneously use the knowledge shared by the V-echelon topology. Furthermore, utilizing the Success-based Population Distribution (SPD) policy, the search agents traverse the search space by using a mutation strategy rooted in quantum mechanics.

### 3.2 V-Echelon Communication Topology

The flight formation of migrating birds patterns the V-echelon communication construction and allows search agents of each flock to communicate knowledge that has been obtained. A Header (H) and two subsets for Left (L) and Right (R) lines make up this communication structure. The aerial navigation of migrating birds utilizing V-echelon topology is depicted in Fig. 1.

### 3.3 Quantum-based Navigation

Bird flocks utilize a quantum-based navigation system for exploring the search space. This system comprises a Success-based Population Distribution (SPD) policy, qubit-crossover, and two mutation strategies named "DE/quantum/I" and "DE/quantum/II". The assignment of a specific mutation strategy to a flock is based on the policy defined in Eq. (2),

$$SR_m(t) = \left( \left( \sum_{i \in f_m} \frac{\sum_{j=1}^n \tau_{ij}}{n} \right) / |f_m| \right) \times 100 \quad (2)$$

where  $SR_m$  denotes the success rate of mutation strategy  $M_m$ , and  $\tau_{ij}$  is equal to 1 if  $M_m$  improved  $a_j$  of the  $i$ -th flock; else,  $\tau_{ij}$  equals to 0.

The DE/quantum/I and DE/quantum/II are two quantum mutation strategies represented by mathematical Eqs. (3) and (4), respectively. These equations utilize several variables such as  $x_i(t)$  which refers to the  $i$ -th search agent,  $x_{Vechelon}(t)$  representing the position of the search agent followed by  $a_i$ ,  $x_j$  and  $x_j$  are two random positions selected from Long-Term

Memory ( $LTM$ ) and Short-Term Memory ( $STM$ ). In addition, the variable  $v_H(t+1)$  indicates the position of the header in the V-echelon topology, and is computed using Eq. (5). The quantum orientation for avian  $a_i$ , denoted by  $S_i$ , is also utilized in the computation and is presented in [49, 90].  $LB$  and  $UB$  are the search space's lower and upper boundaries.

$$\begin{aligned} v_i(t+1) &= x_{best}(t) + S_i(t) \\ &\times (x_{Vechelon}(t) - x_{j \in LTM}(t)) \\ &+ S_i(t) \times (x_{Vechelon}(t) - x_{best}(t)) \\ &+ S_i(t) \times (x_{j \in LTM}(t) - x_{j \in STM}(t)) \end{aligned} \quad (3)$$

$$\begin{aligned} v_i(t+1) &= S_i(t) \times (x_{best}(t) - x_{Vechelon}(t)) \\ &+ S_i(t) \times (x_i(t) - x_{j \in LTM}(t) - x_{j \in STM}(t)) \end{aligned} \quad (4)$$

$$v_H(t+1) = S_i(t) \times x_{best} + (LB + (UB - LB) \times rand(0, 1)) \quad (5)$$

By crossing the mutant vector  $v_i(t+1)$  with its parent  $x_i(t)$ , the trial vector  $u_i(t+1)$  is generated, where  $|\psi_i|_d$  is a qubit-crossover probability for the  $d$ -th dimension [49].

$$u_{id}(t+1) = \begin{cases} x_{id}(t+1), & |\psi_{id}| < rand \\ v_{id}(t+1), & |\psi_{id}| \geq rand \end{cases} \quad (6)$$

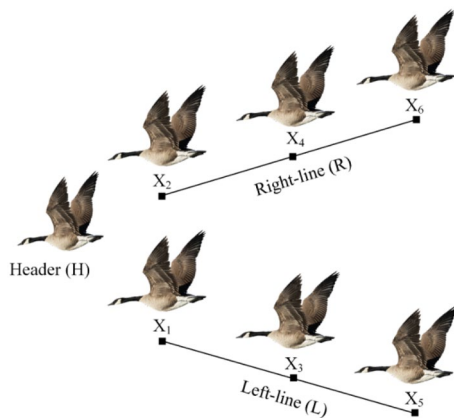


Fig. 1 The V-shaped formation

---

#### Algorithm 1. The pseudocode of QANA.

---

**Input:** Number of search agents ( $N$ ), Maximum Iterations ( $MaxIt$ ), Number of flocks ( $k$ ), and Dimension size ( $D$ ).

**Output:** The global best solution.

1. Begin
  2. Initialize the population
  3. Set  $t = 1$ .
  4. While  $t \leq MaxIt$
  5.   Constructing  $K$  flocks.
  6.   Calculating fitness value for each search agent.
  7.   Constructing  $LTM$  and  $STM$ .
  8.   Forming the V-echelon topology.
  9.   Assigning each flock to a mutation strategy using SPD policy based on Eq. (2).
  10.   For  $j = 1:K$
  11.     For  $i = 1:N/K$
  12.       Generating mutant vector  $v_i(t+1)$  for search agent  $a_i$  of flock  $f_j$  based on Eqs. (3–5).
  13.       Generating trial vector  $u_i(t+1)$  using Eq. (6).
  14.       If  $F(u_i(t+1)) \leq F(x_i(t))$
  15.          $x_i(t+1) = u_i(t+1)$
  16.       Else
  17.          $x_i(t+1) = x_i(t)$ .
  18.       End If
  19.     End For
  20.   End For
  21.   Updating the global best solution.
  22.    $t = t + 1$ .
  23. End while
-

### 4 Improved Binary Quantum-based Avian Navigation Optimizer Algorithm (IBQANA)

As stated earlier, ineffective binarization methods, prolonged convergence, and local optimum entrapment are the main problems of most binary metaheuristic algorithms that limit their performance in the FSS problem. First, this section introduces a novel binarization method, namely the Hybrid Binary Operator (HBO), to effectively map the continuous values into binary solutions and develop the binary version of the QANA, named BQANA-HBO, accordingly. Then, an improved version of BQANA-HBO, named IBQANA is proposed to cope with local optimum trapping and slow convergence rate by introducing a Distance-Based Binary Search Strategy (DBSS), which includes two different phases to adjust the position of the inferior search agents.

#### 4.1 Hybrid Binary Operator (HBO)

As stated in Sect. 2, although the most popular binarization technique for metaheuristic algorithms is the transfer function, it cannot produce efficient binary solutions for certain metaheuristic methods [68]. On the other hand, the thresholding method presented in Eq. (7) is more effective in developing a binary version of the QANA [68]; however, this method has a major drawback that hinders the metaheuristic algorithm from utilizing its full potential in finding the optimal solution.

$$b_i^d(t + 1) = \begin{cases} 1, & \text{if } x_i^d(t) > 0.5 \\ 0, & \text{if } x_i^d(t) \leq 0.5 \end{cases} \tag{7}$$

The issue lies in the thresholding method, which can only produce valid binary solutions for continuous values within the range of [0, 1]. Consequently, this approach fails to appropriately map continuous values outside these boundaries, such as negative values and values exceeding 1. Therefore, this section proposes a new binary method, Hybrid Binary Operator (HBO), for effectively mapping continuous solutions into binary space.

In the proposed HBO, if the continuous solution produced by the QANA is in the interval [0, 1], the thresholding method is used to convert it into a binary solution. However, if the generated continuous solution violates the lower bound (*lb*) or the upper bound (*ub*), according to its binary position in the current iteration, the logical operator AND ( $\wedge$ ) is used to update the binary solution based on Eq. (8).

$$b_i^d(t + 1) = \begin{cases} b \wedge b_i^d(t), & \text{if } x_i^d(t + 1) > ub \\ lb \wedge b_i^d(t), & \text{if } x_i^d(t + 1) < lb \end{cases} \tag{8}$$

The procedure of the HBO to determine the binary solution for each dimension of a search agent in the BQANA-HBO algorithm is depicted in Fig. 2.

Despite the effective binary mapping provided by HBO, the developed BQANA-HBO still cannot offer high-accuracy solutions with fast convergence. Therefore, in the following, we propose IBQANA to enhance the solution quality and convergence rate of the BQANA-HBO algorithm by introducing DBSS. This innovative combination is expected to yield more efficient and accurate results, making it a promising avenue for advancing the field.

#### 4.2 Distance-based Binary Search Strategy (DBSS)

This section proposes the IBQANA by introducing a DBSS to improve the performance of inferior search agents and speed up the convergence rate. The proposed search strategy consists of two phases of exploration and exploitation. In each iteration, if the new fitness of *i*-th search agent is equal or greater than its fitness in the current iteration, a new binary position is generated based on one of two different phases. As mentioned in Sect. 3, the QANA algorithm has *STM* and *LTM* to preserve the obtained solutions. In the proposed DBSS, two memories are merged and form an *Archive*. Then, according to the obtained value of *P* through Eq. (9), a new binary position is generated based on one of the exploitation and exploration phases,

$$P(t) = 1 - \frac{t}{MaxIt} \tag{9}$$

where *t* represents the present iteration, and *MaxIt* is the maximum number of iterations. According to this equation, the value of *P* gradually decreases from 1 to 0 as the number of iterations progresses.

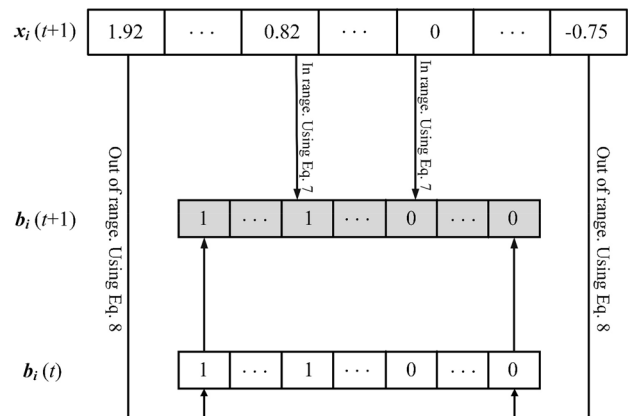


Fig. 2 Process of updating the binary positions using the HBO

#### 4.2.1 Exploration Phase

If a random number between 0 and 1 is determined to be bigger than the  $P$  value in the current iteration, a new binary solution is generated based on the uniform crossover operator according to Eq. (10),

$$b_{\text{new}}^d(t+1) = \begin{cases} \text{Archive}_{\text{far}}^d(t), & \text{if } r_1^d \geq r_2^d \\ b_i^d(t+1), & \text{otherwise} \end{cases} \quad (10)$$

where  $r_1$  and  $r_2$  are random numbers within the range of (0,1), and  $\text{Archive}_{\text{far}}^d(t)$  indicates the binary position of the farthest member of the *Archive* from the position of  $\mathbf{b}_i(t+1)$ , which is determined based on Eq. (11),

$$\text{Archive}_{\text{far}}^d(t) = \max(\text{HD}_{ik}(t)) \quad (11)$$

where  $\text{HD}_{ik}(t)$  is the Hamming distance between  $\mathbf{b}_i(t+1)$  and the archive members, the Hamming distance is defined in Eq. (12), where  $M$  represents the total number of positions stored in the archive.

$$\text{HD}_{ik}(t) = \sum_{d=1}^D (b_i^d(t+1) - \text{Archive}_k^d(t)) \quad (12)$$

$$i = [1, 2, \dots, N], k = [1, 2, \dots, M]$$

#### 4.2.2 Exploitation Phase

Suppose the  $P$  value is determined to be less than or equal to a randomly generated number in the range (0, 1). In that case, the new binary solution is generated using the exploitation phase. This phase enhances the solutions' quality without degrading the algorithm's exploration capability. To achieve this, a fitness-dependent weight factor is calculated based on Eq. (13) [91],

$$\omega(t) = \frac{F(b_{\text{best}}(t+1))}{F(b_i(t+1))} \quad (13)$$

where  $F(\cdot)$  denotes the fitness value of a binary position, since according to the fitness function definition presented in the next section, the FSS is a minimization problem; the value produced by this equation is always in the range [0, 1]. Therefore, Eq. (14) generates new binary position during the exploitation phase.

$$b_{\text{new}}^d(t+1) = \begin{cases} b_{\text{best}}^d(t), & \text{if } \omega^d \geq r_3^d \\ b_i^d(t+1), & \text{otherwise} \end{cases} \quad (14)$$

Finally, after calculating the fitness of the new position produced by each of the two introduced phases, a greedy selection between  $\mathbf{b}_{\text{new}}(t+1)$  and  $\mathbf{b}_i(t+1)$  determines the value of  $\mathbf{b}_i(t+1)$ . The pseudocode of the proposed

DBSS is presented in Algorithm 2, and the pseudocode of the proposed Improved Binary Quantum-based Avian Navigation Optimizer Algorithm (IBQANA) is shown in Algorithm 3.

---

**Algorithm 2.** The pseudocode of the proposed DBSS.

---

**Input:**  $\mathbf{b}_i(t)$ ,  $\mathbf{b}_i(t+1)$ ,  $F(\mathbf{b}_i(t+1))$ ,  $LTM(t)$ ,  $STM(t)$ ,  $P(t)$ ,  $\mathbf{b}_{\text{best}}(t+1)$ .

**Output:**  $\mathbf{b}_i(t+1)$ .

1. Begin
  2.  $\text{Archive}(t) = \text{Merge } STM \text{ and } LTM$ .
  3. If  $P(t) > \text{rand}$
  4. For  $j = 1:D$
  5. Computing the HD between  $\mathbf{b}_i(t+1)$  and members of *Archive* ( $t$ ) based on Eq. (12).
  6. Crossover  $\mathbf{b}_i(t+1)$  and the farthest archived member using Eq. (10) to generate  $\mathbf{b}_{\text{new}}$ .
  7. End For
  8. Else
  9. For  $j = 1:D$
  10. Computing  $w$  using Eq. (13).
  11. Crossover  $\mathbf{b}_i(t+1)$  and  $\mathbf{b}_{\text{best}}(t+1)$  using Eq. (14) to generate  $\mathbf{b}_{\text{new}}(t+1)$ .
  12. End For
  13. End If
  14. Computing fitness value of  $\mathbf{b}_{\text{new}}(t+1)$ .
  15. If  $F(\mathbf{b}_{\text{new}}(t+1)) \leq F(\mathbf{b}_i(t+1))$
  16.  $F(\mathbf{b}_i(t+1)) = F(\mathbf{b}_{\text{new}}(t+1))$ .
  17.  $\mathbf{b}_i(t+1) = \mathbf{b}_{\text{new}}(t+1)$ .
  18. End If
- 

---

**Algorithm 3.** The pseudocode of the proposed IBQANA.

---

**Input:** Maximum Iterations ( $MaxIt$ ), Number of search agents ( $N$ ), Number of flocks ( $k$ ), and Dimension size ( $D$ ).

**Output:** The global best solution.

1. Begin
  2. Initialize the population
  3. Set  $t = 1$ .
  4. While  $t \leq MaxIt$
  5. Constructing  $K$  flocks.
  6. Calculating fitness value for each search agent.
  7. Constructing  $LTM$  and  $STM$ .
  8. Forming the V-echelon topology.
  9. Assigning each flock to a mutation strategy using SPD policy based on Eq. (2).
  10. For  $j = 1:K$
  11. For  $i = 1:N/K$
  12. Generating mutant vector  $\mathbf{v}_i(t+1)$  for  $\mathbf{a}_i$  of flock  $\mathbf{f}_j$  based on Eqs. (3–5).
  13. Generating trial vector  $\mathbf{u}(t+1)$  using Eq. (6).
  14. Converting  $\mathbf{u}_i(t+1)$  to a binary vector  $\mathbf{b}_i(t+1)$  using Eqs. (7) and (8).
  15. Calculating fitness of  $\mathbf{b}_i(t+1)$ .
  16. If  $F(\mathbf{b}_i(t+1)) \geq F(\mathbf{b}_i(t))$
  17. Updating the position of  $\mathbf{b}_i(t+1)$  using Algorithm 2.
  18. End If
  19. End For
  20. End For
  21. Updating the global best solution.
  22.  $t = t + 1$ .
  23. End while
- 

### 4.3 Computational and Time Complexity

As shown in Algorithm 3, the proposed IBQANA consists of four phases: initialization, multi-flock construction, movement, and fitness evaluation. In the initialization phase (line

2),  $N$  search agents are randomly distributed into a  $D$ -dimensional search space with an  $O(ND)$  computational complexity. In the multi-flock construction phase (line 5),  $k$  flocks with population size  $n$  (where  $n = N/k$ ) are constructed with the computational complexity of  $O(kn)$ . Then, the fitness value of each search agent is computed with the computational complexity of  $O(ND)$ . The movement phase consists of memory construction, V-echelon topology formation, and three types of movement vectors: mutant, trial, and binary. The long-term and short-term memories with  $K'$  and  $K''$  capacities are constructed with an  $O(K'D) + O(K''D)$  computational complexity. Then, the V-echelon topology (line 8) for each  $k$  flock is formed with an  $O(kn)$  complexity. Finally, line 9 computes the SPD policy with an  $O(kn)$  complexity to assign mutant vectors to each flock. Mutant vectors (line 12) defined in Eqs. (3–5) are computed with  $O(ND)$  complexity, and the trial vectors and binary vector (lines 13 and 14) defined in Eqs. (6–8) are calculated with complexity of  $O(2ND)$ . In the fitness evaluation phase (line 15), the fitness value of each search agent is computed with the computational complexity of  $O(ND)$ , and Algorithm 2 runs with a computational complexity of  $O(ND)$ . The algorithm repeats these phases until it reaches the maximum iteration  $T$ . Thus, the total computational complexity of IBQANA can be  $O(ND + T(kn + ND + (K'D + K''D) + kn + kn + ND + 2ND + ND + ND))$ . Since  $kn = N$ , we can simplify the computational complexity of IBQANA to  $O(ND + T(3N + 5ND + (K'D + K''D)))$ . The computational complexity can be equal to  $O(TND)$ , because  $N$  is always bigger than  $K'$  and  $K''$ .

To compute the time complexity, the Average Time ( $AT$ ) required to find the solution, known as the run time, is computed using Eq. (15) [49, 92]. This equation considers the total number of runs ( $M$ ) and the computational time of algorithm  $A$  for each run ( $RT_{A,i}$ ).

$$AT_A = \frac{1}{M} \sum_{i=1}^M RT_{A,i} \quad (15)$$

Table 1 compares the run times (in seconds) of the IBQANA algorithm and comparative algorithms on different datasets. It can be seen that, while it may not be the fastest algorithm, it consistently has an acceptable run time across most datasets. However, it's worth noting that using IBQANA for real-time applications might be a limitation due to its relatively longer run times in certain datasets. Nevertheless, IBQANA proves to be a promising candidate for accuracy-critical applications, particularly in medical datasets, making it a promising choice for researchers and practitioners in this domain.

## 5 Experimental Assessment

This section evaluates the IBQANA algorithm's performance through experiments to tackle the FSS problem on 12 medical datasets with various feature sizes. First, to evaluate the effectiveness of the suggested binarization technique, the Hybrid Binary Operator (HBO) is applied to develop the binary QANA (BQANA-HBO) algorithm and compare its performance with the binary QANA developed using thresholding method, and representatives of 20 binary versions of QANA developed using five groups of transfer functions. In the second experiment, we evaluate the improved version of this algorithm called IBQANA and contrast the obtained results with BQANA-HBO, as well as seven other comparative algorithms, including BBA [56], VPSO [69], BPSOGSA [70], QBHHO [71], BDA [58], BWOA-TV [72], and BQANA [68]. These algorithms encompass highly cited, recent, and diverse approaches, allowing for a comprehensive comparison.

**Table 1** Run times of IBQANA and comparative algorithms for different datasets

Datasets	BDE	BBA	VPSO	BPSOGSA	BDA	QBHHO	BWOA-TV2	BQANA-HBO	IBQANA
Diabetes	228.71	192.23	325.14	315.83	313.98	722.32	280.89	356.04	531.87
Heart	211.50	138.77	305.35	1202.50	299.87	657.31	294.54	382.46	577.41
Hepatitis	127.44	136.85	364.33	377.75	277.61	511.56	273.59	349.55	516.37
Lymphography	191.82	200.85	310.30	317.58	374.62	608.53	431.12	392.69	669.24
SPECT	234.79	191.67	294.90	299.09	269.56	623.83	336.92	355.57	568.02
WBCD	147.93	234.68	286.84	299.97	235.68	608.38	272.97	382.46	696.49
LSVT	1649.80	283.82	340.28	394.37	236.99	580.31	268.67	342.02	606.52
Parkinson	578.99	460.48	655.25	503.13	542.52	1159.90	452.18	530.73	838.72
Colon	252.61	147.31	361.11	333.09	398.17	668.22	285.93	325.98	711.72
SRBCT	302.10	1169.10	376.33	367.96	419.11	736.59	446.87	378.46	526.11
Leukemia	496.74	370.78	499.84	481.91	677.64	922.33	468.28	501.57	791.43
Prostate tumor	687.45	315.02	625.87	543.36	1187.20	1070.10	5263.80	563.14	966.49



## 5.1 Parameter Settings

To verify that the comparisons are accurate and fair, the parameters for the comparative algorithms were set to the values stated in their respective original publications, which are provided in Table 2, while the initial values for the shared parameters, such as maximum number of iterations (*MaxIt*) and population size (*N*), were set to 20 and 300, respectively, for all algorithms. In this study, all experiments, including those for the other algorithms used in the comparison, were conducted 20 times independently on a laptop with an Intel Core i7-10750H CPU and 24.0 GB of RAM using MATLAB R2022a.

In this work, the K-Nearest Neighbor (K-NN) classifier with Euclidean distance and  $K=5$  is used to determine the classification accuracy of produced feature subsets based on Eq. (16),

$$Accuracy = \frac{(TP + TN)}{(TP + FN + TN + FP)} \quad (16)$$

where TP represents the count of accurately identified positive samples, while TN refers to the count of correctly classified negative samples. FN indicates the count of positive samples incorrectly classified as negative, and FP represents the number of negative samples incorrectly classified as positive. Also, the Classification Error (CE) is calculated using Eq. (17).

$$CE = 1 - Accuracy \quad (17)$$

The goals of this study, including maximization of the classification accuracy and minimization of the number of features, are aggregated in an objective function given in Eq. (18),

$$Fitness = \alpha \times CE + (1 - \alpha) \frac{N_{sf}}{N_{tf}} \quad (18)$$

where  $\alpha$  denotes the significance of classification accuracy,  $N_{tf}$ , and  $N_{sf}$  are the total number of features of the dataset and the number of selected features, respectively. Since

classification accuracy is the most crucial metric for medical datasets, following related studies [57, 71], we considered  $\alpha=0.99$ .

## 5.2 Description of Medical Datasets

This research evaluates the effectiveness of IBQANA and competing algorithms in selecting effective features using 12 medical benchmark datasets, primarily from the UCI machine learning repository [93]. Table 3 presents statistical information on the datasets under consideration. The K-fold cross-validation method was employed to prevent overfitting issues, with  $k_{fold}=10$ . This method divides the dataset into  $k$  folds, with the classifier utilizing  $k-1$  folds as training sets and one fold as the testing set.

## 5.3 Evaluation of the Proposed Hybrid Binary Operator (HBO)

This section investigates the effectiveness of the introduced HBO binarization method on the developed binary QANA (BQANA-HBO) to select effective features from the medical datasets and compare the obtained results with the results obtained from the binary QANA using the thresholding method (BQANA) and five representatives of S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic transfer functions [68]. The results are tabulated in Table 4, where the best average fitness value achieved for each medical dataset is emphasized in bold font. Also, the last row shows the average of Friedman test ranking results for each algorithm.

The reported results in Table 4 signify that the introduced HBO provides the superior mapping of continuous values of QANA into binary solutions in contrast to other binarization methods in terms of average fitness value. The findings also reveal that the binary QANA developed using the thresholding method (BQANA) provides better solutions than the investigated transfer functions and is ranked second. Among the investigated transfer functions,  $Q_3$  provides competitive results. Conversely, it was found that the S-shaped transfer function yields the poorest performance when used to

**Table 2** Parameters of the algorithms

Algorithm	Parameter settings
BBA	$A=0.9, r=0.9, Q_{min}=0, Q_{max}=2$
VPSO	$c_1=c_2=2, w$ is linearly decreased from 0.9 to 0.4
BPSOGSA	$c'_1=(-2r^3/MaxIt^3)+2, c'_2=(2r^3/MaxIt^3), G_0=1, \alpha=20$
QBHHO	$\beta=1.5, Q_4$ transfer function, and $x_{max}=5$
BDA	$D_{max}=6$
BWOA-TV	Time-varying transfer function = $TV_2, a$ is linearly decreased from 2 to 0
BQANA	The number of flocks ( $k$ )=2, LTM size ( $K'$ )=2, and STM size ( $K''$ )=10
IBQANA	$k=2, K'=2, K''=10, P$ is linearly decreased from 1 to 0

**Table 3** The description of the medical datasets

No	Medical datasets	No. samples	No. features	Classes	Missing values
1	Pima Diabetes	768	8	2	Yes
2	Statlog (Heart)	270	13	2	No
3	Hepatitis	155	19	2	Yes
4	Lymphography	148	18	4	No
5	Single-photon emission computed tomography (SPECT)	267	22	2	No
6	Wisconsin breast cancer diagnosis (WBCD)	569	30	2	No
7	LSVT voice rehabilitation	126	309	2	No
8	Parkinson	756	754	2	No
9	Colon	62	2000	2	No
10	Small round blue-cell tumor (SRBCT)	83	2308	4	No
11	Leukemia	72	7129	4	No
12	Prostate tumor	102	10,509	2	No

convert continuous QANA values into binary solutions to select effective features from medical datasets.

#### 5.4 Evaluation of the Proposed IBQANA

This section undertakes an evaluation and investigation of the efficacy of the proposed IBQANA on 12 medical datasets, followed by a comparison of the results with those generated by seven well-established metaheuristic algorithms. The outcomes are comprehensively presented in Tables 5 and 6, providing details on the average (Avg), minimum (Min), standard deviation (Std), or maximum (Max) fitness and classification accuracy.

Table 5 compares the fitness values of investigated FSS algorithms on 12 medical datasets, where the bold values represent the best average fitness value achieved for each medical dataset. The fitness function, described in Eq. (18), considers classification accuracy and the number of selected features. IBQANA consistently outperforms other algorithms in fitness values, with the lowest fitness value and standard deviation among all comparatives. Comparing IBQANA and BQANA-HBO highlights the effect of the introduced DBSS to make IBQANA a promising choice for FSS in medical datasets. It is also noticeable that the BQANA-HBO ranks second, demonstrating an advantage over competitors for one medical dataset. The achievements of IBQANA are mostly because of its strengths in the effective mapping of continuous solutions to binary ones using the introduced HBO and updating inferiors' position using the introduced DBSS.

From the results tabulated in Table 6, it can be observed that IBQANA outperforms other algorithms in terms of average classification accuracy on most datasets. For example, on the Pima dataset, IBQANA achieved an average accuracy of 77.459, which is higher than the average accuracy of other algorithms. Similar trends are observed on different datasets,

such as Heart, Hepatitis, Lymphography, SPECT, WBCD, LSVT, Parkinson, Colon, and SRBCT, where IBQANA consistently demonstrates competitive or superior performance compared to other algorithms.

Figure 3 visually depicts the comparative convergence behavior analysis between IBQANA and other relevant algorithms for FSS purposes in the context of 12 medical datasets. The average of the fitness function in 20 separate runs has been used to plot the convergence curves. The plotted curves demonstrate that the proposed IBQANA has the quickest convergence towards optimal solutions compared to other algorithms for most datasets. Also, by comparing the performance of IBQANA against other algorithms on colon, leukemia, and prostate tumor datasets, it can be concluded that as the quantity of features grows, the proposed algorithm maintains its scalability and reaches better solutions than the comparative algorithms. The ability of the IBQANA to bypass local optimum solutions can be seen in the curves related to diabetes, lymphography, WBCD, colon, leukemia, and prostate tumor datasets. It is also noticeable that the IBQANA demonstrates significant advantages over BQANA-HBO primarily due to its innovative approach of updating the positions of inferior search agents using the introduced DBSS. This improvement empowers IBQANA to achieve accelerated convergence and effectively evade being trapped in local optima in various scenarios.

Since in medical datasets, the accuracy of classification is considered the most crucial criterion, the classification accuracy results of IBQANA and comparative algorithms are shown in the form of a boxplot in Fig. 4. In general, the plots demonstrate that the proposed IBQANA can enhance the classification accuracy by selecting effective features and discarding irrelevant ones. The plots indicate that the IBQANA achieved higher median classification accuracy for eight datasets and higher maximum accuracy rate for six datasets. It can also be noticed that the BQANA-HBO

**Table 4** The comparison between the BQANA-HBO and representatives of each transfer function family

Datasets	Metrics		$S_4$	$V_1$	$U_4$	$Z_3$	$Q_3$	BQANA	BQANA-HBO
Pima	Fitness	Avg	0.2386	0.2386	0.2393	0.2345	0.2380	0.2316	<b>0.2292</b>
		Min	0.2343	0.2330	0.2318	0.2319	0.2319	0.2291	0.2228
	Accuracy	Avg	76.505	76.462	76.397	76.955	76.500	77.208	77.380
		Max	76.962	77.093	77.093	77.228	77.076	77.486	77.997
HeartEW	Fitness	Avg	0.1493	0.1518	0.1493	0.1435	0.1509	0.1384	<b>0.1394</b>
		Min	0.1395	0.1411	0.1432	0.1387	0.1416	0.1329	0.1358
	Accuracy	Avg	85.333	85.093	85.278	85.963	85.074	86.426	86.278
		Max	86.296	86.296	85.926	86.296	85.926	87.037	86.667
Hepatitis	Fitness	Avg	0.1243	0.1227	0.1223	0.1225	0.1231	0.1051	<b>0.1033</b>
		Min	0.1132	0.1118	0.1056	0.0996	0.1061	0.0981	0.0934
	Accuracy	Avg	87.871	88.008	88.038	88.129	87.917	89.775	89.946
		Max	89.042	89.083	89.708	90.417	89.708	90.500	91.042
Lymphography	Fitness	Avg	0.1434	0.1422	0.1423	0.1313	0.1432	<b>0.1128</b>	0.1162
		Min	0.1324	0.1263	0.1258	0.1132	0.1303	0.1008	0.0974
	Accuracy	Avg	86.060	86.152	86.169	87.407	85.995	89.131	88.736
		Max	87.191	87.857	87.857	89.238	87.286	90.381	90.667
SPECT Heart	Fitness	Avg	0.2503	0.2468	0.2468	0.2416	0.2453	0.2231	<b>0.2216</b>
		Min	0.2415	0.2225	0.2260	0.2167	0.2326	0.2038	0.2080
	Accuracy	Avg	75.171	75.496	75.462	76.120	75.565	77.873	77.949
		Max	76.097	77.935	77.536	78.618	76.823	79.786	79.402
WBCD	Fitness	Avg	0.0501	0.0494	0.0490	0.0488	0.0480	0.0463	<b>0.0455</b>
		Min	0.0475	0.0477	0.0469	0.0459	0.0451	0.0427	0.0438
	Accuracy	Avg	95.389	95.404	95.356	95.617	95.415	95.550	95.515
		Max	95.611	95.602	95.598	95.777	95.777	95.952	95.783
LSVT	Fitness	Avg	0.2947	0.2896	0.2602	0.2824	0.2790	0.1915	<b>0.1462</b>
		Min	0.2789	0.2796	0.2219	0.2321	0.2471	0.1169	0.0852
	Accuracy	Avg	70.776	71.196	73.894	72.038	72.157	80.881	85.301
		Max	72.372	72.244	77.628	77.051	75.513	88.205	91.410
Parkinson	Fitness	Avg	0.2341	0.2115	0.2011	0.2355	0.2049	<b>0.1604</b>	0.1647
		Min	0.2160	0.1805	0.1865	0.1931	0.1838	0.1279	0.1235
	Accuracy	Avg	76.883	78.956	79.947	76.800	79.607	83.976	83.551
		Max	78.702	82.000	81.488	80.958	81.746	87.181	87.558
Colon	Fitness	Avg	0.1012	0.1008	0.0995	0.1019	0.0947	0.0775	<b>0.0686</b>
		Min	0.0949	0.0849	0.0916	0.0827	0.0809	0.0481	0.0468
	Accuracy	Avg	90.321	90.298	90.417	90.345	90.798	92.500	93.298
		Max	90.952	91.905	91.190	92.143	92.143	95.238	95.476
SRBCT	Fitness	Avg	0.0129	0.0135	0.0101	0.0132	0.0047	0.0040	<b>0.0007</b>
		Min	0.0053	0.0031	0.0021	0.0041	0.0018	0.0003	0.0002
	Accuracy	Avg	99.236	99.069	99.382	99.243	99.833	99.833	100.000
		Max	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg	0.1010	0.0974	0.0960	0.0984	0.0915	0.0638	<b>0.0548</b>
		Min	0.0884	0.0837	0.0690	0.0716	0.0702	0.0426	0.0254
	Accuracy	Avg	90.339	90.607	90.714	90.696	91.071	93.920	94.705
		Max	91.607	91.786	93.393	93.214	93.214	96.250	97.500
Prostate Tumor	Fitness	Avg	0.1219	0.1206	0.1147	0.1146	0.1113	0.0534	<b>0.0476</b>
		Min	0.1044	0.1030	0.0984	0.1004	0.0882	0.0199	0.0200
	Accuracy	Avg	88.232	88.291	88.714	89.036	89.082	94.777	95.305
		Max	90.000	90.091	90.091	90.364	91.273	98.000	98.000
Friedman test			5.86	5.67	4.63	4.54	4.22	1.74	<b>1.35</b>

**Table 5** Comparison in terms of fitness values of algorithms

Datasets	Metrics	BDE	BBA	VPSO	BPSOGSA	BDA	QBHHO	BWOA-TV2	BQANA-HBO	IBQANA
Pima	Avg	0.2324	0.2350	0.2308	0.2314	0.2317	0.2337	0.2388	0.2292	<b>0.2286</b>
	Std	0.0022	0.0022	0.0032	0.0035	0.0018	0.0033	0.0037	0.0028	0.0023
	Min	0.2292	0.2318	0.2216	0.2229	0.2280	0.2266	0.2280	0.2228	0.2228
Heart	Avg	0.1390	0.1416	0.1396	0.1379	0.1386	0.1417	0.1524	0.1394	<b>0.1378</b>
	Std	0.0056	0.0037	0.0027	0.0018	0.0030	0.0031	0.0040	0.0024	0.0016
	Min	0.1308	0.1366	0.1345	0.1351	0.1308	0.1351	0.1468	0.1358	0.1358
Hepatitis	Avg	0.1177	0.1146	0.0999	0.1027	0.1013	0.1108	0.1246	0.1033	<b>0.0983</b>
	Std	0.0133	0.0093	0.0070	0.0087	0.0074	0.0051	0.0078	0.0056	0.0063
	Min	0.0931	0.1043	0.0876	0.0924	0.0924	0.0998	0.1056	0.0934	0.0924
Lymphography	Avg	0.1246	0.1547	0.1138	0.1136	0.1154	0.1266	0.1420	0.1162	<b>0.1061</b>
	Std	0.0151	0.0119	0.0074	0.0110	0.0082	0.0057	0.0087	0.0124	0.0073
	Min	0.1050	0.1372	0.1044	0.0990	0.1054	0.1130	0.1254	0.0974	0.0979
SPECT	Avg	0.2298	0.2360	0.2200	0.2161	0.2230	0.2308	0.2448	0.2216	<b>0.2116</b>
	Std	0.0103	0.0116	0.0098	0.0083	0.0129	0.0066	0.0072	0.0066	0.0081
	Min	0.1994	0.2181	0.2081	0.2049	0.2012	0.2150	0.2266	0.2080	0.1994
WBCD	Avg	0.0506	0.0481	0.0463	0.0461	0.0464	0.0459	0.0497	0.0455	<b>0.0449</b>
	Std	0.0045	0.0022	0.0011	0.0011	0.0015	0.0004	0.0013	0.0007	0.0010
	Min	0.0465	0.0450	0.0437	0.0444	0.0434	0.0448	0.0471	0.0438	0.0431
LSVT	Avg	0.2845	0.2864	0.2730	0.2765	0.2803	0.1515	0.2524	<b>0.1462</b>	0.1939
	Std	0.0087	0.0173	0.0258	0.0203	0.0155	0.0184	0.0301	0.0412	0.0540
	Min	0.2647	0.2458	0.2182	0.2217	0.2249	0.1254	0.1637	0.0852	0.1073
Parkinson	Avg	0.2512	0.2238	0.1972	0.2168	0.1938	0.1673	0.2156	0.1647	<b>0.1498</b>
	Std	0.0038	0.0231	0.0373	0.0354	0.0388	0.0075	0.0159	0.0322	0.0114
	Min	0.2429	0.1896	0.1554	0.1689	0.1549	0.1546	0.1778	0.1235	0.1219
Colon	Avg	0.1020	0.0954	0.0848	0.0907	0.0925	0.0821	0.0983	0.0686	<b>0.0599</b>
	Std	0.0071	0.0073	0.0070	0.0062	0.0087	0.0099	0.0043	0.0134	0.0065
	Min	0.0882	0.0804	0.0778	0.0710	0.0776	0.0501	0.0816	0.0468	0.0458
SRBCT	Avg	0.0097	0.0075	0.0046	0.0052	0.0052	0.0006	0.0147	0.0007	<b>0.0003</b>
	Std	0.0051	0.0082	0.0001	0.0025	0.0029	0.0003	0.0035	0.0007	0.0001
	Min	0.0057	0.0038	0.0044	0.0045	0.0039	0.0002	0.0046	0.0002	0.0001
Leukemia	Avg	0.0937	0.0943	0.0773	0.0775	0.0768	0.0706	0.1028	0.0548	<b>0.0406</b>
	Std	0.0076	0.0099	0.0118	0.0100	0.0094	0.0081	0.0064	0.0188	0.0104
	Min	0.0766	0.0740	0.0579	0.0668	0.0560	0.0560	0.0879	0.0254	0.0259
Prostate Tumor	Avg	0.1024	0.1143	0.0987	0.0920	0.1009	0.0561	0.1200	0.0476	<b>0.0386</b>
	Std	0.0082	0.0132	0.0099	0.0105	0.0143	0.0176	0.0066	0.0157	0.0170
	Min	0.0828	0.0895	0.0833	0.0643	0.0806	0.0289	0.1012	0.0200	0.0001
Friedman		7.16	7.41	4.39	4.43	4.86	4.54	8.05	2.83	<b>1.33</b>

represents the potential for increasing classification accuracy by selecting effective features as a low computational cost algorithm.

Given that a primary objective of FSS is to minimize the number of features utilized, Fig. 5 presents a comparison of the algorithms based on the rate of feature reduction achieved for each medical dataset, as determined through Eq. (19). In this equation,  $N_{sf}$  denotes the number of features selected by the algorithm, and  $N_{f}$  represents the total number of features in the dataset of interest.

$$R(\%) = 1 - \left( \frac{N_{sf}}{N_f} \right) \times 100 \tag{19}$$

The results in Fig. 5 demonstrate that the BDE algorithm has the weakest performance in minimizing the number of features in all 12 datasets. In contrast, QBHHO, BQANA-HBO, and IBQANA have performed best in reducing the number of features. Although QBHHO predominantly minimizes the number of features more than IBQANA, it should be noticed that discarding effective



**Table 6** Comparison in terms of classification accuracy of algorithms

Datasets	Metrics	BDE	BBA	VPSO	BPSOGSA	BDA	QBHHO	BWOA-TV2	BQANA-HBO	IBQANA
Pima	Avg	77.210	76.827	77.278	77.249	77.159	76.933	76.493	77.380	77.459
	Std	0.182	0.239	0.301	0.314	0.152	0.310	0.338	0.280	0.241
	Max	77.481	77.216	78.122	77.987	77.474	77.614	77.471	77.997	78.124
Heart	Avg	86.537	86.111	86.296	86.481	86.426	86.037	85.019	86.278	86.444
	Std	0.514	0.407	0.360	0.190	0.346	0.342	0.407	0.306	0.186
	Max	87.407	86.667	87.037	86.667	87.407	86.667	85.556	86.667	86.667
Hepatitis	Avg	88.700	88.835	90.317	90.035	90.179	89.146	87.810	89.946	90.454
	Std	1.311	0.930	0.719	0.867	0.818	0.540	0.764	0.608	0.648
	Max	91.125	89.833	91.625	91.042	91.083	90.292	89.708	91.042	91.042
Lymphography	Avg	88.100	84.855	89.031	89.069	88.871	87.650	86.171	88.736	89.779
	Std	1.514	1.210	0.774	1.134	0.852	0.598	0.864	1.321	0.780
	Max	90.000	86.476	89.905	90.619	89.952	89.095	87.952	90.667	90.619
SPECT	Avg	77.344	76.555	78.179	78.584	77.900	76.971	75.677	77.949	78.988
	Std	1.004	1.130	0.989	0.848	1.304	0.669	0.706	0.734	0.854
	Max	80.228	78.248	79.387	79.772	80.185	78.647	77.521	79.402	80.228
WBCD	Avg	95.505	95.582	95.687	95.724	95.654	95.462	95.369	95.515	95.620
	Std	0.420	0.236	0.154	0.171	0.193	0.064	0.181	0.141	0.173
	Max	95.959	95.962	95.956	95.959	95.962	95.611	95.774	95.783	95.952
LSVT	Avg	72.051	71.538	72.920	72.564	72.176	84.724	74.571	85.301	80.599
	Std	0.877	1.719	2.598	2.052	1.564	1.851	3.020	4.094	5.346
	Max	74.103	75.513	78.462	78.077	77.756	87.372	83.526	91.410	89.167
Parkinson	Avg	75.354	77.886	80.569	78.607	80.902	83.154	78.427	83.551	85.019
	Std	0.429	0.845	0.929	0.91345	0.91143	0.91929	0.90548	0.93298	0.94131
	Max	76.323	0.739	0.703	0.619	0.876	0.958	0.417	1.268	0.638
Colon	Avg	90.536	90.845	91.929	91.345	91.143	91.929	90.548	93.298	94.131
	Std	0.720	0.739	0.703	0.619	0.876	0.958	0.417	1.268	0.638
	Max	91.905	92.381	92.619	93.333	92.619	95.000	92.143	95.476	95.476
SRBCT	Avg	99.715	99.715	100.00	99.944	99.938	100.00	98.979	100.00	100.00
	Std	0.507	0.824	0.000	0.248	0.280	0.000	0.353	0.000	0.000
	Max	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Leukemia	Avg	91.304	90.964	92.688	92.679	92.732	93.018	90.089	94.705	96.080
	Std	0.762	0.989	1.188	1.005	0.950	0.794	0.652	1.743	1.042
	Max	93.036	93.036	94.643	93.750	94.821	94.464	91.607	97.500	97.500
Prostate Tumor	Avg	90.486	88.936	90.536	91.209	90.300	94.350	88.341	95.305	96.214
	Std	0.795	1.326	1.003	1.062	1.437	1.775	0.667	1.478	1.670
	Max	92.273	91.455	92.091	94.000	92.364	97.091	90.273	98.000	100.00

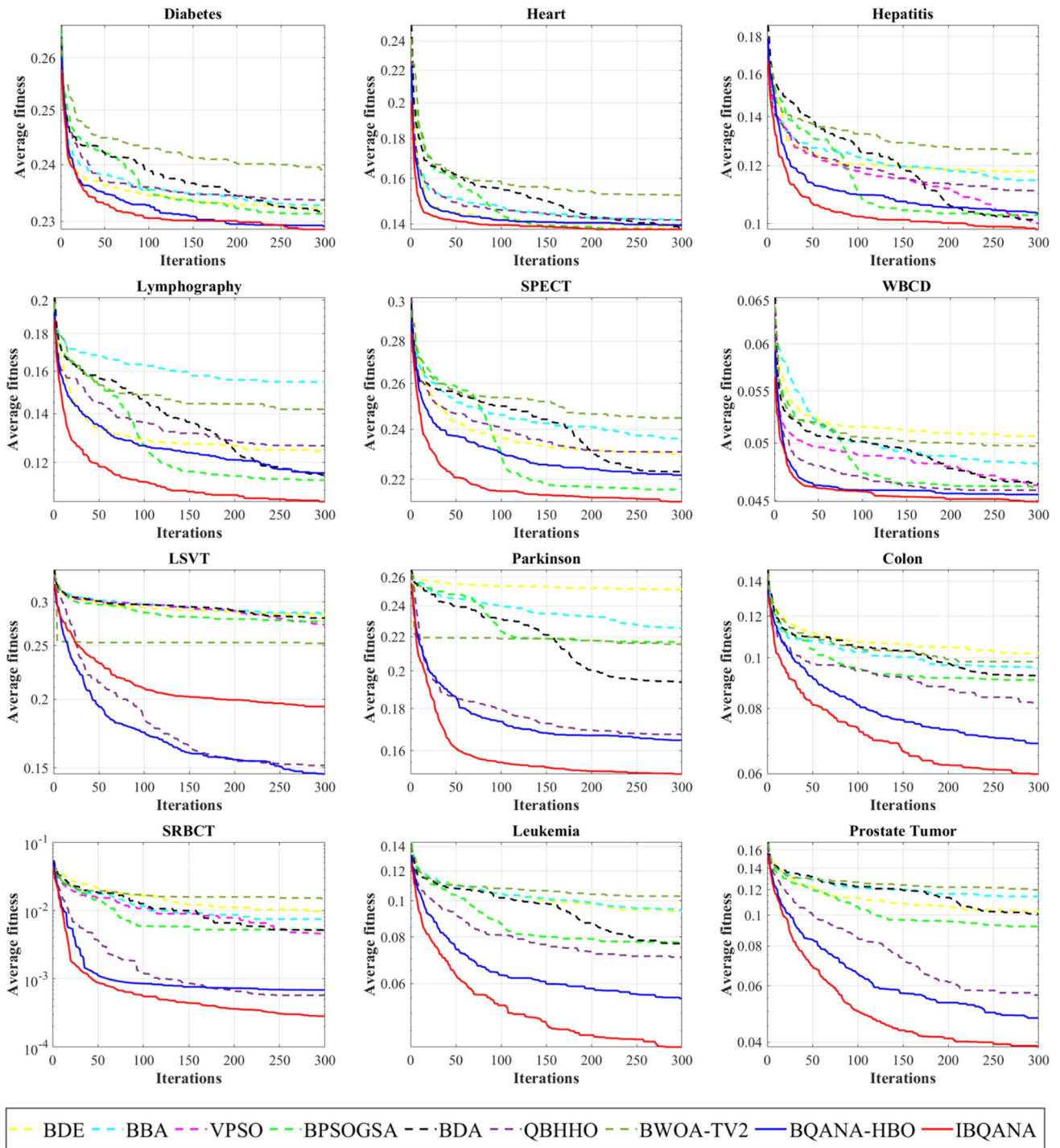
features degrades the accuracy of the classifier, which is not acceptable in medical datasets.

The nonparametric Friedman test [94] is conducted to compare and rank the performance of the algorithms in selecting effective features from the investigated datasets based on their obtained fitness value in 20 runs. The findings reported in Table 7 reveal that the proposed IBQANA is ranked first for 11 datasets and third for one dataset. It is also noticeable that the IBQANA is scalable as it performs best for high-dimensional datasets, including Parkinson, colon, SRBCT, leukemia, and prostate tumor. Furthermore, BQANA-HBO is ranked first for one dataset, second

for eight datasets, fourth for one dataset, and fifth for two datasets.

## 6 Applicability of the Proposed IBQANA on a COVID-19 Case Study

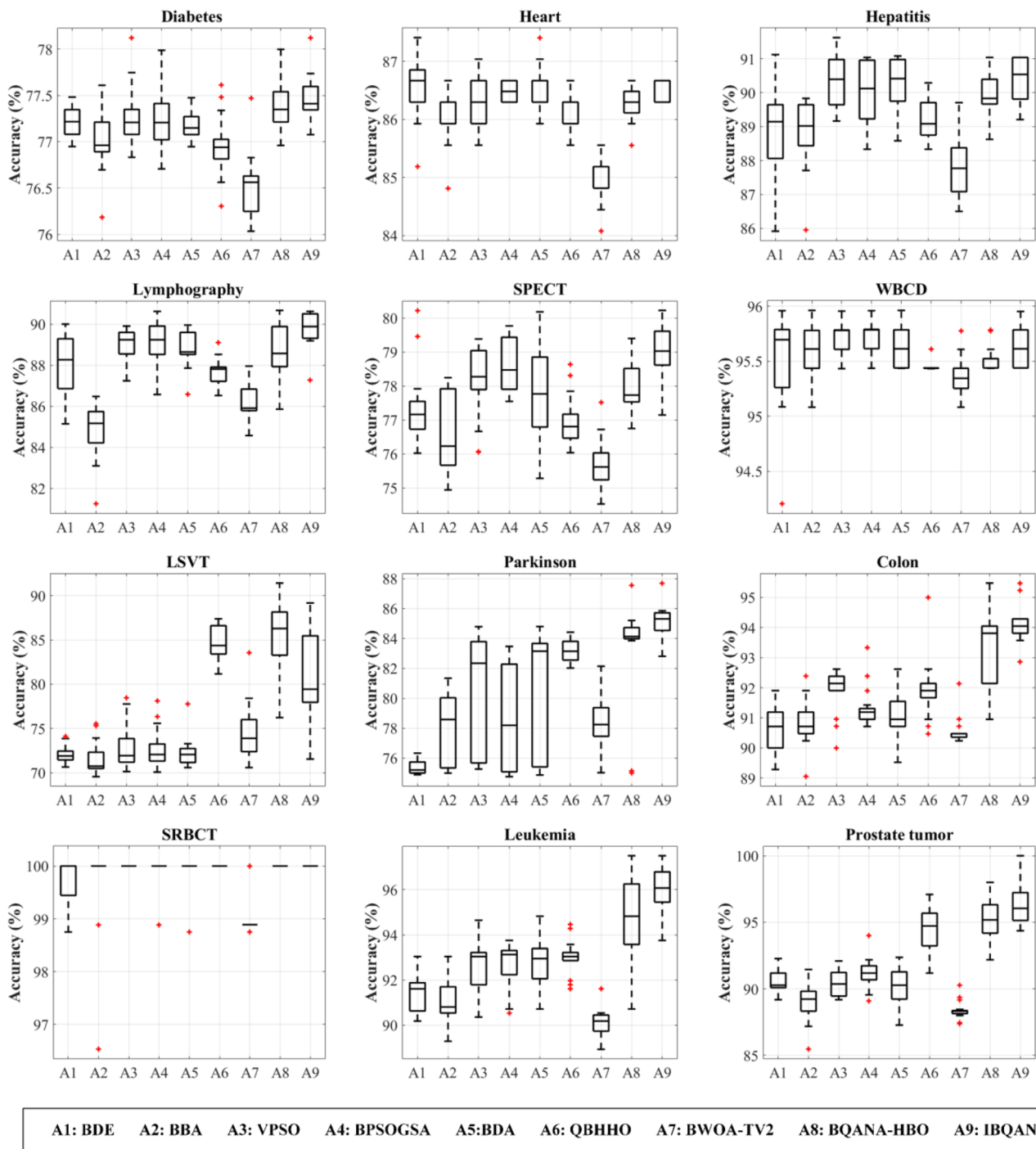
The infectious disease, COVID-19, attributed to the viral agent SARS-CoV-2, has garnered immense attention on a global scale since its initial emergence in January 2019, as corroborated by reference [95]. This virus causes severe acute respiratory syndrome. Due to its quick spread around



**Fig. 3** Convergence comparison of the IBQANA and comparative algorithms

the globe, World Health Organization (WHO) recognized it as a global crisis. Reportedly, COVID-19 has infected 619,429,000 and killed nearly 6,537,236 people worldwide ever since [96]. Machine learning has recently emerged as an effective approach to solve many problems and can be

used to combat COVID-19 through screening [97], monitoring [98], prediction [99], and diagnosis [83, 100, 101]. To make this possible, medical feature selection techniques extract useful features from clinical datasets, which are then



**Fig. 4** Boxplot of the classification accuracy obtained by the IBQANA and comparative algorithms

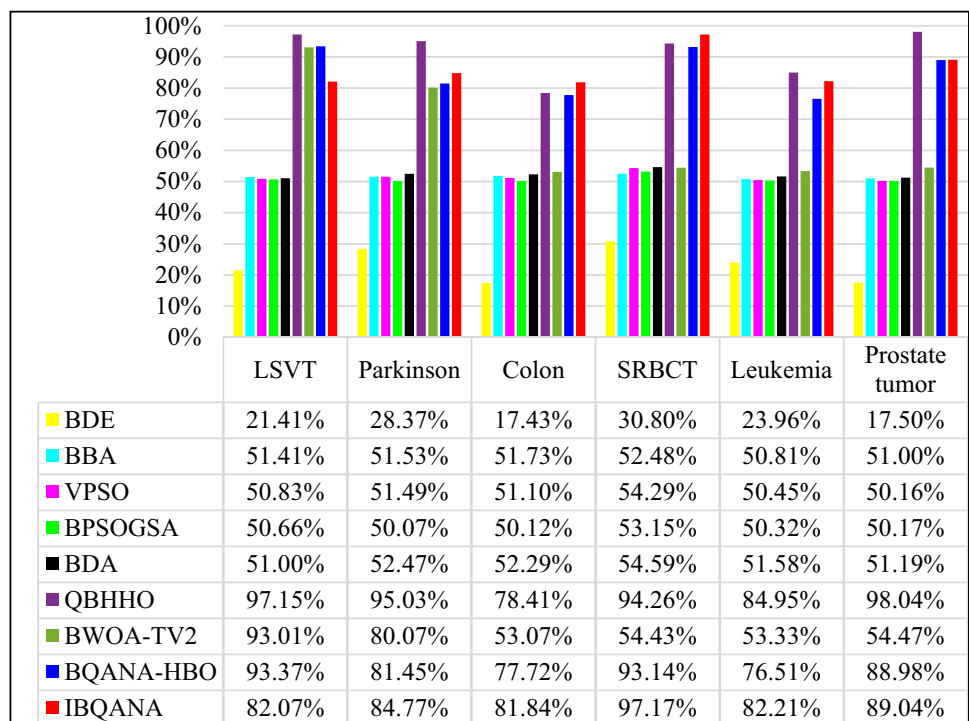
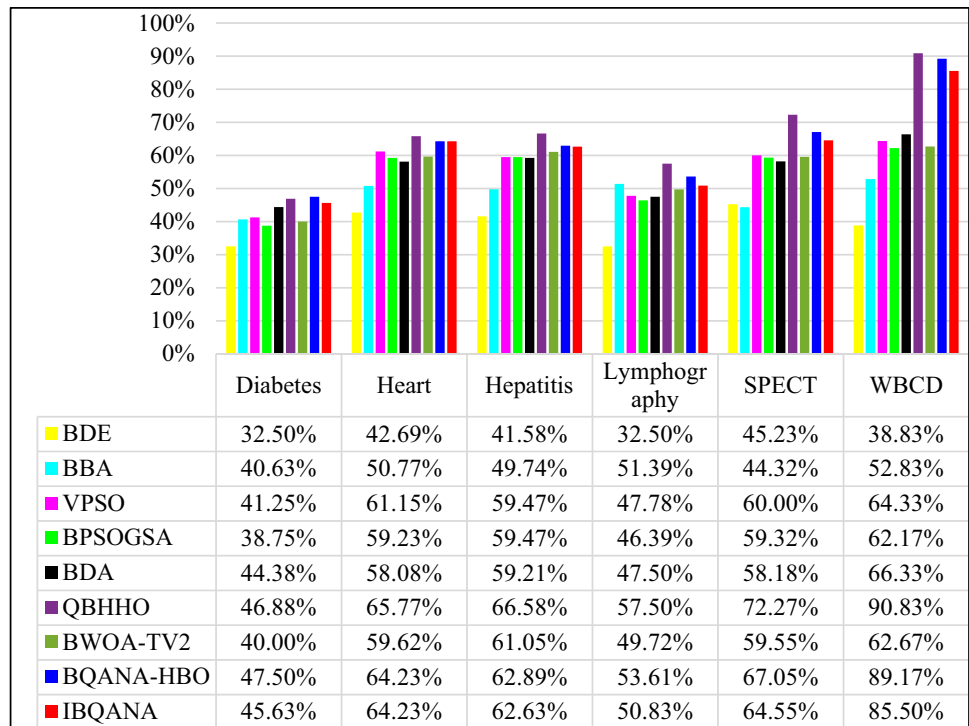
used to develop diagnosis algorithms to identify COVID-19 disease [102].

In this section, an evaluation of the potential applicability and performance of the proposed IBQANA is conducted on the preprocessed version [103] of the COVID-19 dataset [104], which is described in Table 8. This table shows that the dataset contains 13 features to predict the patient's

condition. The categorical columns in the preprocessed dataset have been converted to numerical values by assigning a unique number to each category. The experimental environments and parameters of the algorithms have been set to the values specified in Sect. 5.

The experimental results illustrated in the first plot of Fig. 6 reveal that the proposed IBQANA generates the best

**Fig. 5** The feature reduction rate of algorithms for each dataset



solutions with minimum fitness value among the investigated algorithms, and the BQANA-HBO attained the second rank with a similar convergence behavior (Fig. 6a). It is also noticeable that IBQANA achieves the quickest convergence towards optimal solutions and can generate optimal solutions faster than other algorithms. The boxplot in Fig. 6b shows that BQANA-HBO provides the highest maximum

classification accuracy, while IBQANA and BDE algorithms have the highest median accuracy. Ultimately, Fig. 6c presents each algorithm's minimum and average number of selected features, where IBQANA has the shortest bar for the average number of features. In contrast, QBHHO has the shortest bar for the minimum number of features.



**Table 7** The Friedman test for the fitness obtained by each algorithm

Datasets	BDE	BBA	VPSO	BPSOGSA	BDA	QBHHO	BWOA-TV2	BQANA-HBO	IBQANA
Diabetes	5.75	6.60	3.53	4.33	5.10	7.65	8.95	1.80	1.30
Heart	3.55	7.15	4.98	2.48	3.55	7.40	9.00	4.65	2.25
Hepatitis	7.90	7.10	2.25	3.88	3.35	6.10	8.80	4.35	1.28
Lymphography	6.25	9.00	3.38	2.95	4.08	6.30	8.00	4.00	1.05
SPECT	6.25	7.55	3.45	2.40	4.45	6.55	9.00	4.25	1.10
WBCD	8.45	7.10	4.63	3.95	4.85	3.95	8.45	2.45	1.18
LSVT	7.60	8.55	5.95	6.10	6.80	2.00	4.00	1.25	2.75
Parkinson	8.95	6.90	4.55	6.90	5.05	2.90	6.10	2.60	1.05
Colon	8.70	7.00	4.20	5.25	5.80	3.10	7.95	2.00	1.00
SRBCT	8.30	6.60	4.65	5.95	4.90	2.58	8.60	2.43	1.00
Leukemia	7.75	7.25	5.55	4.80	4.50	2.95	9.00	2.15	1.05
Prostate tumor	6.45	8.15	5.55	4.15	5.90	3.00	8.80	2.00	1.00
Average ranks	7.16	7.41	4.39	4.43	4.86	4.54	8.05	2.83	1.33
Overall rank	7	8	3	4	6	5	9	2	<b>1</b>

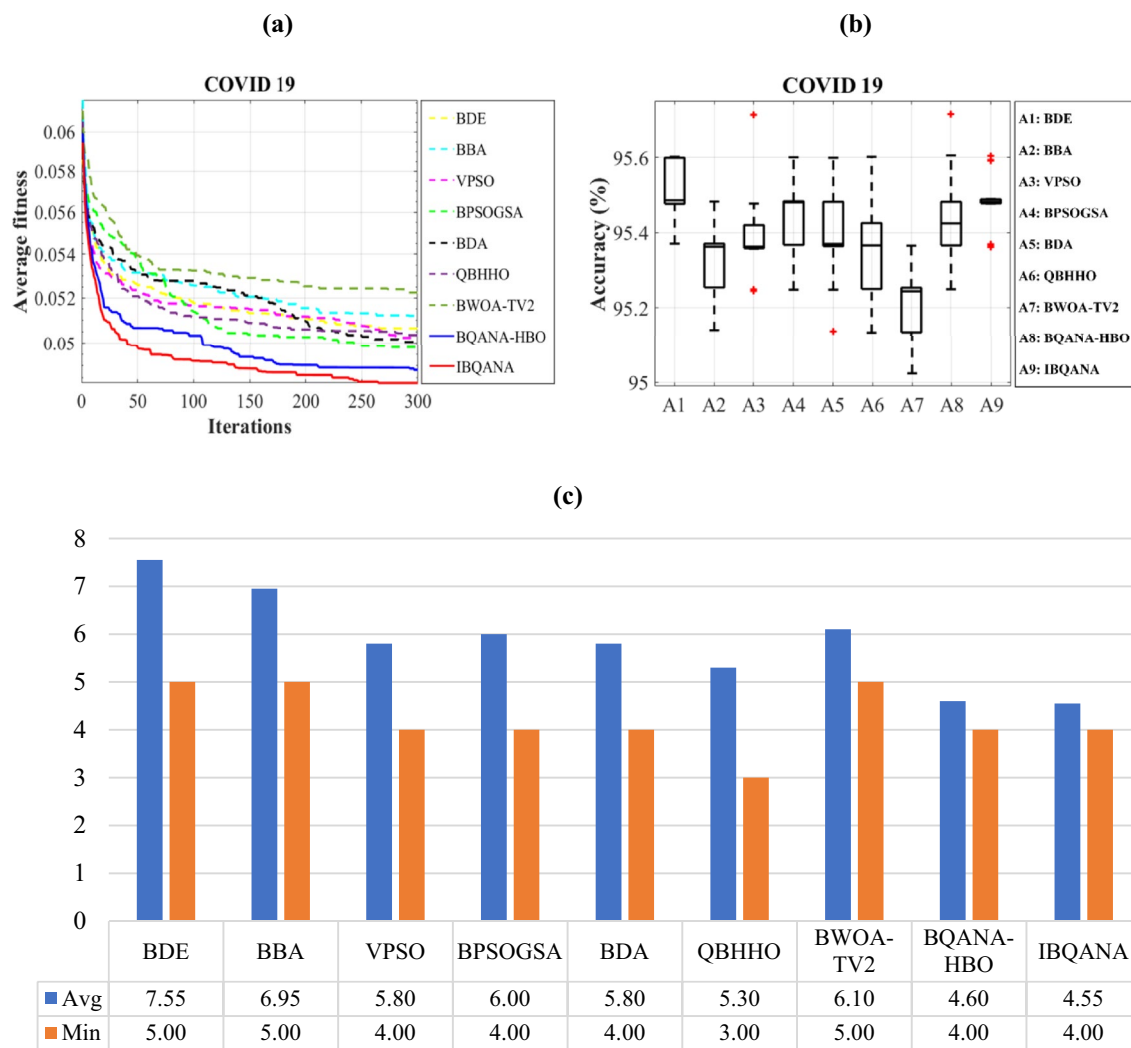
## 7 Conclusion and Future Work

FSS is crucial in data analysis, identifying relevant features, and reducing the computational burden. Existing literature suggests that metaheuristic algorithms effectively find optimal feature subsets quickly. However, current binary metaheuristic algorithms suffer from slow convergence and lack an effective binarization method, resulting in suboptimal solutions. This paper introduces the Improved Binary Quantum-based Avian Navigation Optimizer Algorithm (IBQANA) specifically designed for FSS on medical datasets. The algorithm employs a Hybrid Binary Operator (HBO) to effectively convert continuous values into binary solutions. In addition, a Distance-Based

Binary Search Strategy (DBSS) is introduced to enhance the performance of inferior search agents and accelerate convergence. DBSS operates through a two-phase search strategy, combining exploration and exploitation phases based on an adaptive probability function. This innovative approach effectively steers clear of local optima, ensuring optimal results. Experimental evaluation on 12 medical datasets demonstrates that IBQANA outperforms seven established algorithms regarding fitness, classification accuracy, and the number of selected features. It exhibits scalability in selecting features from high-dimensional datasets and demonstrates the fastest convergence among the tested algorithms. The application of IBQANA for COVID-19 detection underscores its practical significance and potential impact on the medical community. This

**Table 8** Description of the novel coronavirus 2019 dataset

No	Features name	Description
F <sub>1</sub>	Patient location	The geographic location of each patient included in the dataset
F <sub>2</sub>	Country of origin	The country where each patient was living at the time of their diagnosis
F <sub>3</sub>	Patient gender	The gender of each patient included in the dataset
F <sub>4</sub>	Patient age	The age range of patients included in the dataset
F <sub>5</sub>	Visited Wuhan	Whether each patient had visited Wuhan, China prior to their diagnosis (1 = yes, 0 = no)
F <sub>6</sub>	From Wuhan	Whether each patient was from Wuhan, China (1 = yes, 0 = no)
F <sub>7</sub>	Fever	Whether each patient presented with fever as a symptom of COVID-19 (1 = yes, 0 = no)
F <sub>8</sub>	Cough	Whether each patient presented with cough as a symptom of COVID-19 (1 = yes, 0 = no)
F <sub>9</sub>	Cold	Whether each patient presented with cold-like symptoms as a symptom of COVID-19 (1 = yes, 0 = no)
F <sub>10</sub>	Fatigue	Whether each patient presented with fatigue as a symptom of COVID-19 (1 = yes, 0 = no)
F <sub>11</sub>	Body pain	Whether each patient presented with body aches or pain as a symptom of COVID-19 (1 = yes, 0 = no)
F <sub>12</sub>	Malaise	Whether each patient presented with a general feeling of malaise as a symptom of COVID-19 (1 = yes, 0 = no)
F <sub>13</sub>	Time to hospitalization	The number of days between symptom onset and hospitalization for each patient in the dataset
F <sub>14</sub>	Patient outcome	Whether each patient either recovered from COVID-19 or died as a result of the illness



**Fig. 6** Comparison of the IBQANA and comparative algorithms on novel coronavirus 2019 dataset in terms of fitness value convergence (a), classification accuracy (b), and number of selected features (c)

research offers a promising solution to the FSS problem in medical data preprocessing, with implications for diagnostic tool development. Future research can explore HBO as a binarization method for other continuous metaheuristic algorithms and implement DBSS to mitigate limitations in other binary metaheuristic algorithms, such as slow convergence and local optima trapping.

**Data Availability** The datasets used during the current study are available from the corresponding author upon reasonable request.

**Declarations**

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

**References**

1. Maimon, O., & Rokach, L. (2005). *Data mining and knowledge discovery handbook* (p. 1). Springer.
2. Mirjalili, S., Faris, H., & Aljarah, I. (2019). *Evolutionary machine learning techniques* (p. 12). Springer.
3. Duangsoithong, R., Windeatt, T. (2009) Relevant and redundant feature analysis with ensemble classification. *Seventh International Conference on Advances in Pattern Recognition*, Kolkata, India 247–250
4. Jović, A., Brkić, K., Bogunović, N. (2015) A review of feature selection methods with applications. *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 1200–1205
5. Chen, C.-W., Tsai, Y.-H., Chang, F.-R., & Lin, W.-C. (2020). Ensemble feature selection in medical datasets: combining filter, wrapper, and embedded feature selection results. *Expert Systems*, 37(5), e12553. <https://doi.org/10.1111/exsy.12553>

6. Taha, A., Cosgrave, B., & McKeever, S. (2022). Using feature selection with machine learning for generation of insurance insights. *Applied Sciences*, *12*(6), 3209.
7. Deriche, M. (2009) Feature selection using ant colony optimization. *6th International Multi-Conference on Systems, Signals and Devices*, Djerba, Tunisia, 1–4.
8. Cervante, L., Xue, B., Zhang, M., Shang, L. (2012) Binary particle swarm optimisation for feature selection: A filter based approach. *IEEE Congress on Evolutionary Computation*, Brisbane, QLD, Australia, 1–8.
9. Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1), 273–324. [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X)
10. Huang, J., Cai, Y., & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters*, *28*(13), 1825–1844.
11. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(7–8), 1157–1182.
12. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *MachINE Learning*, *46*(1), 389–422. <https://doi.org/10.1023/A:1012487302797>
13. Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, *3*(7–8), 1357–1370.
14. Purohit, A., Chaudhari, NS., Tiwari, A. (2010) Construction of classifier with feature selection based on genetic programming. *IEEE Congress on Evolutionary Computation*, Barcelona, Spain, 1–5.
15. Akinola, O., Agushaka, O., & Ezugwu, A. (2022). Binary dwarf mongoose optimizer for solving high-dimensional feature selection problems. *PLoS ONE*, *17*(10), 1–26. <https://doi.org/10.1371/journal.pone.0274850>
16. Kumar, V., & Minz, S. (2014). Feature selection: a literature review. *SmartCR*, *4*(3), 211–229.
17. Koller, D., Sahami, M. (1996) Toward optimal feature selection. *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. Bari, Italy, 1996: 284–292.
18. Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence*, *32*, 112–123.
19. Foroutan, I., & Sklansky, J. (1987). Feature selection for automatic classification of non-gaussian data. *IEEE Transactions on Systems, Man, and Cybernetics*, *17*(2), 187–198.
20. Nadimi-Shahraki, M. H., Fatahi, A., Zamani, H., Mirjalili, S., Abualigah, L., & Abd Elaziz, M. (2021). Migration-based moth-flame optimization algorithm. *Processes*, *9*(12), 2276.
21. Kennedy, J., Eberhart, R. (1948) Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, 1995, 1942–1948.
22. Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–73.
23. Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.
24. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39.
25. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471.
26. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.
27. Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, *89*, 228–249.
28. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.
29. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872.
30. Li, K., Xu, G., Zhao, G., Dong, Y., Wang, D. (2011) Cloud task scheduling based on load balancing ant colony optimization. *Sixth Annual ChinaGrid Conference*. Liaoning, China, 3–9.
31. Awad, A., El-Hefnawy, N., & Abdel-kader, H. (2015). Enhanced particle swarm optimization for task scheduling in cloud computing environments. *Procedia Computer Science*, *65*, 920–929.
32. Omara, F. A., & Arafa, M. M. (2010). Genetic algorithms for task scheduling problem. *Journal of Parallel and Distributed Computing*, *70*(1), 13–22. <https://doi.org/10.1016/j.jpdc.2009.09.009>
33. Nadimi-Shahraki, M. H., Fatahi, A., Zamani, H., Mirjalili, S., & Oliva, D. (2022). Hybridizing of whale and moth-flame optimization algorithms to solve diverse scales of optimal power flow problem. *Electronics*, *11*(5), 831.
34. Farhat, M., Kamel, S., Atallah, A. M., & Khan, B. (2022). Developing a marine predator algorithm for optimal power flow analysis considering uncertainty of renewable energy sources. *International Transactions on Electrical Energy Systems*, *2022*, 3714475. <https://doi.org/10.1155/2022/3714475>
35. Nadimi-Shahraki, M. H., Fatahi, A., Zamani, H., Mirjalili, S., & Abualigah, L. (2021). An improved moth-flame optimization algorithm with adaptation mechanism to solve numerical and mechanical engineering problems. *Entropy*, *23*(12), 1637.
36. Yıldız, B. S., Pholdee, N., Bureerat, S., Erdaş, M. U., Yıldız, A. R., & Sait, S. M. (2021). Comparison of the political optimization algorithm, the Archimedes optimization algorithm and the levy flight algorithm for design optimization in industry. *Materials Testing*, *63*(4), 356–359.
37. Sharma, S., Saha, A. K., Roy, S., Mirjalili, S., & Nama, S. (2022). A mixed sine cosine butterfly optimization algorithm for global optimization and its application. *Cluster Computing*, *25*(6), 4573–4600. <https://doi.org/10.1007/s10586-022-03649-5>
38. Nadimi-Shahraki, M. H., & Zamani, H. (2022). DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization. *Expert Systems with Applications*, *198*, 116895.
39. Yıldız, B. S., Pholdee, N., Bureerat, S., Yıldız, A. R., & Sait, S. M. (2021). Robust design of a robot gripper mechanism using new hybrid grasshopper optimization algorithm. *Expert Systems*, *38*(3), e12666.
40. Gupta, S., Abderazek, H., Yıldız, B. S., Yıldız, A. R., Mirjalili, S., & Sait, S. M. (2021). Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems. *Expert Systems with Applications*, *183*, 115351.
41. Yıldız, B. S., Kumar, S., Pholdee, N., Bureerat, S., Sait, S. M., & Yıldız, A. R. (2022). A new chaotic Lévy flight distribution optimization algorithm for solving constrained engineering problems. *Expert Systems*, *39*(8), e12992.
42. Yıldız, B. S., Pholdee, N., Panagant, N., Bureerat, S., Yıldız, A. R., & Sait, S. M. (2022). A novel chaotic Henry gas solubility optimization algorithm for solving real-world engineering problems. *Engineering with Computers*, *38*(2), 871–883. <https://doi.org/10.1007/s00366-020-01268-5>

43. Nadimi-Shahraki, M. H., Zamani, H., Fatahi, A., & Mirjalili, S. (2023). MFO-SFR: An enhanced moth-flame optimization algorithm using an effective stagnation finding and replacing strategy. *Mathematics*, *11*(4), 862.
44. Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: a nature-inspired metaheuristic. *Expert Systems with Applications*, *152*, 113377.
45. Tu, J., Chen, H., Wang, M., & Gandomi, A. H. (2021). The colony predation algorithm. *Journal of Bionic Engineering*, *18*(3), 674–710.
46. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, *157*, 107250.
47. Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2022). Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Computer Methods in Applied Mechanics and Engineering*, *392*, 114616.
48. Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022). Dwarf mongoose optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, *391*, 114570.
49. Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2021). QANA: Quantum-based avian navigation optimizer algorithm. *Engineering Applications of Artificial Intelligence*, *104*, 104314.
50. Abdel-Basset, M., El-Shahat, D., Chakraborty, R. K., & Ryan, M. (2021). Parameter estimation of photovoltaic models using an improved marine predators algorithm. *Energy Conversion and Management*, *227*, 113491.
51. Wang, S., Jia, H., Abualigah, L., Liu, Q., & Zheng, R. (2021). An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. *Processes*, *9*(9), 1551.
52. El-Kenawy, E.-S.M., Mirjalili, S., Khodadadi, N., Abdelhamid, A. A., Eid, M. M., El-Said, M., & Ibrahim, A. (2023). Feature selection in wind speed forecasting systems based on meta-heuristic optimization. *PLoS ONE*, *18*(2), e0278491.
53. Kennedy, J., Eberhart, R.C. (1997) A discrete binary version of the particle swarm algorithm. *IEEE International Conference on Systems, Man, and Cybernetics*. Computational cybernetics and simulation, Orlando, FL, USA, 1997, 4104–4108.
54. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). BGSA: Binary gravitational search algorithm. *Natural Computing*, *9*(3), 727–745.
55. Gong, T. (2007). *Tuson* (pp. 251–262). AL. Differential Evolution for Binary Encoding. Berlin.
56. Nakamura, RYM., Pereira, LAM., Costa, KA., Rodrigues, D., Papa, JP., Yang, XS. (2012) BBA: A Binary Bat Algorithm for Feature Selection. *25th SIBGRAPI Conference on Graphics, Patterns and Images*, Ouro Preto, Brazil, 2012, 291–297.
57. Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, *172*, 371–381.
58. Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, *27*(4), 1053–1073.
59. Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Alam, A.-Z., Mirjalili, S., & Fujita, H. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, *154*, 43–67.
60. He, Y., Xie, H., Wong, T.-L., & Wang, X. (2018). A novel binary artificial bee colony algorithm for the set-union knapsack problem. *Future Generation Computer Systems*, *78*, 77–86.
61. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.
62. Nama, S., Saha, A. K., Chakraborty, S., Gandomi, A. H., & Abualigah, L. (2023). Boosting particle swarm optimization by backtracking search algorithm for optimization problems. *Swarm and Evolutionary Computation*, *79*, 101304.
63. Chakraborty, S., Saha, A. K., & Chhabra, A. (2023). Improving whale optimization algorithm with elite strategy and its application to engineering-design and cloud task scheduling problems. *Cognitive Computation*. <https://doi.org/10.1007/s12559-022-10099-z>
64. Bansal, P., Gehlot, K., Singhal, A., & Gupta, A. (2022). Automatic detection of osteosarcoma based on integrated features and feature selection using binary arithmetic optimization algorithm. *Multimedia Tools and Applications*, *81*(6), 8807–8834.
65. Liu, Y., Heidari, A. A., Cai, Z., Liang, G., Chen, H., Pan, Z., Alsufyani, A., & Bourouis, S. (2022). Simulated annealing-based dynamic step shuffled frog leaping algorithm: optimal performance design and feature selection. *Neurocomputing*, *503*, 325–362.
66. Khalid, A. M., Hamza, H. M., Mirjalili, S., & Hosny, K. M. (2022). BCOVIDOA: a novel binary coronavirus disease optimization algorithm for feature selection. *Knowledge-Based Systems*, *248*, 108789.
67. Xu, Z., Heidari, A. A., Kuang, F., Khalil, A., Mafarja, M., Zhang, S., Chen, H., & Pan, Z. (2023). Enhanced Gaussian bare-bones grasshopper optimization: Mitigating the performance concerns for feature selection. *Expert Systems with Applications*, *212*, 118642.
68. Nadimi-Shahraki, M. H., Fatahi, A., Zamani, H., & Mirjalili, S. (2022). Binary approaches of quantum-based avian navigation optimizer to select effective features from high-dimensional medical data. *Mathematics*, *10*(15), 2770.
69. Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1–14.
70. Mirjalili, S., Wang, G.-G., & Coelho, Ld. S. (2014). Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Computing and Applications*, *25*(6), 1423–1435.
71. Too, J., Abdullah, A. R., & Mohd Saad, N. (2019). A new quadratic binary Harris hawk optimization for feature selection. *Electronics*, *8*(10), 1130.
72. Kahya, M. A., Altamir, S. A., & Algamil, Z. Y. (2021). Improving whale optimization algorithm for feature selection with a time-varying transfer function. *Numerical Algebra, Control & Optimization*, *11*(1), 87.
73. Liao, C.-J., Tseng, C.-T., & Luarn, P. (2007). A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research*, *34*(10), 3099–3111.
74. Marandi, A., Afshinmanesh, F., Shahabadi, M., Bahrami, F. (2006) Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna. *IEEE International Conference on Evolutionary Computation*, Vancouver, BC, Canada, 2006, 3212–3218.
75. Aslan, M., Gunduz, M., & Kiran, M. S. (2019). JayaX: jaya algorithm with xor operator for binary optimization. *Applied Soft Computing*, *82*, 105576.
76. Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19–34.
77. Jordehi, A. R. (2019). Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes. *Applied Soft Computing*, *78*, 465–480.



78. Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2019). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 31(1), 171–188.
79. Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., Noman, N. (2020). A novel U-shaped transfer function for binary particle swarm optimisation. *Soft Computing for Problem Solving* 2019, Singapore, 241–259.
80. Nadimi-Shahraki, M. H., Zamani, H., Asghari Varzaneh, Z., & Mirjalili, S. (2023). A Systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. *Archives of Computational Methods in Engineering*. <https://doi.org/10.1007/s11831-023-09928-7>
81. Mohammadzadeh, H., & Gharehchopogh, F. S. (2021). A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study Email spam detection. *Computational Intelligence*, 37(1), 176–209.
82. Turkoglu, B., Uymaz, S. A., & Kaya, E. (2022). Binary artificial algae algorithm for feature selection. *Applied Soft Computing*, 120, 108630.
83. Piri, J., Mohapatra, P., Acharya, B., Gharehchopogh, F. S., Gerogianis, V. C., Kanavos, A., & Manika, S. (2022). Feature selection using artificial gorilla troop optimization for biomedical data: a case analysis with COVID-19 data. *Mathematics*, 10(15), 2742.
84. Abualigah, L., & Diabat, A. (2022). Chaotic binary group search optimizer for feature selection. *Expert Systems with Applications*, 192, 116368.
85. Shaddeli, A., Soleimani Gharehchopogh, F., Masdari, M., & Solouk, V. (2022). An improved african vulture optimization algorithm for feature selection problems and its application of sentiment analysis on movie reviews. *Big Data and Cognitive Computing*, 6(4), 104.
86. Helmi, A. M., Al-qaness, M. A., Dahou, A., & Abd Elaziz, M. (2023). Human activity recognition using marine predators algorithm with deep learning. *Future Generation Computer Systems*, 142, 340–350.
87. Al-Qaness, M. A., Helmi, A. M., Dahou, A., & Elaziz, M. A. (2022). The applications of metaheuristics for human activity recognition and fall detection using wearable sensors: A comprehensive analysis. *Biosensors*, 12(10), 821.
88. Ren, W., Bashkandi, A. H., Jahanshahi, J. A., AlHamad, A. Q. M., Javaheri, D., & Mohammadi, M. (2023). Brain tumor diagnosis using a step-by-step methodology based on courtship learning-based water strider algorithm. *Biomedical Signal Processing and Control*, 83, 104614.
89. Nadimi-Shahraki, M. H., Asghari Varzaneh, Z., Zamani, H., & Mirjalili, S. (2023). Binary starling murmuration optimizer algorithm to select effective features from medical data. *Applied Sciences*, 13(1), 564.
90. Tanabe, R., Fukunaga, A. (2013) Success-history based parameter adaptation for differential evolution. *IEEE congress on evolutionary computation*, Cancun, Mexico, 71–78.
91. Pelusi, D., Mascella, R., Tallini, L., Nayak, J., Naik, B., & Deng, Y. (2020). An improved moth-flame optimization algorithm with hybrid search phase. *Knowledge-Based Systems*, 191, 105277.
92. Gupta, S., & Deep, K. (2019). A novel random walk grey wolf optimizer. *Swarm and Evolutionary Computation*, 44, 101–112. <https://doi.org/10.1016/j.swevo.2018.01.001>
93. Kelly, M., Longjohn, R., Nottingham, K. The UCI machine learning repository. Retrieved April 1, 2022, from <https://archive.ics.uci.edu>
94. Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1), 86–92.
95. Kutsuna, S. (2021). Clinical manifestations of coronavirus disease 2019. *Japan Medical Association*, 4(2), 76–80.
96. TheWorldometers: COVID-19 Coronavirus Pandemic. Retrieved 24 Sep 2022 from <https://www.worldometers.info/coronavirus/>
97. Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., & Al Emadi, N. (2020). Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, 8, 132665–132676. <https://doi.org/10.1109/ACCESS.2020.3010287>
98. Snyder, S. E., & Husari, G. (2021). Thor: A deep learning approach for face mask detection to prevent the COVID-19 pandemic. In S. E. Snyder (Ed.), *Southeast Con 2021, Atlanta*. USA: GA.
99. Babukarthik, R., Adiga, V. A. K., Sambasivam, G., Chandramohan, D., & Amudhavel, J. (2020). Prediction of COVID-19 using genetic deep learning convolutional neural network (GDCNN). *IEEE Access*, 8, 177647–177666. <https://doi.org/10.1109/ACCESS.2020.3025164>
100. Chakraborty, S., Saha, A. K., Nama, S., & Debnath, S. (2021). COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. *Computers in Biology and Medicine*, 139, 104984. <https://doi.org/10.1016/j.combiomed.2021.104984>
101. Kumar Sahoo, S., Houssein, E. H., Premkumar, M., Kumar Saha, A., & Emam, M. M. (2023). Self-adaptive moth flame optimizer combined with crossover operator and Fibonacci search strategy for COVID-19 CT image segmentation. *Expert Systems with Applications*, 227, 120367. <https://doi.org/10.1016/j.eswa.2023.120367>
102. Hemdan, E.E.-D., Shouman, M. A., & Karar, M. E. (2020). Covidxnet: a framework of deep learning classifiers to diagnose covid-19 in x-ray images. *ArXiv preprint ArXiv*. <https://doi.org/10.48550/arXiv.2003.11055>
103. N.C. Virus, Dataset, Kaggle. Retrieved 25 September 2020 from <https://www.kaggle.com/datasets/sudalairajkumar/novel-coronavirus-2019-dataset>
104. Iwendi, C., Bashir, A. K., Peshkar, A., Sujatha, R., Chatterjee, J. M., Pasupuleti, S., Mishra, R., Pillai, S., & Jo, O. (2020). COVID-19 patient health prediction using boosted random forest algorithm. *Frontiers in Public Health*, 8, 357.

**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.