mathematics

MDPI

Article

Binary Approaches of Quantum-Based Avian Navigation Optimizer to Select Effective Features from High-Dimensional Medical Data

Mohammad H. Nadimi-Shahraki [1,2,3,*], Ali Fatahi [1,2], Hoda Zamani [1,2] and Seyedali Mirjalili [3,4,*]

[1] Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran
[2] Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad 8514143131, Iran
[3] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane 4006, Australia
[4] Yonsei Frontier Lab, Yonsei University, Seoul 03722, Korea
* Correspondence: nadimi@iaun.ac.ir (M.H.N.-S.); ali.mirjalili@torrens.edu.au (S.M.)

Abstract: Many metaheuristic approaches have been developed to select effective features from different medical datasets in a feasible time. However, most of them cannot scale well to large medical datasets, where they fail to maximize the classification accuracy and simultaneously minimize the number of selected features. Therefore, this paper is devoted to developing an efficient binary version of the quantum-based avian navigation optimizer algorithm (QANA) named BQANA, utilizing the scalability of the QANA to effectively select the optimal feature subset from high-dimensional medical datasets using two different approaches. In the first approach, several binary versions of the QANA are developed using S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic transfer functions to map the continuous solutions of the canonical QANA to binary ones. In the second approach, the QANA is mapped to binary space by converting each variable to 0 or 1 using a threshold. To evaluate the proposed algorithm, first, all binary versions of the QANA are assessed on different medical datasets with varied feature sizes, including Pima, HeartEW, Lymphography, SPECT Heart, PenglungEW, Parkinson, Colon, SRBCT, Leukemia, and Prostate tumor. The results show that the BQANA developed by the second approach is superior to other binary versions of the QANA to find the optimal feature subset from the medical datasets. Then, the BQANA was compared with nine well-known binary metaheuristic algorithms, and the results were statistically assessed using the Friedman test. The experimental and statistical results demonstrate that the proposed BQANA has merit for feature selection from medical datasets.

Keywords: optimization; feature selection; binary metaheuristic algorithms; swarm intelligence algorithms; medical datasets; transfer functions; classification; machine learning

MSC: 68T20

check for updates

Citation: Nadimi-Shahraki, M.H.; Fatahi, A.; Zamani, H.; Mirjalili, S. Binary Approaches of Quantum-Based Avian Navigation Optimizer to Select Effective Features from High-Dimensional Medical Data. *Mathematics* **2022**, *10*, 2770. https://doi.org/10.3390/math10152770

Academic Editor: Junde Wu

Received: 6 July 2022
Accepted: 1 August 2022
Published: 4 August 2022

1. Introduction

In recent years, artificial intelligence technologies have been used to solve various problems [1], which dictates the importance of storing data and information. With continued advances in science, a plethora of enormous datasets, including a large number of features, are being stored in different fields, such as business, text mining, biology, and medicine. Since medical datasets are often gathered for different purposes and from different sources, they may have challenges and complexities, such as structural and type heterogeneity, high dimensional, outliers, missing values, skewness, integration, and irrelevant and redundant features [2–4]. The existence of irrelevant and redundant features may degrade the accuracy of the classifier and bring additional computational costs [5]. To tackle this issue, many effective methods have been proposed to select effective features by reducing such disadvantageous features [6–10]. Feature selection (FS) is employed in a wide range of real-world applications, including disease diagnosis [11–15], text clustering [16,17], intrusion detection systems [18–21], e-mail spam detection [22–25], and genomic analysis [25–29].

The FS algorithms are broadly classified into filter-based, wrapper-based, and embedded-based methods [30–32]. The filter-based methods assess and rank features of datasets based on principle criteria such as distance, information, similarity, consistency, and statistical measures [33,34]. Although filter-based methods demand lower computational costs than other methods, they cannot provide satisfactory performance. The wrapper-based methods search for an optimal feature subset using a predetermined learning algorithm for evaluating the feature subsets. The advantages of both filter-based and wrapper-based methods are combined in embedded-based methods. These methods incorporate the search for an optimal feature subset as part of the classifier training process [32]. The wrapper-based methods can generally provide greater classification accuracy than other methods by using a machine-learning algorithm to assess possible solutions [6,35]. Since assessing 2^N subsets of problem space with N features is an NP-hard issue, near-optimal subsets are discovered using approximate algorithms that heuristically search for an optimal subset [36–38].

Metaheuristic algorithms are a subset of approximate algorithms that have been used for solving many NP-hard problems in different fields of science, such as engineering design [39–50], task scheduling [51–53], engineering prediction [54–58], and optimal power flow [59–64] problems. When tackling the FS problem, metaheuristic algorithms have shown outstanding results in prior studies [65–68]. For instance, Emary et al. [69] introduced two versions of binary grey wolf optimizer (bGWO) to solve the FS problem as a wrapper-based method. The first approach was developed by performing stochastic crossover among the three best solutions, while in the second approach, the authors applied the S-shaped transfer function to convert continuous solutions of GWO to binary ones. Mafarja et al. [70] proposed a binary grasshopper optimization algorithm (BGOA) to tackle the feature selection problem within a wrapper-based framework by applying S-shaped and V-shaped transfer functions as the first mechanism. The second mechanism employs a new method that combines the finest solutions found so far. Furthermore, a mutation operator is used in the BGOA algorithm to improve the exploration phase.

Sindhu et al. [71] proposed an improved sine cosine algorithm (ISCA) that includes a feature selection elitism technique and a new best solution update method to select the best features and increase the classification accuracy. Dhiman et al. [72] developed eight binary versions of the emperor penguin optimizer to solve the FS problem by employing S-shaped and V-shaped transfer functions. In this study, 25 standard benchmark functions have been used to validate the results of the developed algorithms. The results revealed that the V_4 transfer function provides better solutions than other transfer functions. A binary farmland fertility algorithm (BFFA) has been proposed by Naseri et al. [18] to tackle feature selection problems in intrusion detection systems using a V-shaped transfer function. Although many metaheuristic algorithms have been developed in the FS domain, most of them are not scalable enough to overcome small and large datasets.

Quantum-based avian navigation optimizer algorithm (QANA) [73] is a recently introduced evolutionary algorithm inspired by the navigation mechanism of migratory birds during long-distance aerial paths for solving continuous optimization problems. The QANA provides competitive results by employing several operators, including population partitioning, self-adaptive quantum orientation, a qubit-crossover, and two mutation strategies. Moreover, the gained information is shared among search agents using a V-echelon communication topology. The experimental evaluations reveal that QANA is scalable for solving high-dimensional problems. It is worth mentioning that when working with high-dimensional datasets, tackling optimization problems becomes particularly difficult due to the curse of dimensionality problems [74,75].

This paper aims to extend our earlier study [73] by using our proposed binary QANA (BQANA) to overcome the curse of dimensionality difficulties in the FS domain and generate high-quality solutions using two approaches. In the first approach, the canonical QANA is converted to binary using 20 different transfer functions from five categories of S-shaped [76], V-shaped [77], U-shaped [78], Z-shaped [79], and quadratic [80,81] to solve FS problem in medical datasets. The transfer functions are discussed in this paper, and then they are paired with the QANA to develop several binary QANA models. In the second approach, a threshold is assigned for each dimension to map the continuous solutions of QANA to binary and develop BQANA without any further computational cost. The effectiveness of the proposed approaches is investigated on 10 medical datasets with various scales. To validate the proposed algorithms, the results of the winner version of binary QANA named BQANA were compared with the results of nine well-known metaheuristic algorithms, including binary particle swarm optimization (BPSO) [82], ant colony optimization (ACO) [83], binary differential evolution (BDE) [84], binary bat algorithm (BBA) [85], feature selection based on whale optimization algorithm (FSWOA) [11], binary ant lion optimizer (BALO) [86], binary dragonfly algorithm (BDA) [87], quadratic binary Harris hawk optimization (QBHHO) [80], and binary atom search optimization (BASO) [88]. The convergence behavior, the average number of selected features, and the accuracy of the proposed BQANA and comparative algorithms were visualized and investigated for all datasets. Moreover, the BQANA is statistically assessed by the Friedman test to demonstrate the algorithm's superiority. The main contributions of this study are summarized as follows.

- Introducing binary approaches of quantum-based avian navigation optimizer algorithm (QANA) to select effective features from high-dimensional medical datasets.
- The binary QANA variants have been developed by adapting the main components of the standard QANA.
- Comparing the behavior of QANA with different transfer functions from five different categories, S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic, to develop 20 versions of binary QANA based on the first approach.
- Applying the second approach as a low-cost and effective method to develop the superior version of binary QANA named BQANA by assigning a threshold for each variable.
- Dimensionality reduction, generating solutions with high accuracy and a minimum number of features are obtained by the second approach.
- The experiments prove that the BQANA developed by the second approach provides superior results compared to the first approach and nine other comparative algorithms in terms of increasing classification accuracy and minimizing the number of features for 10 medical datasets with various scales.

2. Related Works

The Fs is an NP-hard problem with discrete search space, in which the number of potential solutions grows exponentially as the number of features grows. Hence, metaheuristic algorithms are known as powerful optimizers in the literature. Ant colony optimization (ACO) [89] is a discrete metaheuristic algorithm inspired by the behavior of some ants in nature that has been applied for solving FS problems in different fields such as text categorization [90], image feature selection [91], intrusion detection system [92], and email spam classification [93]. As most of the metaheuristic algorithms are proposed for continuous search spaces, many researchers applied metaheuristic algorithms to discover an optimal feature subset by converting the continuous solutions into a binary form using logical operators or different transfer functions [94]. Logical operators have been applied for producing binary solutions due to their low computational costs [95]. Boolean particle swarm optimization was first proposed by Marandi et al. [96] to solve antenna design by converting continuous particle swarm optimization (PSO) into a binary form using three Boolean operators. In [97], the authors proposed a binary form of the Jaya algorithm named Jayax using the xor logical operator and incorporating a local search module to boost the algorithm's performance.

Many researchers apply different transfer functions to convert continuous values of optimizer algorithms into binary ones. The most well-known transfer functions used in the literature are S-shaped [76], V-shaped [77,98], U-shaped [78], Z-shaped [78], and quadratic [81] transfer functions. In 1997, Kennedy and Eberhart [76] introduced a binary version of the particle swarm optimization (BPSO) algorithm by applying the sigmoid transfer function to solve discrete optimization problems [82,99]. The sigmoid function is known as the S-shaped transfer function and has been applied to many other metaheuristic algorithms. Gong et al. [84] proposed binary differential evolution (BDE) algorithm to apply the differential evolution algorithm in discrete search space. To construct binary-adapted DE operators, DE operator templates are explicitly specified through the forma analysis. In [85] authors proposed a binary bat algorithm (BBA) for solving FS by applying the S-shaped transfer function to restrict the new search agent's position to only binary values. A new binary algorithm named feature selection based on whale optimization algorithm (FSWOA) was proposed by Zamani et al. [11] to handle the dimensionality of medical data using the whale optimization algorithm (WOA). To map continuous solutions of WOA to binary ones, the authors applied the S-shaped transfer function. The binary dragonfly algorithm (BDA) is the binary version of the dragonfly algorithm proposed by Mirjalili [87] which mimics the static and dynamic swarming behaviors of dragonflies in nature. The exploration and exploitation of the algorithm are modeled by the social interaction of dragonflies in searching for foods, avoiding enemies, and navigating when swarming dynamically or statistically.

The V-shaped transfer function introduced by Rashdi et al. [77] is a symmetric transfer function initially used in binary gravity search algorithm (BGSA) to map continuous values of GSA into binary ones. Emary et al. [86] proposed a binary antlion optimizer (BALO) for finding optimal feature subsets by applying S-shaped and V-shaped transfer functions. The findings indicate that the developed binary algorithm based on V-shaped transfer functions outperforms the S-shaped transfer functions. In a comparative study, Mirjalili et al. [98] evaluated six variants of S-shaped and V-shaped transfer functions on the traditional BPSO algorithm. The results indicate that the newly presented V-shaped family of transfer functions significantly outperforms the original BPSO. Too et al. [88] proposed eight versions of the binary atom search optimization (BASO) algorithm to effectively select an optimal feature subset by applying S-shaped and V-shaped transfer functions. In comparison to other BASO versions, the results showed that BASO with S-shaped transfer function (S_1) is highly capable of finding effective features.

Mirjalili et al. [68] proposed a new U-shaped transfer function for the PSO algorithm to convert continuous values of velocity to binary solutions. The obtained results reveal that the U-shaped transfer functions greatly enhance the performance of BPSO. The DEOSA proposed by Guha et al. [100] is a discrete combination of equilibrium optimizer and simulated annealing for selecting optimal features. This algorithm uses a U-shaped transfer function to convert continuous values into binary. Nadimi-Shahraki et al. [101] proposed an enhanced version of the whale optimization algorithm named E-WOA to solve continuous optimization problems using a pooling mechanism and three robust search strategies. To address the FS problem, the solutions of E-WOA are converted to binary form using a U-shaped (U_2) transfer function.

The Z-shaped [79] and quadratic [81] transfer functions are two recently proposed transfer functions for mapping continuous solutions to binary ones. The quadratic binary Harris hawk optimization (QBHHO) [80] algorithm is a binary version of the Harris hawk optimization algorithm developed by applying quadratic transfer functions for converting continuous solutions to binary. Considering a threshold for each variable is another efficient method to map continuous solutions to binary ones. Hafez et al. [102] utilized the sine cosine algorithm (SCA) to address the FS problem by assigning a variable threshold (0.5) to convert solutions to binary form. In [103], the authors proposed a PSO-based FS algorithm with a variable-length representation called VLPSO. The results showed that the variable-length representation enhances the scalability of PSO. The algorithm uses a predefined threshold (0.6) to map solutions into binary form.

3. Quantum-Based Avian Navigation Optimizer Algorithm (QANA)

QANA is a recent population-based metaheuristic algorithm inspired by the navigation behavior of migratory birds during long-distance aerial routes. The QANA is modeled using multi-flock construction and quantum-based navigation which consists of two mutation strategies and a qubit-crossover operator to explore the search space effectively.

3.1. Multi-Flock Construction

Initially, the population of migratory birds is randomly divided into multi-flocks. Next, the migratory birds' flight formation is mimicked in this algorithm to distribute the gained information among the search agents by adopting a V-echelon communication topology. Suppose V indicates a set of n members of the flock f_q, which includes a header (H) and two subsets called right-line (R) and left-line (L) considered in a V-shaped formation. The migratory birds' aerial maneuver using the V-echelon topology is depicted in Figure 1.

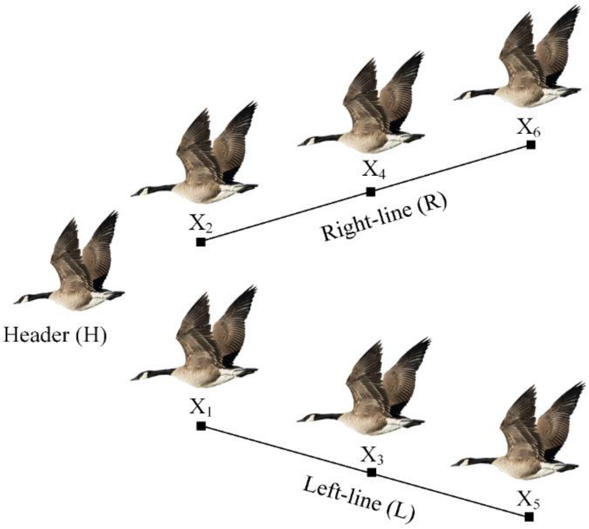

Figure 1. The V-shaped formation consists of a header (H), left-line (L), and right-line (R) [73].

3.2. Quantum-Based Navigation (Movement Strategy)

The flocks use quantum-based navigation to explore the search space, which includes a success-based population distribution (SPD) policy, two mutation strategies including "DE/quantum/I" and "DE/quantum/II," and a qubit-crossover operator. Each flock is dynamically allocated to one of these mutation techniques throughout the optimization process, depending on the SPD policy presented in Equation (1),

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

where f_m is the set of flocks that used M_m in iteration t, and τ_{ij} is equal to 1 if M_m improved a_j of the i-th flock in the set f_m; otherwise, τ_{ij} is equal to 0.

Quantum mutation strategies, including DE/quantum/I and DE/quantum/II, are described by Equations (2) and (3), where $x_i\ (t)$ denotes the position of search agent a_i in the current iteration t, $x_{V\_echelon}\ (t)$ is the position of the search agent followed by a_i, and $x_{best}\ (t)$ is the location of the best search agent. $x_j \in STM\ (t)$ and $x_j \in LTM\ (t)$ are randomly picked from short-term memory (STM) and long-term memory (LTM), respectively. Equation (4) is used to calculate the trial vector $v_H\ (t + 1)$ as a leader in the V-echelon topology, where L and U are the lower and upper bounds of the search space and S_i is the quantum orientation for avian $a_{i,}$ which is defined in [73], and it also uses parameter adaptation mechanism based on a historical record of successful parameter [104].

$$\begin{gathered} v_i(t+1) = x_{best}(t) + S_i(t) \times \left(x_{V_{echelon}}(t) - x_{j \in LTM}(t)\right) + \\ S_i(t) \times \left(x_{V_{echelon}}(t) - x_{best}(t)\right) + S_i(t) \times \left(x_{j \in LTM}(t) - x_{j \in STM}(t)\right) \end{gathered} \tag{2}$$

$$\begin{gathered} v_i(t+1) = S_i(t) \times \left(x_{best}(t) - x_{V_{echelon}}(t)\right) + \\ S_i(t) \times \left(x_i(t) - x_{j \in LTM}(t) - x_{j \in STM}(t)\right) \end{gathered} \tag{3}$$

$$v_H(t+1) = S_i(t) \times x_{best} + (L + (U - L) \times rand(0,1)) \tag{4}$$

To construct trial vector u_i (t + 1), the mutant vector v_i (t + 1) is crossed by its parent x_i (t) using Equation (5), where $\mid \psi_i\rangle_d$ is a qubit-crossover probability of the d-th dimension.

$$u_{id}(t+1) = \begin{cases} x_{id}(t+1), & |\psi_i\rangle_d < rand \\ v_{id}(t+1), & |\psi_i\rangle_d \geq rand \end{cases} \tag{5}$$

In each iteration, Equation (6) computes a qubit-crossover $\mid \psi_i\rangle_d$ for each dimension of the trial vector u_i (t + 1), where the parameter $\mid \psi_R\rangle_d$ is a random integer that serves as a coefficient for modifying the length of the vector $\mid \psi_i\rangle_d$ in the Bloch sphere.

$$\mid \psi_i\rangle_d = \mid \psi_R\rangle_d \times \left(cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi} sin\left(\frac{\theta}{2}\right)|1\rangle\right) \quad \theta, \varphi = rand \times \frac{\pi}{2} \tag{6}$$

Based on the avian navigator modeling given in the previous sections, the flowchart of the quantum-based avian navigation optimizer algorithm (QANA) is presented in Figure 2.

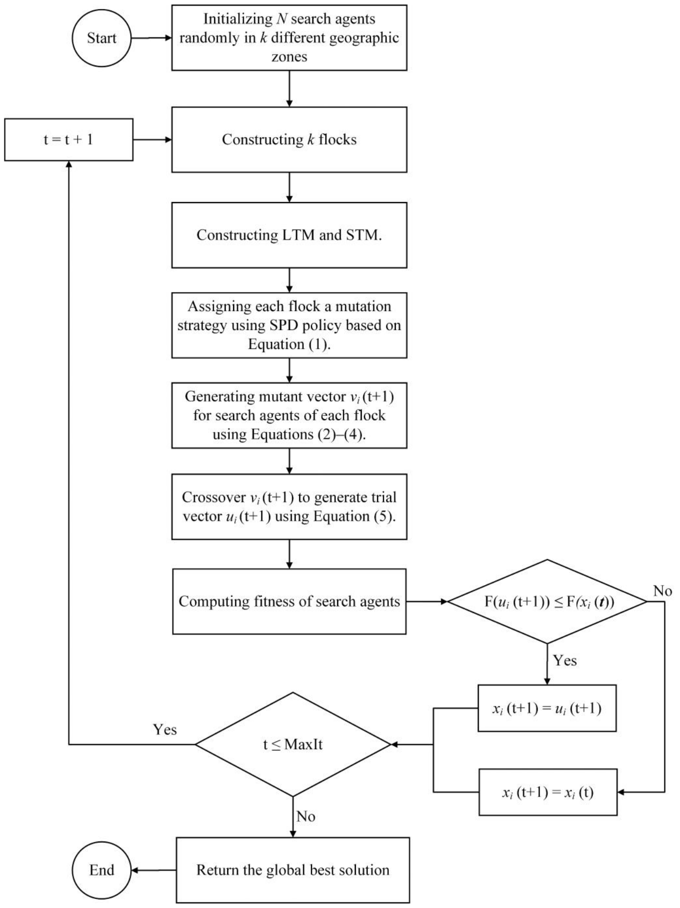
Start
Initializing N search agents randomly in k different geographic zones
t = t + 1
Constructing k flocks
Constructing LTM and STM.
Assigning each flock a mutation strategy using SPD policy based on Equation (1).
Generating mutant vector v_i(t+1) for search agents of each flock using Equations (2)–(4).
Crossover v_i(t+1) to generate trial vector u_i(t+1) using Equation (5).
Computing fitness of search agents
F(u_i (t+1)) ≤ F(x_i (t))
No
Yes
x_i (t+1) = u_i (t+1)
Yes
t ≤ MaxIt
x_i (t+1) = x_i (t)
No
End
Return the global best solution

Figure 2. Flowchart of the QANA.

4. The Proposed Binary QANA

According to the previous study [73], QANA outperforms other well-known optimizers in various continuous search space benchmark tests. In comparison to its competitors, QANA outperforms them in terms of exploration and exploitation abilities. Hence, the main components of the standard QANA are utilized to develop binary QANA for solving the FS problem. To develop binary QANA, initially, solutions are randomly generated in the range [0, 1]. The iterative procedure is continued after initialization until the stopping condition (maximum number of iterations) is met. In each iteration, the positions of search agents are mapped to binary using the transfer function (first approach) or by assigning a threshold for each variable (second approach). In the first approach, 20 different transfer functions from five categories, S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic, are applied to map the continuous solutions of the canonical QANA to binary ones. While in the second approach, the QANA is converted to binary by simply assigning a threshold for each dimension. Both approaches are described and investigated in more detail in the following subsections.

4.1. Binary QANA Development Using Different Transfer Functions

In accordance with the literature, the transfer function has a crucial role in mapping continuous solutions to discrete space. The output of a transfer function is in the range of [0, 1]. The value of the search agent's position determines the probability of changing the solution of the previous iteration, where the transfer function has to provide a large enough probability of changing the previous solution for a higher value of the search agent's position. On the other hand, the computed probability of changing the solution should also be low if the value is low. Based on the above discussion, choosing a suitable transfer function will enhance the algorithm's performance in solving the FS problem. Hence, in this study, the four versions of each transfer function S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic are discussed and applied to develop different variants of binary QANA.

4.1.1. Binary QANA Using S-Shaped Transfer Function (S-BQANA)

The S-shaped transfer function first used in BPSO [76] employs the sigmoid function (S_2) to map continuous position into binary form based on Equation (7),

$$TF_S\left(x_i^d(t+1)\right) = 1/(1+\exp^{-x_i^d(t)}) \tag{7}$$

where $x_i^d(t)$ is the position of the i-th search agent in the d-th dimension at the current iteration. The new position of the search agent is then updated using Equation (8), where r is a random number in [0, 1].

$$b_i^d(t+1) = \begin{cases} 1, & if \;\; r < TF_s\left(x_i^d(t+1)\right) \\ 0, & \quad if \;\; r \geq TF_s\left(x_i^d(t+1)\right) \end{cases} \tag{8}$$

S_2 and three other variants of S-shaped transfer functions are presented visually in Figure 3 and mathematically in Table 1. As shown in Figure 3, S_1 sharply grows and hits saturation as the value of the position increases substantially higher than S_2, while S_3 and S_4 saturations begin later than S_2. Hence, among these four versions of the S-shaped transfer function, S_1 generates the highest probability for the same value, while S_4 returns the lowest value.

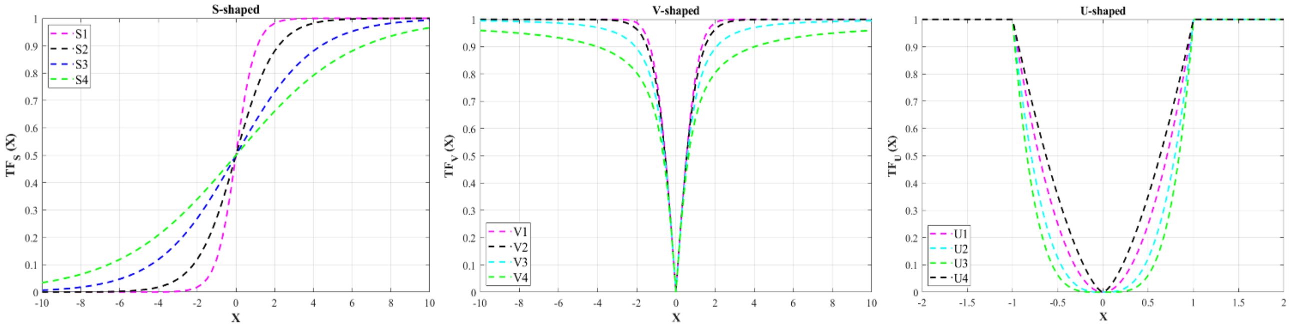

Figure 3. The S-shaped, V-shaped, and U-shaped transfer functions.

Table 1. The formulation of S-shaped, V-shaped, and U-shaped family transfer functions.

No.	S-Shaped Transfer Functions	No.	V-Shaped Transfer Functions	No.	U-Shaped Transfer Functions
S_1	$TF_S(x) = 1/(1+\exp^{-2x})$	V_1	$TF_V(x) = \left\lvert erf\left(\frac{\sqrt{\pi}}{2}x\right)\right\rvert$	U_1	$TF_U(x) = \alpha\lvert x^{1.5}\rvert$
S_2	$TF_S(x) = 1/(1+\exp^{-x})$	V_2	$TF_V(x) = \lvert tan\,h(x)\rvert$	U_2	$TF_U(x) = \alpha\lvert x^{2}\rvert$
S_3	$TF_S(x) = 1/(1+\exp^{-x/2})$	V_3	$TF_V(x) = \left\lvert (x)/\sqrt{1+x^2}\right\rvert$	U_3	$TF_U(x) = \alpha\lvert x^{3}\rvert$
S_4	$TF_S(x) = 1/(1+\exp^{-x/3})$	V_4	$TF_V(x) = \left\lvert \frac{2}{\pi}arctan\left(\frac{\pi}{2}x\right)\right\rvert$	U_4	$TF_U(x) = \alpha\lvert x^{4}\rvert$

The S-shaped transfer function has certain flaws. In SI algorithms, if the value is 0, the next solution remains the same as the present position. To put it another way, the 0 value indicates that the new location should not be modified. However, with a chance of 0.5, the new position in the S-shaped transfer function may be modified to 0 or 1. Also, in the SI algorithms, there is no difference between large positive or negative values, as a large absolute position value implies that the present search agent's location is insufficient and that a significant movement is necessary to attain the ideal position. A tiny absolute value also indicates that the present search agent's location is near the ideal solution and that only a small distance is required to reach it. However, in S-shaped transfer functions, a positive value results in a higher likelihood (probability of 1), whereas a negative value results in a probability of 0 for the following particles' location, which is in contrast with the natural movements of SI algorithms.

4.1.2. Binary QANA Using V-Shaped Transfer Function (V-BQANA)

The V-shaped transfer function (hyperbolic) is a symmetric function introduced by Rashdi et al. [77] to develop a binary version of the gravity search algorithm named BGSA. According to Equation (9), this function calculates the probability of changing the value of each dimension based on the position of each search agent in continuous search space,

$$TF_v\left(x_i^d(t+1)\right) = \left\lvert tanh\left(x_i^d(t)\right)\right\rvert \tag{9}$$

where $x_i^d(t)$ indicates the position value of the i-th search agent in the d-th dimension at the current iteration. As illustrated in Equation (10), the position updating rules of V-shaped transfer functions are quite different from S-shaped transfer functions,

$$b_i^d(t+1) = \begin{cases} \left(b_i^d(t)\right)^{-1}, & if \quad r < TF_v\left(x_i^d(t+1)\right) \\ b_i^d(t), & if \quad r \geq TF_v\left(x_i^d(t+1)\right) \end{cases} \tag{10}$$

where $b_i^d(t)$ and $x_i^d(t)$ represent the binary position and continuous position value of the i-th search agent in the d-th dimension at the current iteration, $\left(b_i^d(t)\right)^{-1}$ is the complement of $b_i^d(t)$, and r denotes a random value in [0, 1]. Based on this rule, if the value obtained from the transfer function is equal to or greater than r, the value of the d-th dimension will change to the complement of the current binary position as the continuous value is high enough to change the current position. In contrast, the binary position of the d-th dimension remains constant if the value obtained from the transfer function is less than r. As can be seen in Table 1, three new transfer functions have been introduced by implementing different mathematical equations. According to Figure 3, transfer functions V_1, V_2, V_3, and V_4 provide the highest probability of switching search agents' positions, respectively.

Unlike the S-shaped transfer functions, V-shaped transfer functions do not require search agents to take 0 or 1 values, as they let search agents with low values remain at their current positions or switch to their complements if their value is high enough. Also, the V-shaped transfer functions solve the shortcomings of the S-shaped transfer functions by assigning 0 probability of changing the position of a search agent with zero value and considering the absolute value of the continuous position in the equations to avoid assigning a probability of 0 for search agents with negative values. Moreover, in another study, Mirjalili et al. [98] evaluated and compared different versions of sigmoid and hyperbolic functions, which showed the relative superiority of hyperbolic family functions in solving the FS problem.

4.1.3. Binary QANA Using U-Shaped Transfer Function (U-BQANA)

In a recent study, Mirjalili et al. [78] proposed a new U-shaped transfer function for the PSO algorithm to map continuous solutions to binary ones. This transfer function comes with two control parameters to modify the range of exploration and exploitation. As can be seen in Figure 3, similar to the V-shaped transfer function, U-shaped is a symmetric function, which means that it assigns 0 probability of changing the position of a search agent with 0 value. Also, as the absolute value of continuous position is considered in this transfer function, there are no differences between the positive and negative values. The mathematical formulation of the U-shaped transfer function is presented in Equations (11) and (12),

$$TF_u\left(x_i^d(t+1)\right) = \alpha\left|\left(x_i^d(t)\right)^{\beta}\right| \tag{11}$$

$$b_i^d(t+1) = \begin{cases} \left(b_i^d(t)\right)^{-1}, & if\ r < TF_u\left(x_i^d(t+1)\right) \\ b_i^d(t), & if\ \ r \geq TF_u\left(x_i^d(t+1)\right) \end{cases} \tag{12}$$

where α and β are two control parameters for determining the slope and saturation point of the U-shaped transfer function. b_i^d and x_i^d represent the binary and continuous positions of the i-th search agent in the d-th dimension, respectively. r is a uniform random value in [0, 1].

Table 1 and Figure 3 illustrate different versions of the U-shaped transfer function labeled U_1, U_2, U_3, and U_4 that were established using different values of control parameters. The α control parameter determines the U-shaped curve's saturation point. In contrast, β modifies the exploration range of the transfer function by changing the width of the U-shaped transfer function's basin. Hence, it is noticeable that U_4 provides a higher exploration range than other variations. It is also noticeable that all U-shaped variants offer higher exploration than V-shaped ones.

4.1.4. Binary QANA Using Z-Shaped Transfer Function (Z-BQANA)

The Z-shaped transfer function proposed by Guo et al. [79] is a symmetric transfer function applied to denote the probability that an element of the position vector will change from 0 to 1 in the BPSO algorithm. Based on this transfer function, when the continuous position value is 0, the probability of change should be low because when the particle reaches the best value, the continuous position value should be lowered to 0, and the probability of the particle's position change should be 0. The Z-shaped transfer function is defined mathematically based on Equations (13) and (14),

$$TF_z\left(x_i^d(t+1)\right) = \sqrt{1 - a^{x_i^d(t+1)}} \tag{13}$$

$$b_i^d(t+1) = \begin{cases} \left(b_i^d(t)\right)^{-1}, & if\ r < TF_z\left(x_i^d(t+1)\right) \\ b_i^d(t), & if\ r \geq TF_z\left(x_i^d(t+1)\right) \end{cases} \tag{14}$$

where b_i^d and x_i^d represent the binary and continuous positions of the i-th search agent in the d-th dimension, respectively, and a denotes a positive integer. A collection of Z-shaped function families is generated by modifying the value of a, the formulas and figures of which are presented in Table 2 and Figure 4, respectively. The Z-shaped transfer function is an asymmetric mapping function, as seen in Figure 4. The asymmetric mapping function essentially fulfills the absolute value to calculate the mapping probability of the particle position vector variation, resulting in a quick convergence. The function's slope varies when the parameter $D_{particle} = D_{Function} \times 15$ is changed. The lesser the slope of the function, the greater $D_{particle} = D_{Function} \times 15$. Hence, when the value remains constant, the probability of obtaining small changes in the location of the i-th particles is greater.

Table 2. The formulation of Z-shaped and quadratic family transfer functions.

No.	Z-Shaped Transfer Functions	No.	Quadratic Transfer Functions
Z$_1$	$TF_Z(x) = \sqrt{1-2^x}$	Q$_1$	$TF_Q(x) = \begin{cases} \left\lvert\frac{x}{(0.5\ x_{max})}\right\rvert, & if\ x < 0.5\ x_{max} \\ 1\quad, & if\ x \geq 0.5\ x_{max} \end{cases}$
Z$_2$	$TF_Z(x) = \sqrt{1-5^x}$	Q$_2$	$TF_Q(x) = \begin{cases} \left(\frac{x}{(0.5\ x_{max})}\right)^2, & if\ x < 0.5\ x_{max} \\ 1\quad, & if\ x \geq 0.5\ x_{max} \end{cases}$
Z$_3$	$TF_Z(x) = \sqrt{1-8^x}$	Q$_3$	$TF_Q(x) = \begin{cases} \left(\frac{x}{(0.5\ x_{max})}\right)^3, & if\ x < 0.5\ x_{max} \\ 1\quad, & if\ x \geq 0.5\ x_{max} \end{cases}$
Z$_4$	$TF_Z(x) = \sqrt{1-20^x}$	Q$_4$	$TF_Q(x) = \begin{cases} \left(\frac{x}{(0.5\ x_{max})}\right)^{\frac{1}{2}}, & if\ x < 0.5\ x_{max} \\ 1\quad, & if\ x \geq 0.5\ x_{max} \end{cases}$

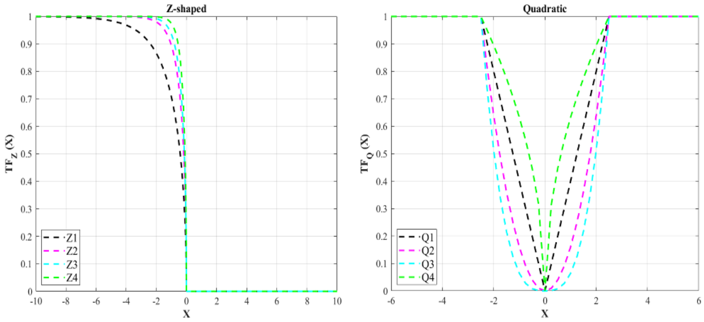

Figure 4. The Z-shaped and quadratic transfer functions.

4.1.5. Binary QANA Using Quadratic Transfer Function (Q-BQANA)

The quadratic transfer function proposed by Rezaee Jordehi [81] is a recent transfer function used for converting continuous solutions of the PSO to binary ones based on Equations (15) and (16),

$$TF_Q\left(x_i^d(t+1)\right) = \begin{cases} \left(\frac{x_i^d(t)}{0.5\ x_{max}}\right)^2, & if\ x_i^d(t) < 0.5\ x_{max} \\ 1, & if\ x_i^d(t) \geq 0.5\ x_{max} \end{cases} \tag{15}$$

$$b_i^d(t+1) = \begin{cases} \left(b_i^d(t)\right)^{-1}, & if\ r < TF_Q\left(x_i^d(t+1)\right) \\ b_i^d(t), & if\ r \geq TF_Q\left(x_i^d(t+1)\right) \end{cases} \tag{16}$$

where TF_Q denotes the quadratic transfer function and b_i^d and x_i^d represent the binary and continuous positions of the i-th search agent in the d-th dimension, respectively. The variable r is a random number in [0, 1]. The three other variants of the quadratic transfer function [80] are presented mathematically in Table 2 and visualized in Figure 4.

4.2. Binary QANA Development Using Variable Threshold (BQANA)

The previous subsections introduced different variants of binary QANA based on five different categories of transfer functions. Although transfer functions are widely used in the literature of FS, they impose an additional computational cost. Furthermore, transfer functions cannot provide superior results for every metaheuristic algorithm, especially for high-dimensional datasets. On the other hand, the QANA proved to be an effective problem solver in solving high-dimensional problems as it provides adequate search space coverage [73]. It is expected that the BQANA developed based on the second approach can generate suitable candidates for solving the FS problem. Hence, this section proposes the superior version of binary QANA, named BQANA, by simply using a threshold for assigning continuous solutions of the QANA into binary. In this approach, the generated continuous solutions are converted to binary form based on Equation (17),

$$b_i^d(t+1) = \begin{cases} 1, & if\ \ x_i^d(t) > 0.5 \\ 0, & if\ \ x_i^d(t) \leq 0.5 \end{cases} \tag{17}$$

where b_i^d is the binary solution of the i-th search agent in the d-th dimension, $x_i^d(t)$ denotes the continuous position of the i-th search agent in the d-th dimension at iteration t. The general procedure of selecting effective features with BQANA is illustrated in Figure 5, where the algorithm receives the dataset with all features as input and returns an optimal feature subset as output.

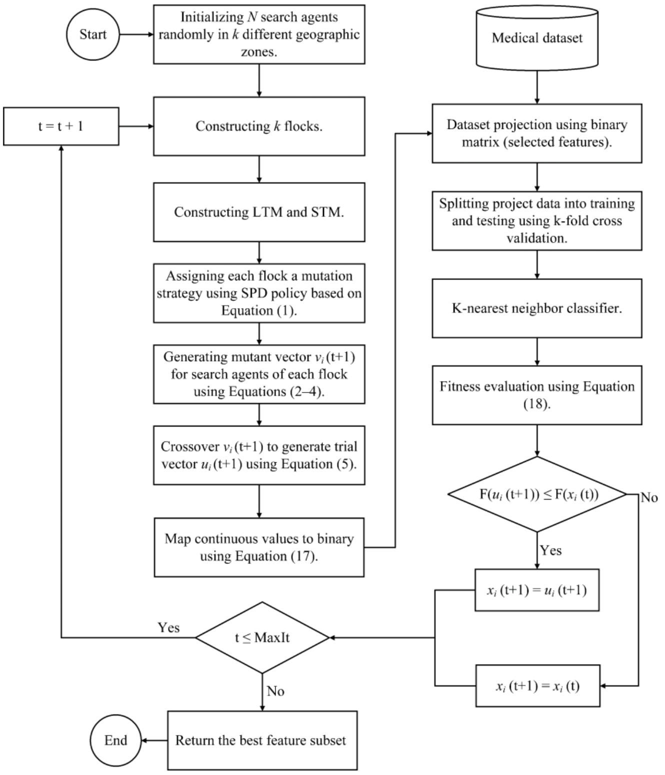
Start
Initializing N search agents randomly in k different geographic zones.
t = t + 1
Constructing k flocks.
Constructing LTM and STM.
Assigning each flock a mutation strategy using SPD policy based on Equation (1).
Generating mutant vector v_i(t+1) for search agents of each flock using Equations (2–4).
Crossover v_i(t+1) to generate trial vector u_i(t+1) using Equation (5).
Map continuous values to binary using Equation (17).
Medical dataset
Dataset projection using binary matrix (selected features).
Splitting project data into training and testing using k-fold cross validation.
K-nearest neighbor classifier.
Fitness evaluation using Equation (18).
F(u_i (t+1)) ≤ F(x_i (t))
No
Yes
x_i (t+1) = u_i (t+1)
x_i (t+1) = x_i (t)
Yes
t ≤ MaxIt
No
Return the best feature subset
End

Figure 5. Flowchart of the proposed BQANA using a variable threshold.

5. Experimental Assessment

In this section, the performance of the proposed binary QANA approaches for solving the FS problem is experimentally assessed on 10 medical datasets of various sizes, which are described in Table 3. Also, the parameters of the algorithms used in this experiment are shown in Table 4. In the first approach, the canonical QANA is converted to binary using 20 different transfer functions from five categories of S-shaped [76], V-shaped [77], U-shaped [78], Z-shaped [79], and quadratic [80,81] to solve FS problem. The comparison results related to different variants of the first approach are tabulated in Tables A1–A5. In the second approach, the QANA is converted to binary by assigning a threshold for each dimension to map the continuous solutions into binary without further computational cost. To select the best algorithms from the first approach, one algorithm is considered representative of each transfer function category. Then, the five selected algorithms are compared against the BQANA developed based on the second approach in Table 5. Ultimately, Table 6 presents a comparison between the proposed BQANA and nine well-known metaheuristic algorithms introduced in the literature, including binary particle swarm optimization (BPSO) [82], ant colony optimization (ACO) [83], binary deferential evolution (BDE) [84], binary bat algorithm (BBA) [85], feature selection based on whale optimization algorithm (FSWOA) [11], binary ant lion optimizer (BALO) [86], binary dragonfly algorithm (BDA) [87], quadratic binary Harris hawk optimization (QBHHO) [80], and binary atom search optimization (BASO) [88]. In Tables 5, 6 and A1–A5, the bold values indicate the winning algorithms, and at the end of each table, the overall comparisons are shown based on the numbers of the wins (W), ties (T), and losses (L).

Table 3. Datasets specifications.

No.	Medical Datasets	No. Samples	No. Features	Classes	Size
1	Pima	768	8	2	Small
2	HeartEw	270	13	2	Small
3	Lymphography	148	18	4	Small
4	SPECT Heart	267	22	2	Small
5	PenglungEW	73	325	7	Medium
6	Parkinson	756	754	2	Medium
7	Colon	62	2000	2	Large
8	SRBCT	83	2308	4	Large
9	Leukemia	72	7129	4	Large
10	Prostate tumor	102	10509	2	Large

Table 4. Parameters of the algorithms.

Algorithm	Parameter Settings
BPSO	$c_1 = c_2 = 2$, w = [0.9 to 0.4].
ACO	$\tau = 1$, $\alpha = 1$, ρ = 0.2, $\beta = 0.1$, $\eta = 1$.
BDE	$Cr = 0.9$.
BBA	$A = 0.9$, $r = 0.9$, $Q_{min} = 0$, $Q_{max} = 2$.
FSWOA	a is linearly decreased from 2 to 0, a_2 = is linearly decreased from -1 to -2, $b = 1$.
BALO	V-shaped transfer function.
BDA	$D_{max} = 6$.
QBHHO	$\beta = 1.5$, Q_4 transfer function, and $x_{max} = 5$.
BASO	$\alpha = 50$, $\beta = 0.2$.
BQANA	The number of flocks (k) = 2, LTM size (K') = 2, and STM size (K'') = 10.

Table 5. The comparison between the BQANA and winners of each family.

Datasets	**Metrics**		**S_4**	**V_2**	**U_4**	**Z_1**	**Q_3**	**BQANA**
Pima	Fitness	Avg	0.2386	0.2389	0.2393	0.2350	0.2380	**0.2316**
		Min	0.2343	0.2332	0.2318	0.2281	0.2319	0.2291
	Accuracy	Avg	76.5053	76.4863	76.3969	76.8837	76.4997	77.2076
		Max	76.9617	76.9532	77.0933	77.4675	77.0762	77.4863
HeartEW	Fitness	Avg	0.1493	0.1513	0.1493	0.1496	0.1509	**0.1384**
		Min	0.1395	0.1432	0.1432	0.1395	0.1416	0.1329
	Accuracy	Avg	85.3333	85.1111	85.2778	85.4074	85.0741	86.4259
		Max	86.2963	85.9259	85.9259	86.2963	85.9259	87.0370
Lymphography	Fitness	Avg	0.1434	0.1425	0.1423	0.1273	0.1432	**0.1128**
		Min	0.1324	0.1247	0.1258	0.1049	0.1303	0.1008
	Accuracy	Avg	86.0595	86.1048	86.1690	87.7024	85.9952	89.1310
		Max	87.1905	87.9048	87.8571	89.9048	87.2857	90.3810
SPECT Heart	Fitness	Avg	0.2503	0.2457	0.2468	0.2450	0.2453	**0.2231**
		Min	0.2415	0.2306	0.2260	0.2288	0.2326	0.2038
	Accuracy	Avg	75.1709	75.6019	75.4623	75.7543	75.5648	77.8725
		Max	76.0969	77.1652	77.5356	77.2080	76.8234	79.7863
PenglungEW	Fitness	Avg	0.1014	0.0993	0.1000	0.1012	0.0977	**0.0765**
		Min	0.0830	0.0842	0.0935	0.0805	0.0828	0.0545
	Accuracy	Avg	90.2946	90.4286	90.3393	90.3929	90.5357	92.6429
		Max	92.1429	91.9643	90.8929	92.3214	91.9643	94.8214
Parkinson	Fitness	Avg	0.2341	0.2200	0.2011	0.2308	0.2049	**0.1604**
		Min	0.2160	0.1882	0.1865	0.1949	0.1838	0.1279
	Accuracy	Avg	76.8834	78.1005	79.9472	77.2476	79.6068	83.9765
		Max	78.7018	81.4895	81.4877	80.8175	81.7456	87.1807
Colon	Fitness	Avg	0.1012	0.0994	0.0995	0.1007	0.0947	**0.0775**
		Min	0.0949	0.0852	0.0916	0.0874	0.0809	0.0481
	Accuracy	Avg	90.3214	90.4286	90.4167	90.4405	90.7976	92.5000
		Max	90.9524	91.9048	91.1905	91.6667	92.1429	95.2381
SRBCT	Fitness	Avg	*0.0129*	0.0120	0.0101	0.0116	0.0047	**0.0040**
		Min	0.0053	0.0038	0.0021	0.0045	0.0018	0.0003
	Accuracy	Avg	99.2361	99.2222	99.3819	99.3750	99.8333	99.8333
		Max	100.00	100.00	100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg	0.1010	0.1000	0.0960	0.0969	0.0915	**0.0638**
		Min	0.0884	0.0827	0.0690	0.0825	0.0702	0.0426
	Accuracy	Avg	90.3393	90.3661	90.7143	90.8304	91.0714	93.9196
		Max	91.6071	92.1429	93.3929	92.1429	93.2143	96.2500
Prostate Tumor	Fitness	Avg	0.1219	0.1201	0.1147	0.1195	0.1113	**0.0534**
		Min	0.1044	0.1074	0.0984	0.1014	0.0882	0.0199
	Accuracy	Avg	88.2318	88.2864	88.7136	88.5409	89.0818	94.7773
		Max	90.0000	89.5455	90.0909	90.2727	91.2727	98.0000
Overall Results	W|T|L		0|0|10	0|0|10	0|0|10	0|0|10	0|0|10	10|0|0

Table 6. The comparison between the BQANA and comparative algorithms.

Datasets	Metrics		BPSO	ACO	BDE	BBA	FSWOA	BALO	BDA	QBHHO	BASO	BQANA
Pima	Fitness	Avg	0.2329	0.2380	0.2324	0.2327	0.2374	0.2323	0.2317	0.2337	0.2353	**0.2316**
		Min	0.2304	0.2317	0.2292	0.2267	0.2266	0.2499	0.2280	0.2266	0.2243	0.2291
	Accuracy	Avg	77.0224	76.519	77.2101	77.0195	76.6752	77.0936	77.1855	76.9329	76.7359	77.2076
		Max	77.3599	77.211	77.4812	77.6094	77.0796	77.6196	77.4744	77.6145	77.8520	77.4863
	No. features	Avg	4.300	4.400	5.400	4.450	5.200	4.450	4.450	4.250	4.000	4.750
		Min	4.000	4.000	4.000	4.000	4.000	4.000	4.000	4.000	2.000	4.000
HeartEW	Fitness	Avg	0.1414	0.1489	0.1390	0.1408	0.1548	0.1407	0.1386	0.1417	0.1426	**0.1384**
		Min	0.1358	0.1395	0.1308	0.1358	0.1468	0.1358	0.1308	0.1351	0.1380	0.1329
	Accuracy	Avg	86.1296	85.333	86.5370	86.1481	84.9074	86.1852	86.426	86.0370	85.9074	86.4259
		Max	86.6667	86.296	87.4074	86.6667	85.5556	86.6667	87.407	86.6667	86.6667	87.0370
	No. features	Avg	5.300	4.150	7.450	4.950	7.050	5.150	5.450	4.450	4.000	5.250
		Min	3.000	3.000	5.000	3.000	4.000	3.000	4.000	3.000	3.000	3.000
lymphography	Fitness	Avg	0.1194	0.1450	0.1246	0.1524	0.1380	0.1146	0.1154	0.1266	0.1275	**0.1128**
		Min	0.1046	0.1303	0.1050	0.1375	0.1252	0.1055	0.1054	0.1130	0.1135	0.1008
	Accuracy	Avg	88.4929	85.900	88.1000	85.0714	86.6905	88.9476	88.871	87.6500	87.7167	89.1310
		Max	90.0000	87.286	90.0000	86.6190	87.9524	89.9048	89.952	89.0952	89.0952	90.3810
	No. features	Avg	9.650	9.600	12.150	8.150	11.300	9.350	9.450	7.650	10.600	9.400
		Min	6.000	5.000	10.000	5.000	9.000	6.000	7.000	6.000	6.000	6.000
SPECT Heart	Fitness	Avg	0.2291	0.2488	0.2298	0.2339	0.2574	**0.2209**	0.2230	0.2308	0.2354	0.2231
		Min	0.2091	0.2348	0.1994	0.2226	0.2458	0.2069	0.2012	0.2150	0.2219	0.2038
	Accuracy	Avg	77.2877	75.234	77.3440	76.6254	74.6197	78.1346	77.900	76.9708	76.5021	77.8725
		Max	79.3875	76.695	80.2279	77.9772	75.7265	79.4729	80.185	78.6467	77.9060	79.7863
	No. features	Avg	9.150	7.800	12.050	9.450	13.500	9.650	9.200	6.100	6.200	8.850
		Min	6.000	2.000	6.000	4.000	10.000	7.000	4.000	2.000	2.000	2.000
PenglungEW	Fitness	Avg	0.0895	0.0977	0.0936	0.0915	0.1093	0.0898	0.0826	0.0843	0.0895	**0.0765**
		Min	0.0807	0.0816	0.0735	0.0832	0.0878	0.0803	0.0681	0.0695	0.0746	0.0545
	Accuracy	Avg	91.4554	90.446	91.3304	91.2143	89.6607	91.7321	92.143	91.7411	91.2500	92.6429
		Max	92.3214	91.964	93.2143	91.9643	91.7857	93.3929	93.571	93.2143	93.0357	94.8214
	No. features	Avg	158.650	98.650	252.750	148.400	225.950	159.550	155.15	80.950	94.400	120.650
		Min	139.000	34.000	204.000	118.000	212.000	143.000	131.00	28.000	38.000	47.000
Parkinson	Fitness	Avg	0.2033	0.1813	0.2512	0.2217	0.2541	0.1953	0.1938	0.1673	0.1703	**0.1604**
		Min	0.1754	0.1628	0.2429	0.1768	0.2500	0.1556	0.1549	0.1546	0.1489	0.1279
	Accuracy	Avg	79.9612	81.813	75.3543	78.0702	75.0166	80.7618	80.902	83.1542	82.9234	83.9765
		Max	82.8018	83.595	76.3228	82.5474	75.3912	84.7895	84.795	84.4123	85.0386	87.1807
	No. features	Avg	367.450	90.550	540.100	355.000	511.650	365.650	358.40	37.450	96.550	130.100
		Min	331.000	7.000	402.000	299.000	184.000	325.000	332.00	8.000	46.000	16.000
Colon	Fitness	Avg	0.0892	0.0983	0.1020	0.0913	0.1098	0.0898	0.0925	0.0821	0.0905	**0.0775**
		Min	0.0710	0.0795	0.0882	0.0684	0.0969	0.0803	0.0776	0.0501	0.0625	0.0481
	Accuracy	Avg	91.4881	90.381	90.5357	91.2381	89.6071	91.4286	91.143	91.9286	91.0714	92.5000
		Max	93.3333	92.143	91.9048	93.5714	90.7143	92.3810	92.619	95.0000	93.8095	95.2381
	No. features	Avg	993.050	534.15	1651.50	918.250	1386.20	988.950	954.20	431.900	422.950	644.300
		Min	963.000	36.000	1355.00	636.000	989.000	939.000	848.00	117.000	164.000	80.000

Table 6. *Cont.*

Datasets	Metrics		BPSO	ACO	BDE	BBA	FSWOA	BALO	BDA	QBHHO	BASO	BQANA
SRBCT	Fitness	Avg	0.0048	0.0061	0.0097	0.0064	0.0241	0.0047	0.0052	**0.0006**	0.0012	0.0040
		Min	0.0046	0.0010	0.0057	0.0025	0.0179	0.0042	0.0039	0.0002	0.0009	0.0003
	Accuracy	Avg	100.000	99.604	99.7153	99.7222	98.2500	100.000	99.938	100.000	100.000	99.8333
		Max	100.000	100.00	100.000	100.000	98.8889	100.000	100.00	100.000	100.000	100.000
	No. features	Avg	1102.00	483.35	1597.20	957.900	1568.70	1077.80	1048.0	132.450	277.850	551.850
		Min	1068.00	138.00	1327.00	566.000	979.000	972.000	4.000	57.000	208.000	77.000
Leukemia	Fitness	Avg	0.0844	0.0925	0.0937	0.0893	0.1105	0.0802	0.0768	0.0706	0.0744	**0.0638**
		Min	0.0739	0.0708	0.0766	0.0700	0.1007	0.0580	0.0560	0.0560	0.0438	0.0426
	Accuracy	Avg	91.9821	90.92	91.3036	91.3482	89.5446	92.4018	92.732	93.0179	92.6786	93.9196
		Max	93.0357	93.036	93.0357	93.3929	90.5357	94.6429	94.821	94.4643	95.7143	96.2500
	No. features	Avg	3552.30	1706.9	5421.20	3270.30	5003.65	3532.25	3451.6	1073.15	1385.95	2546.800
		Min	3496.00	68.000	4057.00	2251.00	4723.00	3451.00	3108.0	315.000	737.000	536.000
Prostate Tumor	Fitness	Avg	0.1026	0.0997	0.1024	0.1047	0.1263	0.0997	0.1009	0.0561	0.0880	**0.0534**
		Min	0.0922	0.0678	0.0828	0.0902	0.1132	0.0814	0.0806	0.0289	0.0704	0.0199
	Accuracy	Avg	90.1409	90.000	90.4864	89.8182	87.9545	90.4364	90.300	94.3500	91.2409	94.7773
		Max	91.1818	93.182	92.2727	91.3636	89.2727	92.2727	92.364	97.0909	93.0000	98.0000
	No. features	Avg	5238.15	786.25	8669.55	4655.00	7425.75	5247.40	5129.7	206.250	1342.50	1833.300
		Min	5157.00	83.000	6645.00	2420.00	7230.00	5138.00	4809.0	65.000	1012.00	82.000
Overall Results	W | T | L		0 | 0 | 10	0 | 0 | 10	0 | 0 | 10	0 | 0 | 10	0 | 0 | 10	1 | 0 | 9	0 | 0 | 10	1 | 0 | 9	0 | 0 | 10	8 | 0 | 2

The comparison tables show the average fitness, minimum fitness, average classification accuracy, maximum classification accuracy, average number of selected features, and minimum number of selected features obtained by each algorithm. The average number of selected features by each algorithm from different datasets with various sizes is visualized in Figures 6 and 7. Also, as classification accuracy is the most important criterion in medical datasets, the boxplot results of 10 different algorithms are exhibited in Figure 8. Furthermore, the convergence curves of fitness values obtained during the optimization process are visualized in Figure 9. Ultimately, the nonparametric Friedman test [105] was used to rank the significance of the algorithms based on their performance in minimizing the fitness, as is shown in Table 7 and Figure 10.

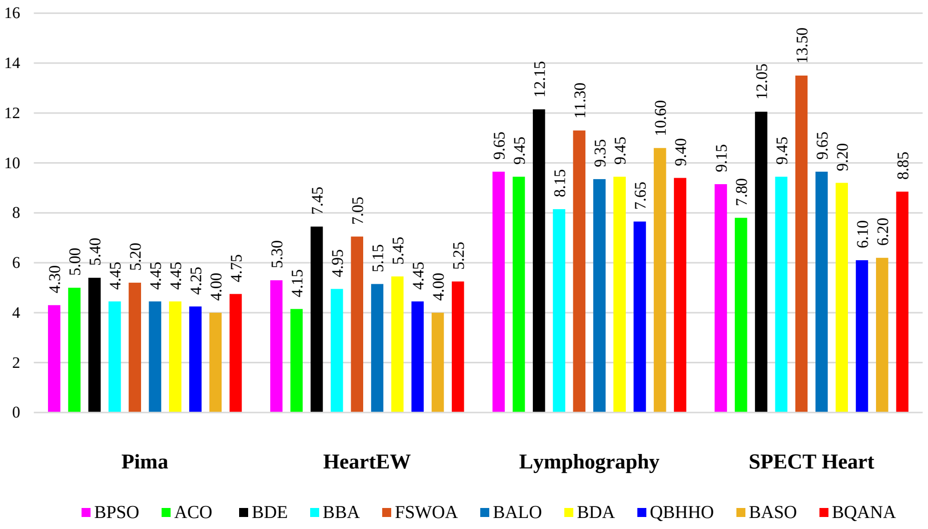

Figure 6. The average number of features selected by BQANA and comparative algorithms on small datasets.

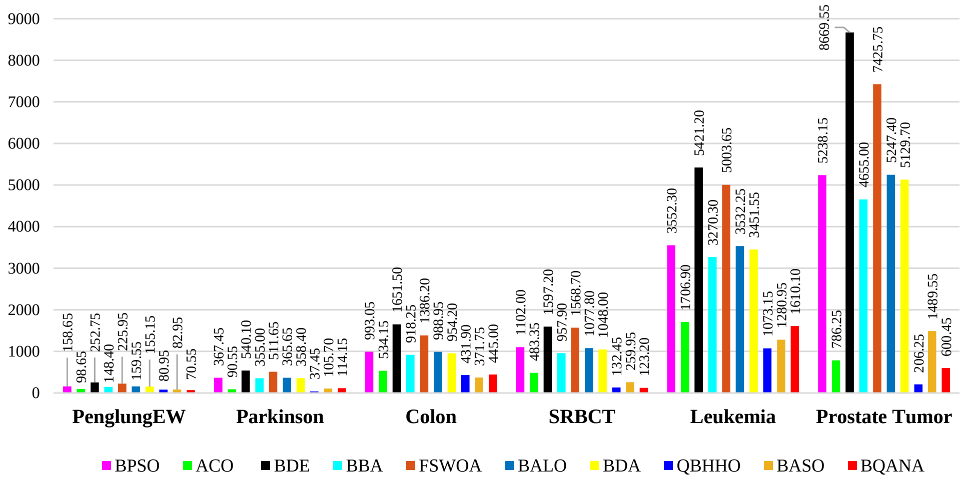

Figure 7. The average number of features selected by BQANA and comparative algorithms on medium and large datasets.

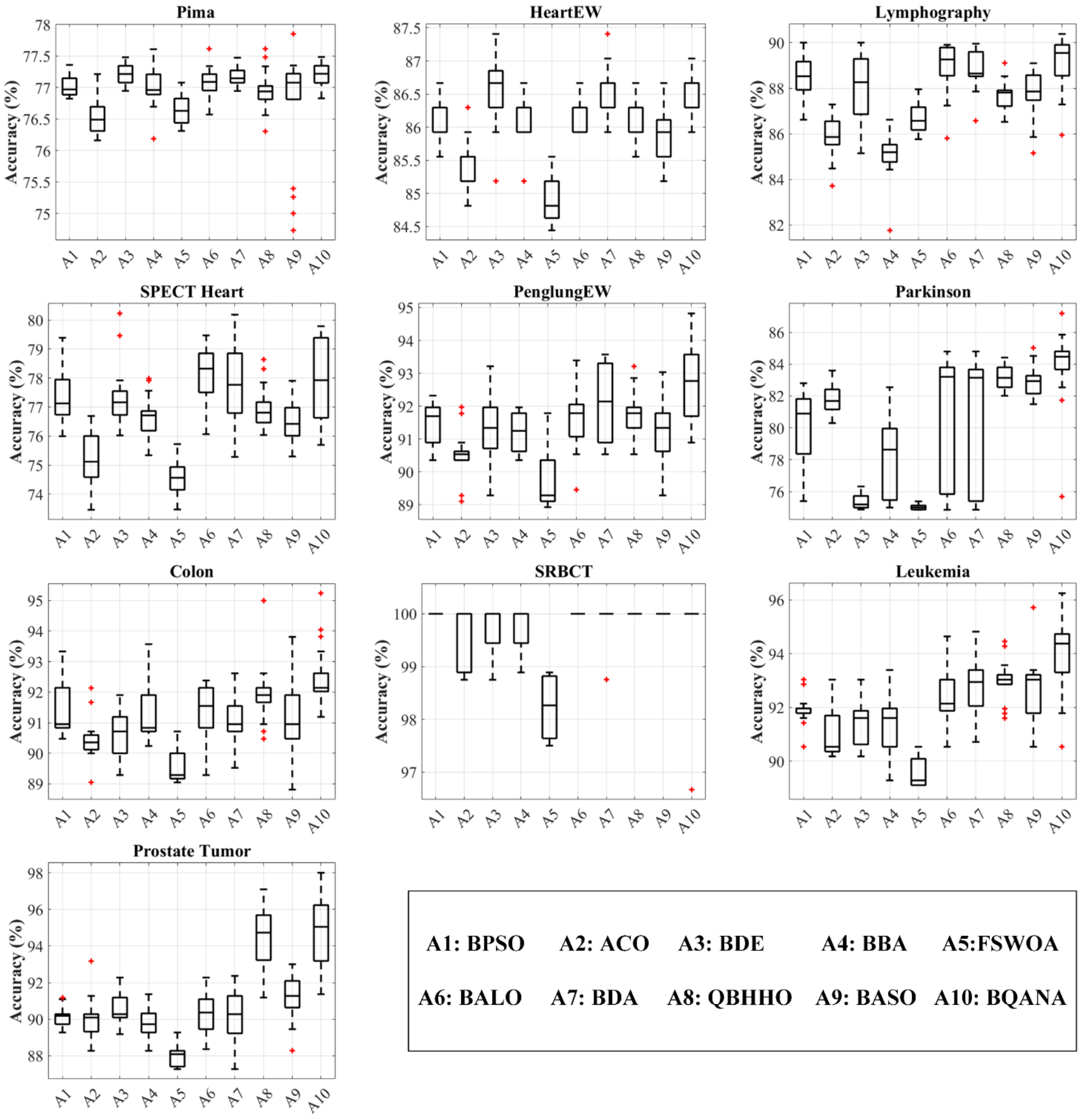

Figure 8. Boxplot of accuracy rate obtained by BQANA and comparative algorithms.

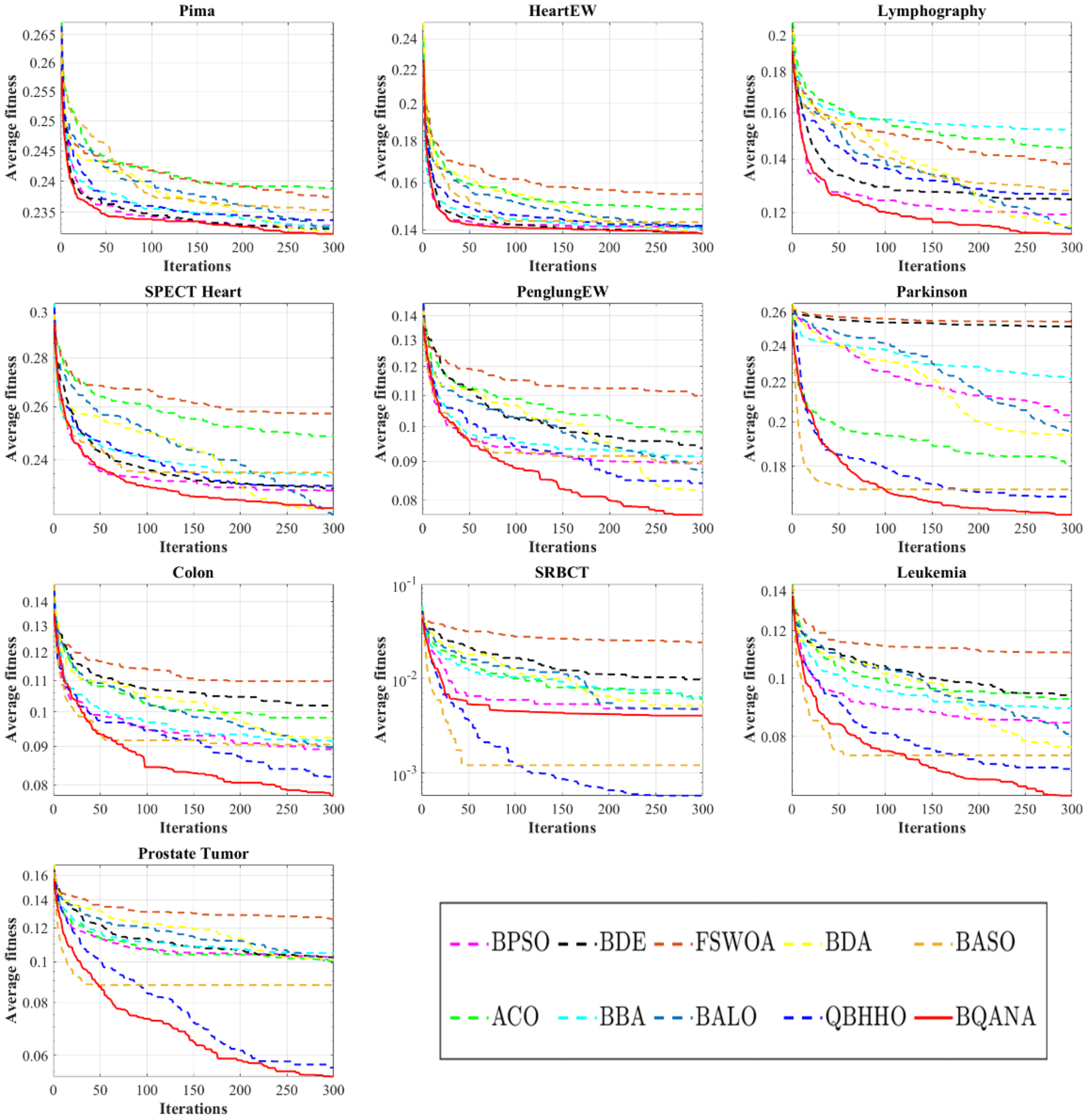

Figure 9. Convergence behavior of the BQANA and comparative algorithms.

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

Datasets	BPSO	ACO	BDE	BBA	FSWOA	BALO	BDA	QBHHO	BASO	BQANA
Pima	5.73	9.70	3.65	5.18	8.90	4.33	2.85	6.85	5.90	1.93
HeartEw	5.97	8.95	2.92	4.57	10.00	4.82	2.20	6.30	7.25	2.00
Lymphography	3.95	9.05	5.35	9.95	7.90	2.63	2.65	6.00	6.05	1.48
SPECT Heart	4.35	9.00	4.85	6.95	10.00	1.85	2.45	4.95	7.80	2.80
PenglungEW	5.45	8.55	7.90	6.95	10.00	4.35	2.95	2.90	4.85	1.10
Parkinson	6.35	5.15	9.10	7.65	9.90	5.10	5.65	2.10	2.80	1.20
Colon	4.55	8.05	8.90	5.95	10.00	4.75	5.45	2.00	4.35	1.00
SRBCT	6.85	5.20	8.85	5.65	9.95	6.30	5.85	1.00	2.30	3.05
Leukemia	6.25	7.90	8.75	6.90	10.00	5.00	3.05	2.00	3.75	1.40
Prostate Tumor	6.80	5.65	6.90	7.85	10.00	5.50	6.15	1.65	3.15	1.35
Average rank	5.62	7.72	6.71	6.76	9.66	4.46	3.92	3.57	4.82	1.73
Overall rank	6	9	7	8	10	4	3	2	5	1

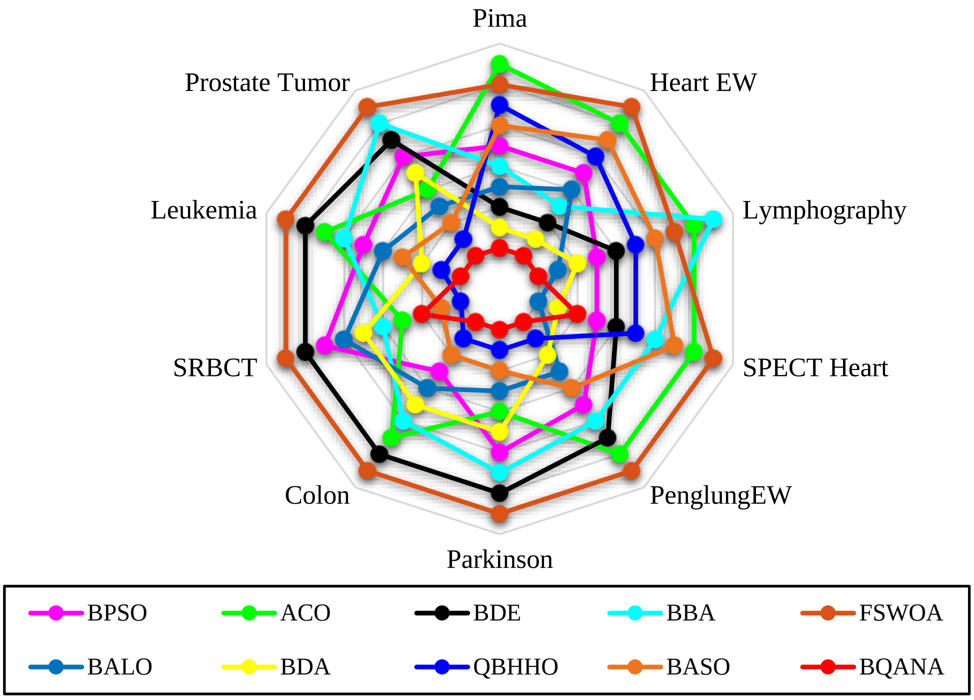

Figure 10. The rank of BQANA and comparative algorithms for feature selection problems.

5.1. Medical Datasets Description

In this study, 10 medical benchmark datasets, mostly from the UCI machine learning repository, are utilized to evaluate the performance of the proposed BQANA and comparative algorithms in solving the FS problem. The benchmark datasets utilized in the experimental evaluation of this study are on an ordinal scale, as common in the literature. Datasets with non-ordinal features can be encoded in the pre-processing stage [106]. Table 3 provides the specifics of the utilized datasets in terms of the number of samples, total number of features, number of classes, and size that is considered small if $N_f < 300$, medium if $300 \leq N_f < 1000$, and considered large if $N_f \geq 1000$, where N_f is the number of features. To avoid overfitting problems, the K-fold cross-validation method divides datasets into k folds where $k_{fold} = 10$. In this method, the classifier uses one fold as the testing set and $k - 1$ folds as the training sets.

The Pima Indian Diabetes dataset aims to diagnose diabetes based on medical examination of females at least 21 years old and being tested for diabetes [107]. The HeartEW dataset [108] predicts the absence or presence of heart disease based on data gathered from 270 samples, 120 samples with a heart problem, and the remaining are healthy. The Lymphography dataset [108] has 18 predictor features and 148 cases, with four distinct values for the class label. The aim of diagnosing cardiac single proton emission computed tomography (SPECT) heart dataset is to discriminate between the normal and abnormal function of patients' hearts using 267 image sets. The PenglungEW is a medium dataset consisting of 73 samples and 325 features with seven different classes. The Parkinson's dataset describes diagnosing healthy persons from those with Parkinson's disease. This dataset is built up of various biological voice measurements with 22 features. The Colon dataset aims to classify tissues as cancerous or normal based on data gathered from 62 colon tissue samples with 2000 genes [109]. There are 83 samples in the small round blue-cell tumor (SRBCT) dataset, each containing 2308 genes. The four classes of this dataset are the Ewing family of tumors (EWS), Burkitt lymphoma (BL) tumors, rhabdomyosarcoma (RMS) tumors, and neuroblastoma (NB) tumors [110]. The data for the Leukemia dataset came from publicly accessible microarray gene data [111]. The bone marrow expressions of 72 samples with 7128 genes are included. The dataset contains two different kinds of Leukemia classifications. The prostate tumor [112] is the largest dataset used in our experiments that contains 10,509 genes from 52 prostate cancers and 50 non-tumor prostate tissues.

5.2. Parameter Settings

In this study, the error rate is calculated using the k-nearest neighbor (KNN) algorithm with Euclidean distance and $k = 5$ to evaluate the effectiveness of selected feature subsets. The objectives of this study are evaluated by a fitness function presented in Equation (18), where CE denotes the classification error, α is the significance of classification quality, N_{sf} and N_{tf} are the numbers of selected features, and the total features of the dataset, respectively. As classification accuracy is the most important factor for medical datasets, we considered $\alpha = 0.99$ for this study.

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

To verify that the comparisons are accurate and fair, all experiments are conducted 20 times independently on a laptop with an Intel Core i7-10750H CPU and 24.0 GB of RAM using MATLAB R2022a. The maximum number of iterations ($MaxIt$) and the population size (N) were set to 300 and 20, respectively. Furthermore, the competing algorithms' parameters were adjusted to the same values as the stated settings in their works, shown in Table 4.

6. Discussion

Tables A1–A5 show the comparison results of applying different transfer functions to develop different binary versions of the QANA based on the first approach. The results indicate that S4, V2, U4, Z1, and Q3 transfer functions provide superior results compared to other family members. Table 5 compares the results of five selected algorithms developed by the first approach with the BQANA developed based on the second approach. Clearly, the BQANA developed using the second approach overcomes the binary algorithms belonging to the first approach. Table 6 further investigates the proposed BQANA's effectiveness by comparing it with nine well-known optimization algorithms of the feature selection domain. The results show that the BQANA achieves superior results in terms of average fitness for most of the datasets. Regarding the BQANA's results shown in Figure 6, it has an average performance in minimizing the number of features, while turning to Figure 7, it is clear that the BQANA and the QBHHO provide the minimum number of features for most of the datasets among the competitors. The boxplots in Figure 8 illustrate the spread of the classification accuracy distribution obtained by each algorithm, in which the BQANA is predominantly the superior algorithm in terms of obtaining the highest accuracy and normal distribution.

Convergence curves plotted in Figure 9 generally suggest that the BQANA has the fastest convergence toward optimum solutions compared to comparative algorithms for most cases. Furthermore, it is noticeable that the BQANA consistently improves the solutions until the final iterations. Overall, the BQANA is fairly scalable as it can find better feature subsets for different scales of medical datasets by maintaining a balance between exploration and exploitation. Based on the results of the Friedman test reported in Table 7, the BQANA is superior to comparative algorithms in feature selection from different scales of medical data. For further statistical evaluation, Figure 10 provides the exploratory data analysis in a radar chart format. It is noticeable in the radar chart that the BQANA surrounds the center of the radar chart for most of the datasets, which shows the superiority of the BQANA over the comparative algorithms.

7. Conclusions

The advancement of information storage technologies in medical science has resulted in the generation of massive amounts of raw datasets with many irrelevant or redundant features. Selecting desirable features will reduce the computational costs and improve the algorithms' accuracy in the data-driven decision-maker software. Although many metaheuristic algorithms have been developed to select effective features, a few can maintain their performance when the number of features increases. This paper introduces an efficient binary version of the quantum-based avian navigation optimizer algorithm (QANA), called BQANA, to select effective features from various scales of medical datasets. The study consists of two approaches for mapping continuous solutions of QANA into binary. In the first approach, 20 different transfer functions from five distinct categories, S-shaped, V-shaped, U-shaped, Z-shaped, and quadratic, were applied to develop different variants of the binary QANA. According to the results, transfer functions cannot generate optimal binary solutions for every metaheuristic algorithm in the FS domain. Moreover, using transfer functions imposes additional computational costs on the optimization algorithms.

In the second approach, a simple threshold with minimum computational costs is used to assign continuous QANA solutions into binary ones to develop the BQANA. All variants were experimentally evaluated on 10 medical datasets to identify the winner variant. The experimental results reveal that the BQANA developed by the second approach generates better solutions than the other variants. Then, the results of the BQANA were compared with results obtained from nine well-known metaheuristic algorithms: BPSO, ACO, BDE, BBA, FSWOA, BALO, BDA, QBHHO, and BASO. Furthermore, the Friedman test was applied to rank the algorithms based on their performance. The experimental results and statistical analysis revealed that the BQANA developed by the second approach outperforms comparative algorithms in selecting effective feature subsets from different scales of medical datasets. In the future, the BQANA can be enhanced by improving its search strategy and using novel and more effective transfer functions. Moreover, the BQANA can be applied to solve real-world applications and other discreet problems such as nurse scheduling.

Author Contributions: Conceptualization, M.H.N.-S.; methodology, M.H.N.-S., A.F. and H.Z.; software, M.H.N.-S., A.F. and H.Z.; validation, M.H.N.-S. and H.Z.; formal analysis, M.H.N.-S., A.F. and H.Z.; investigation, M.H.N.-S., A.F. and H.Z.; resources, M.H.N.-S. and S.M.; data curation, M.H.N.-S., A.F. and H.Z.; writing, M.H.N.-S., A.F. and H.Z.; original draft preparation, M.H.N.-S., A.F. and H.Z.; writing—review and editing, M.H.N.-S., A.F. and H.Z.; visualization, M.H.N.-S., A.F. and H.Z.; supervision, M.H.N.-S. and S.M.; project administration, M.H.N.-S. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and code used in the research may be obtained from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Tables A1–A5 present the comparison results between different versions of binary QANA developed by the first approach in each transfer function family.

Table A1. The comparison results between different versions of S-BQANA.

Datasets	Metrics			S_1	S_2	S_3	S_4
Pima	Fitness	Avg		**0.2353**	0.2357	0.2374	0.2386
		Min		0.2318	0.2305	0.2318	0.2343
	Accuracy	Avg		76.8744	76.7920	76.6131	76.5053
		Max		77.2180	77.3445	77.2215	76.9617
HeartEW	Fitness	Avg		**0.1479**	0.1504	0.1510	0.1493
		Min		0.1382	0.1418	0.1424	0.1395
	Accuracy	Avg		85.5926	85.3148	85.1481	85.3333
		Max		86.6667	86.2963	85.9259	86.2963
Lymphography	Fitness	Avg		**0.1314**	0.1332	0.1361	0.1434
		Min		0.1204	0.1218	0.1136	0.1324
	Accuracy	Avg		87.3667	87.1405	86.8190	86.0595
		Max		88.5714	88.4286	89.1429	87.1905
SPECT Heart	Fitness	Avg		0.2496	0.2504	0.2462	0.2503
		Min		0.2390	0.2359	**0.2321**	0.2415
	Accuracy	Avg		75.3540	75.2158	75.6211	75.1709
		Max		76.4530	76.7236	77.1510	76.0969
PenglungEW	Fitness	Avg		0.1084	0.1042	0.1035	**0.1014**
		Min		0.0984	0.0948	0.0973	0.0830
	Accuracy	Avg		89.7589	90.0804	90.1071	90.2946
		Max		90.7143	91.0714	90.7143	92.1429
Parkinson	Fitness	Avg		0.2538	0.2489	0.2437	**0.2341**
		Min		0.2453	0.2357	0.2191	0.2160
	Accuracy	Avg		75.1071	75.4697	75.9318	76.8834
		Max		75.9281	76.8491	78.4175	78.7018
Colon	Fitness	Avg		0.1088	0.1034	0.1034	**0.1012**
		Min		0.0998	0.0978	0.0978	0.0949
	Accuracy	Avg		89.7381	90.1905	90.1310	90.3214
		Max		90.7143	90.7143	90.7143	90.9524
SRBCT	Fitness	Avg		0.0211	0.0168	0.0165	*0.0129*
		Min		0.0059	0.0060	0.0055	0.0053
	Accuracy	Avg		98.5764	98.9306	98.8889	99.2361
		Max		100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg		0.1048	0.1044	0.1021	**0.1010**
		Min		0.0820	0.0980	0.0973	0.0884
	Accuracy	Avg		90.1250	90.0804	90.2500	90.3393
		Max		92.3214	90.7143	90.7143	91.6071
Prostate Tumor	Fitness	Avg		0.1203	**0.1199**	0.1227	0.1219
		Min		0.1050	0.1106	0.1110	0.1044
	Accuracy	Avg		88.5909	88.5227	88.1727	88.2318
		Max		90.1818	89.4545	89.3636	90.0000
Overall Results	W|T|L			2|0|8	1|0|9	1|0|9	4|0|6

Table A2. The comparison results between different versions of V-BQANA.

Datasets	**Metrics**		**V_1**	**V_2**	**V_3**	**V_4**
Pima	Fitness	Avg	0.2386	0.2389	0.2390	**0.2382**
		Min	0.2330	0.2332	0.2344	0.2343
	Accuracy	Avg	76.4616	76.4863	76.4431	76.5175
		Max	77.0933	76.9532	76.8319	76.9600
HeartEW	Fitness	Avg	0.1518	**0.1513**	**0.1513**	0.1521
		Min	0.1411	0.1432	0.1424	0.1439
	Accuracy	Avg	85.0926	85.1111	85.1111	85.0370
		Max	86.2963	85.9259	85.9259	85.9259
Lymphography	Fitness	Avg	0.1422	0.1425	0.1457	**0.1412**
		Min	0.1263	0.1247	0.1382	0.1252
	Accuracy	Avg	86.1524	86.1048	85.7857	86.2667
		Max	87.8571	87.9048	86.5238	87.9048
SPECT Heart	Fitness	Avg	0.2468	**0.2457**	0.2499	0.2465
		Min	0.2225	0.2306	0.2341	0.2188
	Accuracy	Avg	75.4964	75.6019	75.1952	75.5306
		Max	77.9345	77.1652	76.7236	78.2621
PenglungEW	Fitness	Avg	0.1030	**0.0993**	0.1001	0.0998
		Min	0.0946	0.0842	0.0857	0.0825
	Accuracy	Avg	90.0536	90.4286	90.3839	90.4196
		Max	90.8929	91.9643	91.7857	92.1429
Parkinson	Fitness	Avg	**0.2115**	0.2200	0.2148	0.2160
		Min	0.1805	0.1882	0.1831	0.1908
	Accuracy	Avg	78.9557	78.1005	78.6849	78.6106
		Max	82.0000	81.4895	81.7491	81.0912
Colon	Fitness	Avg	0.1008	**0.0994**	0.1030	0.1022
		Min	0.0849	0.0852	0.0874	0.0970
	Accuracy	Avg	90.2976	90.4286	90.0714	90.1429
		Max	91.9048	91.9048	91.6667	90.4762
SRBCT	Fitness	Avg	0.0135	0.0120	0.0121	**0.0097**
		Min	0.0031	0.0038	0.0030	0.0026
	Accuracy	Avg	99.0694	99.2222	99.2083	99.4236
		Max	100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg	**0.0974**	0.1000	0.0982	0.0977
		Min	0.0837	0.0827	0.0570	0.0757
	Accuracy	Avg	90.6071	90.3661	90.5179	90.5804
		Max	91.7857	92.1429	94.4643	92.8571
Prostate Tumor	Fitness	Avg	0.1206	0.1201	**0.1152**	0.1175
		Min	0.1030	0.1074	0.0876	0.1004
	Accuracy	Avg	88.2909	88.2864	88.7591	88.5864
		Max	90.0909	89.5455	91.4545	90.3636
Overall Results	W | T | L		2 | 0 | 8	3 | 1 | 6	1 | 1 | 8	3 | 0 | 7

Table A3. The comparison results between different versions of U-BQANA.

Datasets	**Metrics**		**U_1**	**U_2**	**U_3**	**U_4**
Pima	Fitness	Avg	0.2399	0.2386	**0.2377**	0.2393
		Min	0.2345	0.2331	0.2292	0.2318
	Accuracy	Avg	76.3406	76.4590	76.5948	76.3969
		Max	76.8267	77.0813	77.4812	77.0933
HeartEW	Fitness	Avg	0.1513	0.1514	0.1514	**0.1493**
		Min	0.1432	0.1418	0.1453	0.1432
	Accuracy	Avg	85.1481	85.1296	85.0926	85.2778
		Max	85.9259	86.2963	85.5556	85.9259
Lymphography	Fitness	Avg	**0.1415**	0.1423	0.1419	0.1423
		Min	0.1267	0.1258	0.1252	0.1258
	Accuracy	Avg	86.2024	86.1524	86.1476	86.1690
		Max	87.7619	87.8571	87.8571	87.8571
SPECT Heart	Fitness	Avg	0.2474	0.2481	**0.2466**	0.2468
		Min	0.2326	0.2328	0.2374	0.2260
	Accuracy	Avg	75.4444	75.3490	75.5221	75.4623
		Max	76.7806	76.8519	76.4815	77.5356
PenglungEW	Fitness	Avg	0.0994	0.1004	**0.0991**	0.1000
		Min	0.0827	0.0880	0.0833	0.0935
	Accuracy	Avg	90.4286	90.3304	90.4375	90.3393
		Max	92.1429	91.6071	91.9643	90.8929
Parkinson	Fitness	Avg	0.2163	0.2102	0.2082	**0.2011**
		Min	0.2003	0.1903	0.1955	0.1865
	Accuracy	Avg	78.4618	79.0351	79.2160	79.9472
		Max	80.0456	80.9509	80.3982	81.4877
Colon	Fitness	Avg	0.1003	0.1004	**0.0975**	0.0995
		Min	0.0945	0.0851	0.0826	0.0916
	Accuracy	Avg	90.3571	90.3333	90.6310	90.4167
		Max	90.9524	91.9048	92.1429	91.1905
SRBCT	Fitness	Avg	0.0112	0.0110	0.0084	**0.0101**
		Min	0.0014	0.0035	0.0005	0.0021
	Accuracy	Avg	99.2361	99.3125	99.4931	99.3819
		Max	100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg	0.0986	0.1001	0.0976	**0.0960**
		Min	0.0863	0.0880	0.0827	0.0690
	Accuracy	Avg	90.4821	90.3214	90.5804	90.7143
		Max	91.7857	91.6071	91.7857	93.3929
Prostate Tumor	Fitness	Avg	0.1164	0.1169	0.1186	**0.1147**
		Min	0.0982	0.0972	0.0977	0.0984
	Accuracy	Avg	88.6045	88.5455	88.4136	88.7136
		Max	90.2727	90.4545	90.3636	90.0909
Overall Results	W|T|L		1|0|9	0|0|10	3|0|7	5|0|5

Table A4. The comparison results between different versions of Z-BQANA.

Datasets	**Metrics**		$\mathbf{Z_1}$	$\mathbf{Z_2}$	$\mathbf{Z_3}$	$\mathbf{Z_4}$
Pima	Fitness	Avg	0.2350	0.2350	0.2345	**0.2331**
		Min	0.2281	0.2279	0.2319	0.2305
	Accuracy	Avg	76.8837	76.9140	76.9554	77.0729
		Max	77.4675	77.4846	77.2283	77.4778
HeartEW	Fitness	Avg	0.1496	0.1472	**0.1435**	0.1465
		Min	0.1395	0.1382	0.1387	0.1380
	Accuracy	Avg	85.4074	85.6852	85.9630	85.7222
		Max	86.2963	86.6667	86.2963	86.6667
Lymphography	Fitness	Avg	**0.1273**	0.1366	0.1313	0.1328
		Min	0.1049	0.1138	0.1132	0.1120
	Accuracy	Avg	87.7024	86.8548	87.4071	87.1881
		Max	89.9048	89.2381	89.2381	89.1905
SPECT Heart	Fitness	Avg	0.2450	0.2491	**0.2411**	**0.2411**
		Min	0.2288	0.2285	0.2167	0.2161
	Accuracy	Avg	75.7543	75.3796	76.1197	76.1830
		Max	77.2080	77.2365	78.6182	78.6325
PenglungEW	Fitness	Avg	0.1012	0.1002	0.0992	**0.0985**
		Min	0.0805	0.0804	0.0811	0.0848
	Accuracy	Avg	90.3929	90.5089	90.6339	90.6875
		Max	92.3214	92.3214	92.3214	91.9643
Parkinson	Fitness	Avg	**0.2308**	0.2369	0.2355	0.2363
		Min	0.1949	0.2082	0.1931	0.1935
	Accuracy	Avg	77.2476	76.6975	76.7999	76.7173
		Max	80.8175	79.5018	80.9579	80.9421
Colon	Fitness	Avg	0.1007	**0.0995**	0.1019	0.1006
		Min	0.0874	0.0781	0.0827	0.0855
	Accuracy	Avg	90.4405	90.4762	90.3452	90.4524
		Max	91.6667	92.6190	92.1429	92.1429
SRBCT	Fitness	Avg	0.0116	**0.0115**	0.0132	0.0152
		Min	0.0045	0.0044	0.0041	0.0038
	Accuracy	Avg	99.3750	99.3750	99.2431	99.0347
		Max	100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg	**0.0969**	0.1003	0.0984	0.0983
		Min	0.0825	0.0830	0.0716	0.0836
	Accuracy	Avg	90.8304	90.4732	90.6964	90.6875
		Max	92.1429	92.1429	93.2143	91.9643
Prostate Tumor	Fitness	Avg	0.1195	0.1178	**0.1146**	0.1178
		Min	0.1014	0.1004	0.1004	0.1005
	Accuracy	Avg	88.5409	88.7182	89.0364	88.6864
		Max	90.2727	90.3636	90.3636	90.3636
Overall Results	W|T|L		3|0|7	2|0|8	2|1|8	2|1|7

Table A5. The comparison results between different versions of Q-BQANA.

Datasets	**Metrics**		**Q1**	**Q2**	**Q3**	**Q4**
Pima	Fitness	Avg	0.2392	0.2390	**0.2380**	0.2385
		Min	0.2331	0.2318	0.2319	0.2344
	Accuracy	Avg	76.4142	76.4458	76.4997	76.4923
		Max	76.9549	77.0933	77.0762	76.8267
HeartEW	Fitness	Avg	0.1522	0.1528	**0.1509**	0.1540
		Min	0.1468	0.1395	0.1416	0.1468
	Accuracy	Avg	85.0556	84.9815	85.0741	84.8333
		Max	85.5556	86.2963	85.9259	85.5556
Lymphography	Fitness	Avg	0.1438	**0.1416**	0.1432	0.1428
		Min	0.1316	0.1252	0.1303	0.1202
	Accuracy	Avg	85.9619	86.1524	85.9952	86.1143
		Max	87.1429	87.8571	87.2857	88.4762
SPECT Heart	Fitness	Avg	0.2474	**0.2441**	0.2453	0.2464
		Min	0.2287	0.2112	0.2326	0.2347
	Accuracy	Avg	75.4345	75.6916	75.5648	75.5221
		Max	77.1795	79.0313	76.8234	76.7949
PenglungEW	Fitness	Avg	0.1015	0.0982	**0.0977**	0.0991
		Min	0.0958	0.0684	0.0828	0.0829
	Accuracy	Avg	90.2054	90.4821	90.5357	90.4821
		Max	90.7143	93.5714	91.9643	92.1429
Parkinson	Fitness	Avg	0.2256	0.2109	0.2049	**0.2188**
		Min	0.1948	0.1888	0.1838	0.1998
	Accuracy	Avg	77.6659	78.9870	79.6068	78.3081
		Max	80.8175	81.2193	81.7456	80.0281
Colon	Fitness	Avg	0.1008	0.0994	**0.0947**	0.1017
		Min	0.0829	0.0783	0.0809	0.0943
	Accuracy	Avg	90.2738	90.3571	90.7976	90.2262
		Max	92.1429	92.3810	92.1429	90.9524
SRBCT	Fitness	Avg	**0.0130**	0.0060	0.0047	0.0089
		Min	0.0026	0.0021	0.0018	0.0024
	Accuracy	Avg	99.1389	99.7153	99.8333	99.5417
		Max	100.00	100.00	100.00	100.00
Leukemia	Fitness	Avg	0.1009	0.0969	**0.0915**	0.0970
		Min	0.0875	0.0844	0.0702	0.0844
	Accuracy	Avg	90.2946	90.6071	91.0714	90.6696
		Max	91.6071	91.7857	93.2143	91.7857
Prostate Tumor	Fitness	Avg	0.1176	0.1126	**0.1113**	0.1201
		Min	0.1032	0.0999	0.0882	0.1040
	Accuracy	Avg	88.5773	88.9864	89.0818	88.3364
		Max	90.0909	90.1818	91.2727	90.0000
Overall Results	W|T|L		1|0|9	2|0|8	6|0|4	1|0|9

References

1. Remeseiro, B.; Bolon-Canedo, V. A review of feature selection methods in medical applications. *Comput. Biol. Med.* **2019**, *112*, 103375. [CrossRef] [PubMed]
2. Esfandiari, N.; Babavalian, M.R.; Moghadam, A.-M.E.; Tabar, V.K. Knowledge discovery in medicine: Current issue and future trend. *Expert Syst. Appl.* **2014**, *41*, 4434–4463. [CrossRef]
3. Hashemi, F.S.G.; Ismail, M.R.; Yusop, M.R.; Hashemi, M.S.G.; Nadimi-Shahraki, M.H.; Rastegari, H.; Miah, G.; Aslani, F. Intelligent mining of large-scale bio-data: Bioinformatics applications. *Biotechnol. Biotechnol. Equip.* **2018**, *32*, 10–29. [CrossRef]
4. Kalantari, A.; Kamsin, A.; Shamshirband, S.; Gani, A.; Alinejad-Rokny, H.; Chronopoulos, A.T. Computational intelligence approaches for classification of medical data: State-of-the-art, future challenges and research directions. *Neurocomputing* **2018**, *276*, 2–22. [CrossRef]
5. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. From data mining to knowledge discovery in databases. *AI Mag.* **1996**, *17*, 37.
6. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
7. Qiu, C. A novel multi-swarm particle swarm optimization for feature selection. *Genet. Program. Evolvable Mach.* **2019**, *20*, 503–529. [CrossRef]
8. Ibrahim, R.A.; Ewees, A.A.; Oliva, D.; Abd Elaziz, M.; Lu, S. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3155–3169. [CrossRef]
9. Gharehchopogh, F.S.; Maleki, I.; Dizaji, Z.A. Chaotic vortex search algorithm: Metaheuristic algorithm for feature selection. *Evol. Intell.* **2021**, *15*, 1777–1808. [CrossRef]
10. Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 454.
11. Zamani, H.; Nadimi-Shahraki, M.-H. Feature selection based on whale optimization algorithm for diseases diagnosis. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 1243.
12. Inbarani, H.H.; Azar, A.T.; Jothi, G. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Comput. Methods Programs Biomed.* **2014**, *113*, 175–185. [CrossRef] [PubMed]
13. Polat, H.; Danaei Mehr, H.; Cetin, A. Diagnosis of chronic kidney disease based on support vector machine by feature selection methods. *J. Med. Syst.* **2017**, *41*, 55. [CrossRef] [PubMed]
14. Chatterjee, S.; Biswas, S.; Majee, A.; Sen, S.; Oliva, D.; Sarkar, R. Breast cancer detection from thermal images using a Grunwald-Letnikov-aided Dragonfly algorithm-based deep feature selection method. *Comput. Biol. Med.* **2022**, *141*, 105027. [CrossRef]
15. Ayar, M.; Isazadeh, A.; Gharehchopogh, F.S.; Seyedi, M. Chaotic-based divide-and-conquer feature selection method and its application in cardiac arrhythmia classification. *J. Supercomput.* **2022**, *78*, 5856–5882. [CrossRef]
16. Bharti, K.K.; Singh, P.K. Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. *Expert Syst. Appl.* **2015**, *42*, 3105–3114. [CrossRef]
17. Abualigah, L.M.Q. *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering*; Springer: Berlin/Heidelberg, Germany, 2019.
18. Naseri, T.S.; Gharehchopogh, F.S. A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems. *J. Netw. Syst. Manag.* **2022**, *30*, 40. [CrossRef]
19. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. *J. Netw. Comput. Appl.* **2011**, *34*, 1184–1199. [CrossRef]
20. Khater, B.S.; Abdul Wahab, A.W.; Idris, M.Y.I.; Hussain, M.A.; Ibrahim, A.A.; Amin, M.A.; Shehadeh, H.A. Classifier Performance Evaluation for Lightweight IDS Using Fog Computing in IoT Security. *Electronics* **2021**, *10*, 1633. [CrossRef]
21. Samadi Bonab, M.; Ghaffari, A.; Gharehchopogh, F.S.; Alemi, P. A wrapper-based feature selection for improving performance of intrusion detection systems. *Int. J. Commun. Syst.* **2020**, *33*, e4434. [CrossRef]
22. Mohammadzadeh, H.; Gharehchopogh, F.S. A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study Email spam detection. *Comput. Intell.* **2021**, *37*, 176–209. [CrossRef]
23. Lee, S.M.; Kim, D.S.; Kim, J.H.; Park, J.S. Spam detection using feature selection and parameters optimization. In Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, Krakow, Poland, 15–18 February 2010; pp. 883–888.
24. Zhang, Y.; Wang, S.; Phillips, P.; Ji, G. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. *Knowl.-Based Syst.* **2014**, *64*, 22–31. [CrossRef]
25. Mohammadzadeh, H.; Gharehchopogh, F.S. Feature selection with binary symbiotic organisms search algorithm for email spam detection. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 469–515. [CrossRef]
26. Abusamra, H. A comparative study of feature selection and classification methods for gene expression data of glioma. *Procedia Comput. Sci.* **2013**, *23*, 5–14. [CrossRef]
27. Hauskrecht, M.; Pelikan, R.; Valko, M.; Lyons-Weiler, J. Feature selection and dimensionality reduction in genomics and proteomics. In *Fundamentals of Data mining in Genomics and Proteomics*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 149–172.
28. Tadist, K.; Najah, S.; Nikolov, N.S.; Mrabti, F.; Zahi, A. Feature selection methods and genomic big data: A systematic review. *J. Big Data* **2019**, *6*, 79. [CrossRef]
29. Xing, E.P.; Jordan, M.I.; Karp, R.M. *Feature Selection for High-Dimensional Genomic Microarray Data*; ICML: Williamstown, MA, USA, 2001; pp. 601–608.

30. Liu, H.; Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502.
31. Sharda, S.; Srivastava, M.; Gusain, H.S.; Sharma, N.K.; Bhatia, K.S.; Bajaj, M.; Kaur, H.; Zawbaa, H.M.; Kamel, S. A hybrid machine learning technique for feature optimization in object-based classification of debris-covered glaciers. *Ain Shams Eng. J.* **2022**, *13*, 101809. [CrossRef]
32. Chen, C.W.; Tsai, Y.H.; Chang, F.R.; Lin, W.C. Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert Syst.* **2020**, *37*, e12553. [CrossRef]
33. Jović, A.; Brkić, K.; Bogunović, N. A review of feature selection methods with applications. In Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205.
34. Hall, M.A.; Smith, L.A. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In Proceedings of the FLAIRS Conference, Orlando, FL, USA, 1–5 May 1999; pp. 235–239.
35. Agrawal, P.; Ganesh, T.; Oliva, D.; Mohamed, A.W. S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection. *Appl. Intell.* **2022**, *52*, 81–112. [CrossRef]
36. Rodrigues, D.; Pereira, L.A.; Papa, J.P.; Weber, S.A. A binary krill herd approach for feature selection. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 1407–1412.
37. Nadimi-Shahraki, M.H.; Fatahi, A.; Zamani, H.; Mirjalili, S.; Abualigah, L.; Abd Elaziz, M. Migration-based moth-flame optimization algorithm. *Processes* **2021**, *9*, 2276. [CrossRef]
38. Tran, B.; Xue, B.; Zhang, M. Adaptive multi-subswarm optimisation for feature selection on high-dimensional classification. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 481–489.
39. Abd Elaziz, M.; Oliva, D. Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm. *Energy Convers. Manag.* **2018**, *171*, 1843–1859. [CrossRef]
40. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. CCSA: Conscious neighborhood-based crow search algorithm for solving global optimization problems. *Appl. Soft Comput.* **2019**, *85*, 105583. [CrossRef]
41. Trojovský, P.; Dehghani, M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors* **2022**, *22*, 855. [CrossRef]
42. Kharrich, M.; Kamel, S.; Hassan, M.H.; ElSayed, S.K.; Taha, I.B.M. An Improved Heap-Based Optimizer for Optimal Design of a Hybrid Microgrid Considering Reliability and Availability Constraints. *Sustainability* **2021**, *13*, 10419. [CrossRef]
43. Gharehchopogh, F.S.; Farnad, B.; Alizadeh, A. A modified farmland fertility algorithm for solving constrained engineering problems. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6310. [CrossRef]
44. Amini, E.; Mehdipour, H.; Faraggiana, E.; Golbaz, D.; Mozaffari, S.; Bracco, G.; Neshat, M. Optimization Study of Hydraulic Power Take-off System for an Ocean Wave Energy Converter. *arXiv* **2021**, arXiv:2112.09803.
45. Ceylan, O.; Neshat, M.; Mirjalili, S. Cascaded H-bridge multilevel inverters optimization using adaptive grey wolf optimizer with local search. *Electr. Eng.* **2021**, 1–15. [CrossRef]
46. Nadimi-Shahraki, M.H.; Fatahi, A.; Zamani, H.; Mirjalili, S.; Abualigah, L. An Improved Moth-Flame Optimization Algorithm with Adaptation Mechanism to Solve Numerical and Mechanical Engineering Problems. *Entropy* **2021**, *23*, 1637. [CrossRef]
47. Abd Elaziz, M.; Elsheikh, A.H.; Oliva, D.; Abualigah, L.; Lu, S.; Ewees, A.A. Advanced metaheuristic techniques for mechanical design problems. *Arch. Comput. Methods Eng.* **2022**, *29*, 695–716. [CrossRef]
48. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Comput. Methods Appl. Mech. Eng.* **2022**, *392*, 114616. [CrossRef]
49. Chakraborty, S.; Sharma, S.; Saha, A.K.; Saha, A. A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. *Artif. Intell. Rev.* **2022**, *55*, 4605–4716. [CrossRef]
50. El-Shorbagy, M. Chaotic Fruit Fly Algorithm for Solving Engineering Design Problems. *Complexity* **2022**, *2022*, 6627409. [CrossRef]
51. Sa'ad, S.; Muhammed, A.; Abdullahi, M.; Abdullah, A.; Hakim Ayob, F. An Enhanced Discrete Symbiotic Organism Search Algorithm for Optimal Task Scheduling in the Cloud. *Algorithms* **2021**, *14*, 200. [CrossRef]
52. Wang, Y.; Yang, Z.; Guo, Y.; Zhou, B.; Zhu, X. A Novel Binary Competitive Swarm Optimizer for Power System Unit Commitment. *Appl. Sci.* **2019**, *9*, 1776. [CrossRef]
53. Izakian, H.; Abraham, A.; Snášel, V. Metaheuristic Based Scheduling Meta-Tasks in Distributed Heterogeneous Computing Systems. *Sensors* **2009**, *9*, 5339–5350. [CrossRef] [PubMed]
54. Eslami, M.; Neshat, M.; Khalid, S.A. A Novel Hybrid Sine Cosine Algorithm and Pattern Search for Optimal Coordination of Power System Damping Controllers. *Sustainability* **2022**, *14*, 541. [CrossRef]
55. Li, W.; Luo, H.; Wang, L.; Jiang, Q.; Xu, Q. Enhanced Brain Storm Optimization Algorithm Based on Modified Nelder–Mead and Elite Learning Mechanism. *Mathematics* **2022**, *10*, 1303. [CrossRef]
56. Heydari, A.; Majidi Nezhad, M.; Neshat, M.; Garcia, D.A.; Keynia, F.; De Santoli, L.; Tjernberg, L.B. A combined fuzzy GMDH neural network and grey wolf optimization application for wind turbine power production forecasting considering SCADA data. *Energies* **2021**, *14*, 3459. [CrossRef]
57. Chou, J.-S.; Chong, W.K.; Bui, D.-K. Nature-inspired metaheuristic regression system: Programming and implementation for civil engineering applications. *J. Comput. Civ. Eng.* **2016**, *30*, 04016007. [CrossRef]

58. Neshat, M. The Application of Nature-Inspired Metaheuristic Methods for Optimising Renewable Energy Problems and the Design of Water Distribution Networks. Ph.D. Thesis, University of Adelaide, Adelaide, Australia, 2020.
59. Mohamed, A.A.; Kamel, S.; Hassan, M.H.; Mosaad, M.I.; Aljohani, M. Optimal Power Flow Analysis Based on Hybrid Gradient-Based Optimizer with Moth–Flame Optimization Algorithm Considering Optimal Placement and Sizing of FACTS/Wind Power. *Mathematics* **2022**, *10*, 361. [CrossRef]
60. Farhat, M.; Kamel, S.; Atallah, A.M.; Khan, B. Optimal power flow solution based on jellyfish search optimization considering uncertainty of renewable energy sources. *IEEE Access* **2021**, *9*, 100911–100933. [CrossRef]
61. Abido, M.A. Optimal power flow using particle swarm optimization. *Int. J. Electr. Power Energy Syst.* **2002**, *24*, 563–571. [CrossRef]
62. Bakirtzis, A.G.; Biskas, P.N.; Zoumas, C.E.; Petridis, V. Optimal power flow by enhanced genetic algorithm. *IEEE Trans. Power Syst.* **2002**, *17*, 229–236. [CrossRef]
63. Nadimi-Shahraki, M.H.; Fatahi, A.; Zamani, H.; Mirjalili, S.; Oliva, D. Hybridizing of Whale and Moth-Flame Optimization Algorithms to Solve Diverse Scales of Optimal Power Flow Problem. *Electronics* **2022**, *11*, 831. [CrossRef]
64. Saha, A.; Bhattacharya, A.; Das, P.; Chakraborty, A.K. Crow search algorithm for solving optimal power flow problem. In Proceedings of the 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 22–24 February 2017; pp. 1–8.
65. Radpour, V.; Gharehchopogh, F.S. A Novel Hybrid Binary Farmland Fertility Algorithm with Naïve Bayes for Diagnosis of Heart Disease. *Sak. Univ. J. Comput. Inf. Sci.* **2022**, *5*, 90–103.
66. Abualigah, L.; Diabat, A. Chaotic binary group search optimizer for feature selection. *Expert Syst. Appl.* **2022**, *192*, 116368. [CrossRef]
67. Ibrahim, R.A.; Abualigah, L.; Ewees, A.A.; Al-Qaness, M.A.; Yousri, D.; Alshathri, S.; Abd Elaziz, M. An electric fish-based arithmetic optimization algorithm for feature selection. *Entropy* **2021**, *23*, 1189. [CrossRef]
68. Mostafa, R.R.; Ewees, A.A.; Ghoniem, R.M.; Abualigah, L.; Hashim, F.A. Boosting chameleon swarm algorithm with consumption AEO operator for global optimization and feature selection. *Knowl. Based Syst.* **2022**, *246*, 108743. [CrossRef]
69. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [CrossRef]
70. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; Ala'M, A.-Z.; Mirjalili, S. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* **2019**, *117*, 267–286. [CrossRef]
71. Sindhu, R.; Ngadiran, R.; Yacob, Y.M.; Zahri, N.A.H.; Hariharan, M. Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. *Neural Comput. Appl.* **2017**, *28*, 2947–2958. [CrossRef]
72. Dhiman, G.; Oliva, D.; Kaur, A.; Singh, K.K.; Vimal, S.; Sharma, A.; Cengiz, K. BEPO: A novel binary emperor penguin optimizer for automatic feature selection. *Knowl.-Based Syst.* **2021**, *211*, 106560. [CrossRef]
73. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. QANA: Quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104314. [CrossRef]
74. Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
75. Nadimi-Shahraki, M.H.; Zamani, H. DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization. *Expert Syst. Appl.* **2022**, *198*, 116895. [CrossRef]
76. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 4104–4108.
77. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSA: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [CrossRef]
78. Mirjalili, S.; Zhang, H.; Mirjalili, S.; Chalup, S.; Noman, N. A novel U-shaped transfer function for binary particle swarm optimisation. In *Soft Computing for Problem Solving 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 241–259.
79. Guo, S.-S.; Wang, J.-S.; Guo, M.-W. Z-shaped transfer functions for binary particle swarm optimization algorithm. *Comput. Intell. Neurosci.* **2020**, *2020*, 6502807. [CrossRef] [PubMed]
80. Too, J.; Abdullah, A.R.; Mohd Saad, N. A new quadratic binary harris hawk optimization for feature selection. *Electronics* **2019**, *8*, 1130. [CrossRef]
81. Jordehi, A.R. Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes. *Appl. Soft Comput.* **2019**, *78*, 465–480. [CrossRef]
82. Liao, C.-J.; Tseng, C.-T.; Luarn, P. A discrete version of particle swarm optimization for flowshop scheduling problems. *Comput. Oper. Res.* **2007**, *34*, 3099–3111. [CrossRef]
83. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1470–1477.
84. Gong, T.; Tuson, A.L. Differential evolution for binary encoding. In *Soft Computing in Industrial Applications*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 251–262.
85. Nakamura, R.Y.; Pereira, L.A.; Costa, K.A.; Rodrigues, D.; Papa, J.P.; Yang, X.-S. BBA: A binary bat algorithm for feature selection. In Proceedings of the 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, Ouro Preto, Brazil, 22–25 August 2012; pp. 291–297.

86. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [CrossRef]
87. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
88. Too, J.; Rahim Abdullah, A. Binary atom search optimisation approaches for feature selection. *Connect. Sci.* **2020**, *32*, 406–430. [CrossRef]
89. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
90. Aghdam, M.H.; Ghasem-Aghaee, N.; Basiri, M.E. Text feature selection using ant colony optimization. *Expert Syst. Appl.* **2009**, *36*, 6843–6853. [CrossRef]
91. Chen, B.; Chen, L.; Chen, Y. Efficient ant colony optimization for image feature selection. *Signal Process.* **2013**, *93*, 1566–1576. [CrossRef]
92. Aghdam, M.H.; Kabiri, P. Feature selection for intrusion detection system using ant colony optimization. *Int. J. Netw. Secur.* **2016**, *18*, 420–432.
93. Renuka, D.K.; Visalakshi, P.; Sankar, T. Improving Email spam classification using ant colony optimization algorithm. *Int. J. Comput. Appl.* **2015**, *22*, 22–26.
94. Taghian, S.; Nadimi-Shahraki, M.H.; Zamani, H. Comparative analysis of transfer function-based binary Metaheuristic algorithms for feature selection. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–6.
95. Papa, J.P.; Rosa, G.H.; Papa, L.P. A binary-constrained Geometric Semantic Genetic Programming for feature selection purposes. *Pattern Recog. Lett.* **2017**, *100*, 59–66. [CrossRef]
96. Marandi, A.; Afshinmanesh, F.; Shahabadi, M.; Bahrami, F. Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna. In Proceedings of the 2006 IEEE international conference on evolutionary computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 3212–3218.
97. Aslan, M.; Gunduz, M.; Kiran, M.S. JayaX: Jaya algorithm with xor operator for binary optimization. *Appl. Soft Comput.* **2019**, *82*, 105576. [CrossRef]
98. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [CrossRef]
99. Lin, J.C.-W.; Yang, L.; Fournier-Viger, P.; Hong, T.-P.; Voznak, M. A binary PSO approach to mine high-utility itemsets. *Soft Comput.* **2017**, *21*, 5103–5121. [CrossRef]
100. Guha, R.; Ghosh, K.K.; Bera, S.K.; Sarkar, R.; Mirjalili, S. Discrete Equilibrium Optimizer Combined with Simulated Annealing for Feature Selection, 4 January 2022, PREPRINT (Version 2). Research Square. Available online: https://assets.researchsquare.com/files/rs-28683/v2_covered.pdf?c=1641365319 (accessed on 30 June 2022).
101. Nadimi-Shahraki, M.H.; Zamani, H.; Mirjalili, S. Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study. *Comput. Biol. Med.* **2022**, *148*, 105858. [CrossRef] [PubMed]
102. Hafez, A.I.; Zawbaa, H.M.; Emary, E.; Hassanien, A.E. Sine cosine optimization algorithm for feature selection. In Proceedings of the 2016 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Sinaia, Romania, 2–5 August 2016; pp. 1–5.
103. Tran, B.; Xue, B.; Zhang, M. Variable-length particle swarm optimization for feature selection on high-dimensional classification. *IEEE Trans. Evol. Comput.* **2018**, *23*, 473–487. [CrossRef]
104. Tanabe, R.; Fukunaga, A. Success-history based parameter adaptation for differential evolution. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 71–78.
105. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]
106. Seger, C. An Investigation of Categorical Variable Encoding Techniques in Machine Learning: Binary versus One-Hot and Feature Hashing. Bachelor's Thesis, KTH School of Electrical Engineering and Computer Science, Stockholm, Sweden, 2018.
107. Kaur, H.; Kumari, V. Predictive modelling and analytics for diabetes using a machine learning approach. *Appl. Comput. Inform.* **2020**, *18*, 90–100. [CrossRef]
108. Blake, C. UCI Repository of Machine Learning Databases. 1998. Available online: http://www.ics.uci.edu/~{}mlearn/MLRepository.html (accessed on 30 June 2022).
109. Alon, U.; Barkai, N.; Notterman, D.A.; Gish, K.; Ybarra, S.; Mack, D.; Levine, A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA* **1999**, *96*, 6745–6750. [CrossRef] [PubMed]
110. Khan, J.; Wei, J.S.; Ringner, M.; Saal, L.H.; Ladanyi, M.; Westermann, F.; Berthold, F.; Schwab, M.; Antonescu, C.R.; Peterson, C. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.* **2001**, *7*, 673–679. [CrossRef]

111. Golub, T.R.; Slonim, D.K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J.P.; Coller, H.; Loh, M.L.; Downing, J.R.; Caligiuri, M.A. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **1999**, *286*, 531–537. [CrossRef]
112. Kar, S.; Sharma, K.D.; Maitra, M. Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique. *Expert Syst. Appl.* **2015**, *42*, 612–627. [CrossRef]