applied sciences

MDPI

Article

Binary Starling Murmuration Optimizer Algorithm to Select Effective Features from Medical Data

Mohammad H. Nadimi-Shahraki [1,2,3,*], Zahra Asghari Varzaneh [4], Hoda Zamani [1,2] and Seyedali Mirjalili [3,5,*]

[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, QLD 4006, Australia
[4] Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman 7616914111, Iran
[5] Yonsei Frontier Lab, Yonsei University, Seoul 03722, Republic of Korea
* Correspondence: nadimi@iaun.ac.ir (M.H.N.-S.); ali.mirjalili@torrens.edu.au (S.M.)

Abstract: Feature selection is an NP-hard problem to remove irrelevant and redundant features with no predictive information to increase the performance of machine learning algorithms. Many wrapper-based methods using metaheuristic algorithms have been proposed to select effective features. However, they achieve differently on medical data, and most of them cannot find those effective features that may fulfill the required accuracy in diagnosing important diseases such as Diabetes, Heart problems, Hepatitis, and Coronavirus, which are targeted datasets in this study. To tackle this drawback, an algorithm is needed that can strike a balance between local and global search strategies in selecting effective features from medical datasets. In this paper, a new binary optimizer algorithm named BSMO is proposed. It is based on the newly proposed starling murmuration optimizer (SMO) that has a high ability to solve different complex and engineering problems, and it is expected that BSMO can also effectively find an optimal subset of features. Two distinct approaches are utilized by the BSMO algorithm when searching medical datasets to find effective features. Each dimension in a continuous solution generated by SMO is simply mapped to 0 or 1 using a variable threshold in the second approach, whereas in the first, binary versions of BSMO are developed using several S-shaped and V-shaped transfer functions. The performance of the proposed BSMO was evaluated using four targeted medical datasets, and results were compared with well-known binary metaheuristic algorithms in terms of different metrics, including fitness, accuracy, sensitivity, specificity, precision, and error. Finally, the superiority of the proposed BSMO algorithm was statistically analyzed using Friedman non-parametric test. The statistical and experimental tests proved that the proposed BSMO attains better performance in comparison to the competitive algorithms such as ACO, BBA, bGWO, and BWOA for selecting effective features from the medical datasets targeted in this study.

Keywords: disease diagnosis; medical data; feature selection; binary metaheuristic algorithms; starling murmuration optimizer (SMO); transfer function

Citation: Nadimi-Shahraki, M.H.; Asghari Varzaneh, Z.; Zamani, H.; Mirjalili, S. Binary Starling Murmuration Optimizer Algorithm to Select Effective Features from Medical Data. *Appl. Sci.* **2023**, *13*, 564. https://doi.org/10.3390/app13010564

Academic Editor: Cosimo Nardi

Received: 20 November 2022
Revised: 20 December 2022
Accepted: 29 December 2022
Published: 31 December 2022

1. Introduction

With recent advancements in medical information technology, a huge volume of raw medical data is rapidly generated from different medical resources such as medical examinations, radiology, laboratory tests, mobile health applications, and wearable healthcare technologies [1–3]. Extracting informative knowledge from these medical data using artificial intelligence and machine learning algorithms can help in faster treatment and significantly reduce patient mortality rates [4,5]. Application of these algorithms in some diseases such as Diabetes, Heart problems, Hepatitis, and Coronavirus is more common

than others due to their high epidemic and mortality rates, expensive tests, and the requirement of special experience [6–8]. One of the main challenges in such disease datasets is the existence of redundant and irrelevant features [9], which can decrease the effectiveness of disease diagnosis systems. In medical data mining and machine learning [10,11], one of the most crucial preprocessing steps is feature selection, which eliminates redundant and irrelevant features to uncover effective ones. Since there are 2^N distinct feature subsets in a dataset with N features, the feature selection problem is NP-hard [12,13]. Therefore, evaluating all feature subsets to find effective features is very costly, and if each feature is added to the dataset, then the complexity will be doubled [13,14].

Filter-based, wrapper-based, and embedded methods are the three main categories of feature selection techniques [15,16]. The classification algorithm is not involved in filter-based methods, which typically operate based on feature ranking. Wrapper-based methods use a classifier algorithm to evaluate individual candidate subsets of features as opposed to filter-based methods [17,18]. Embedded methods combine the qualities of filter and wrapper methods, and the feature selection algorithm is integrated as part of the learning algorithm [16]. Many wrapper feature selection methods based on metaheuristic algorithms have been proposed [15,16] that can effectively solve feature selection problems as an NP-hard problem in a reasonable response time [19,20]. The main goal of using metaheuristic algorithms is to search the feature space and find near-optimal solutions effectively. Metaheuristic algorithms are recognized as robust problem solvers to solve a variety of problems with different types, such as continuous [21], discrete [22–24], and constraint [25,26]. Particle swarm optimization (PSO) [27], ant colony optimization (ACO) [28], differential evolution (DE) [29], cuckoo optimization algorithm (COA) [30], krill herd (KH) [31], social spider algorithm (SSA) [32], crow search algorithm (CSA) [33], grasshopper optimization algorithm (GOA) [34], quantum-based avian navigation optimizer algorithm (QANA) [35] and African vultures optimization algorithm (AVOA) [36] are some of the successful metaheuristic algorithms that are promisingly developed to solve feature selection problems.

Many metaheuristic-based methods have been proposed to select features from medical data [37–39]. However, a few of them can select effective features that may provide acceptable accuracy in diagnosing all the targeted diseases in this study, including Diabetes, Heart problems, Hepatitis, and Coronavirus [40]. The main reason for this drawback is generating and storing many irrelevant and redundant features in the medical processes, which reduces the efficiency of classification algorithms used in disease diagnosis systems. Therefore, a metaheuristic algorithm is needed to select useful and effective features from medical datasets by striking a proper balance between local and global search strategies. Responding to this need, particularly for the datasets targeted in the scope of this study, is our motivation to introduce binary versions of the newly proposed starling murmuration optimizer (SMO) algorithm [41], which can balance between its search strategies efficiently. The SMO algorithm uses a dynamic multi-flock construction and three search strategies: separating, diving, and whirling. Starlings in large flocks turn, dive, and whirl across the sky in SMO. The separating search strategy enriches population diversity by employing the quantum harmonic oscillator. With the help of a quantum random dive operator, the diving search strategy enhances the exploration. In contrast, the whirling search strategy significantly uses cohesion force in the vicinity of promising regions. The SMO algorithm has shown a high ability to solve different complex and engineering problems, but it was not yet developed for solving feature selection problems. The binary version of SMO or BSMO is expected to effectively solve the feature selection problem.

The BSMO algorithm generates candidate subsets of features using two different approaches. The first approach develops binary versions of BSMO using several S-shaped and V-shaped transfer functions. In contrast, in the second approach, BSMO maps each dimension in a continuous solution generated by SMO to 0 or 1 using a variable threshold method. The scope of this study is limited to selecting effective features from four targeted datasets consisting of Diabetes, Heart, Hepatitis, and Coronavirus. The performance of the

BSMO's variants is assessed on targeted datasets in terms of fitness, accuracy, sensitivity, specificity, precision, and error. The results are contrasted with competing binary algorithms like the ant colony optimization (ACO) [28], binary bat algorithm (BBA) [42], binary grey wolf optimization (bGWO) [43], and binary whale optimization algorithm (BWOA) [39]. The main contributions of this study can be summarized as follows.

- Developing the BSMO algorithm as a binary version of the SMO algorithm.
- Transferring the continuous solutions to binary ones effectively using two different approaches, including S-shaped and V-shaped transfer functions and value threshold method.
- Evaluating BSMO on medical datasets targeted in this study and comparing its performance with other popular feature selection algorithms.
- Finding satisfactory results in selecting effective features from the targeted medical datasets.

The rest of this paper is organized as follows. The related works are reviewed in Section 2. A description of the standard SMO algorithm is presented in Section 3. The details of the proposed BSMO algorithm are presented in Section 4. Section 5 includes the experimental evaluation and the comparison between the proposed BSMO and contender algorithms. Section 6 concludes this study and its finding, and suggests some future works.

2. Related Works

Real-world optimization problems have different properties and involve various intricacies, creating critical challenges for optimization algorithms in solving them. Generally, optimization problems in mechanical and engineering applications are mostly faced with multiple properties, such as linear and non-linear constraints in decision variables, non-differentiable objectives, and constraint functions. Therefore, many constraint-handling methods, such as penalty functions, static, dynamic, annealing, adaptive, co-evolutionary, and the death penalty, are developed to cope with such challenges [44]. The other optimization problems, especially in feature selection applications, mostly involve different intricacies such as discrete search spaces, existing irrelevant and redundant features, and high dimensionality feature space. Feature selection is a common way in preprocessing phase to cope with such intricacies by selecting only a small subset of relevant features from the original dataset [45,46]. Feature selection reduces the feature space's dimensionality, speeds up the learning process, simplifies the learned model, and boosts classifier performance by eliminating redundant and irrelevant features [47–49].

The topic of feature selection is presented as a binary optimization problem with the conflicting objectives of reducing the number of features and enhancing classification accuracy. Each solution is presented by a D-dimensional binary vector that only has the two values 0 and 1, where 0 signifies that the corresponding feature is not selected, and 1 indicates that it is selected. The number of dimensions in this binary vector corresponds to the number of features in the initial feature dataset. In many machine learning and data mining tasks, including intrusion detection [50–53], spam detection [54,55], financial problem prediction [56], and classification [57–59]. Particularly, finding an optimal subset of features from medical datasets is a challenging problem that many researchers have recently considered. Metaheuristic algorithms are recognized as prominent problem-solver to solve optimization problems especially feature selection. Based on the source of their inspiration, metaheuristic algorithms may be divided into eight groups: physical-based, biology-based, swarm-based, social-based, mu-sic-based, sport-based, chemistry-based, and math-based [60–62]. Since most metaheuristic algorithms are proposed for continuous problems, many binarization methods such as logical operators, variable threshold methods and transfer functions, are developed to map the continuous feature space to the binary one. In the literature, the most famous transfer functions are S-shaped [63], V-shaped [64–66], U-shaped [67,68], X-shaped transfer function [69], and Z-shaped [70]. This section presents an overview of the most recent related works on metaheuristics for the wrapper feature selection problem in medical data classification.

Nadimi-Shahraki et al. [40] proposed an improved whale optimization algorithm called BE-WOA. In BE-WOA, a pooling mechanism and three effective search strategies, migration, preferential selection, and surrounded prey, are used to improve the WOA to select effective features from medical datasets. BE-WOA also applied to predict Coronavirus 2019 disease or COVID-19. The obtained results prove the efficiency of the BE-WOA algorithm. The gene selection technique is used for high-dimensional datasets where the number of samples is small, and the number of features is large. Finding the best feature subset in a dataset is the process of gene selection [71]. For gene selection, Alirezanejad et al. [72] developed two Xvariance heuristics against mutual congestion. This approach involves ranking the features first. Then, using Monte's cross-validation, ten subsets of features are chosen based on forward feature selection (FFS). To enhance the results, majority voting is applied to the features selected in the prior stage to calculate accuracy, sensitivity, specificity, and matthews correlation coefficient.

Asghari Varzaneh et al. [73] proposed a new COVID-19 intubation prediction strategy using the binary version of the horse herd optimization algorithm to select the effective features. The results of the tests showed that the proposed feature selection method is better than other methods. Pashaei et al. [74] introduced two binary variations of the chimp optimization algorithm using S-shaped and V-shaped transfer functions for biomedical data classification. In a recent study, Nadimi-Shahraki et al. [75] proposed the binary version of the quantum-based avian navigation optimizer algorithm (BQANA) to select the optimal feature subset from high-dimensional medical datasets. The reported results show that the BQANA using a threshold method can dominate all contender algorithms. Alweshah et al. [76] proposed the greedy crossover (GC) operator strategy to boost the exploration capability of the coronavirus herd immunity optimizer (CHIO). Then, some medical datasets were used to evaluate the performance of the proposed algorithm in addressing the feature selection problem in the field of medical diagnosis. The results indicated that the GC operator strikes a balance between the search strategies of the CHIO algorithm.

For challenges involving medical feature selection, Anter et al. [77] proposed a hybrid crow search optimization algorithm combined with chaos theory and a fuzzy c-means algorithm (CFCSA). The suggested algorithm avoids local optima and improves the CSA's convergence using chaos theory and the global optimization method. The test results show the efficiency and stability of CFCSA for solving medical data and real problems. Singh et al. [78] proposed a hybrid ensemble-filter wrapper feature selection algorithm to improve the performance of classifiers in medical data applications. In this algorithm, first, the filter-based method is used based on the weight points to produce the ranking of the features. Then, the sequential forward selection algorithm is used as a wrapper-based feature selection to generate an optimal feature subset. To propose the binary version of the atom search optimization algorithm (ASO), Too et al. [79] applied four S-shaped and four V-shaped transfer functions to solve the feature selection problem. Among the eight presented binary versions, BASO based on the S1–shaped transfer function has the highest performance. Moreover, Mirjalili et al. [67] proposed a new binary version of the PSO algorithm using a U-shaped transfer function to transform continuous velocity values into binary values. The results show that U-shaped transfer functions significantly increase the performance of BPSO.

Elgamal et al. [80] enhanced the reptile search optimization algorithm (RSA) by employing the chaotic map and simulated annealing algorithm to tackle feature selection issues for high-dimensional medical datasets. Applying chaos theory to RSA improves its exploration ability, and hybridizing RSA with the simulated annealing algorithm can avoid local optima trapping. Many metaheuristic algorithms have been proposed to solve feature selection problems, such as binary ant lion optimizer (BALO) [81], return-cost-based binary firefly algorithm (Rc-BBFA) [82], chaotic dragonfly algorithm (CDA) [83], binary chimp optimization algorithm (BChOA) [84], altruistic whale optimization algorithm (AltWOA) [85],

binary African vulture optimization algorithm (BAVOA) [86], and binary dwarf mongoose optimization algorithm (BDMSAO) [87].

Studying related works shows that various metaheuristic algorithms have been used to select effective features from medical data. However, most of them cannot find effective features for providing an acceptable diagnosis of important diseases such as Diabetes, Heart, Hepatitis, and Coronavirus. To respond to this weakness, the BSMO algorithm is introduced to develop a new wrapper feature selection method for these diseases in this study.

3. Starling Murmuration Optimizer (SMO)

SMO is a population-based metaheuristic algorithm recently developed by Zamani et al. [41]. The SMO algorithm is modeled the starlings' behavior during their stunning murmuration using three new search strategies, separating, diving, and whirling. The starling's population is denoted by $S = \{s_1, s_2, \ldots, s_N\}$ where N is the population size. The position of each starling s_i at iteration t is denoted using a vector $X_i(t) = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ and its fitness value is expressed by $F_i(t)$. In first iteration, each $X_i(t)$ is initiated by a uniform random distribution in a D-dimensional search space using Equation (1), where X^L and X^U are lower and upper bounds of the search space, respectively and $rand\ (0,1)$ is a random value between 0 and 1.

$$X_i(t) = X^L + rand(0,1) \times (X^U - X^L), \quad i = 1, 2, \ldots., N \tag{1}$$

For the rest of the iterations, the population of starlings is moved using the separating, diving, and whirling search strategies. The details of these search strategies are discussed in the following sections.

3.1. Separating Search Strategy

The separation search strategy is promoted diversity throughout the population. In this strategy, first, a portion of starlings with size P_{sep} are randomly selected to separate from population S using Equation (2). Then, some dimensions of the selected starlings are updated using Equation (3), where $X_G(t)$ is the global best position, and $X_r(t)$ is randomly selected from a population S. In each iteration, the best position obtained so far is stored, then these positions are joined with the separated positions with size P_{sep}, ultimately $X_{r'}(t)$ is randomly selected from these sets. $Q_1(y)$ is a separation operator which is calculated using Equation (4), where α is the quantum harmonic oscillator, parameters m and k are the particle's mass and strength, respectively and the parameter h is Planck's constant. Moreover, the function H_n is the Hermite polynomial with integer index n, and y is a random number.

$$P_{sep} = \frac{\log(t+D)}{\log(MaxIt) \times 2} \tag{2}$$

$$X_i(t+1) = X_G(t) + Q_1(y) \times (X_{r'}(t) - X_r(t)) \tag{3}$$

$$\mathcal{Q}_1(y) = \left(\frac{\alpha}{2^n \times n! \times \pi^{\frac{1}{2}}}\right)^{\frac{1}{2}} \mathrm{H_n}(\alpha \times \mathrm{y}) \times e^{-0.5 \times \alpha^2 \times \mathrm{y}^2},\ \alpha = \left(\frac{m \times k}{\hbar}\right)^{\frac{1}{4}} \tag{4}$$

The rest of the starlings with a size of $\acute{N}$ $(N - P_{sep})$ is flocked using dynamic multi-flock construction to search the problem space using either diving or whirling search strategies. Each iteration creates a dynamic multi-flock using k non-empty flocks $f_1 \ldots f_k$. First, k best starlings are separated from the population $\acute{N}$ and stored in matrix R, then the rest of the population ($\acute{N}$-R) is divided among the k flocks. Finally, each position of R assigns to each flock such that $f_{1\leftarrow}\ \{R_1\ U f_1\}, \ldots, f_{k\leftarrow}\ \{R_k\ U f_k\}$.

As shown in Equation (6), the diving and whirling search strategies are assigned to the flocks based on the quality of each flock. The quality of each flock ($Q_q\ (t)$) is evaluated using Equation (5), where k is the number of flocks, $sf_{ij}\ (t)$ is the fitness value of the starling

s_i in the flock f_j, and n is the number of starlings in each flock. The parameter μ_Q *(t)* in Equation (6) denotes the average of all flock's quality.

$$Q_q(t) = \frac{\sum_{i=1}^{k} \frac{1}{n} \sum_{j=1}^{n} sf_{ij}(t)}{\frac{1}{n} \sum_{i=1}^{n} sf_{qi}(t)} \tag{5}$$

$$X_i(t+1) = \begin{cases} \text{Diving search strategy} & Q_q(t) \leq \mu_Q(t) \\ \text{Whirling search strategy} & Q_q(t) > \mu_Q(t) \end{cases} \tag{6}$$

3.2. Diving Search Strategy

The diving search strategy is encouraged the selected flocks (Q_q *(t)* $\leq$ μ_Q *(t)*) to explore the search space effectively. The starlings are moved using upward and downward quantum random dives (*QRD*). The starlings of a flock switch among these quantum dives using two quantum probabilities shown in Equation (7), where $\left|\psi^{Up}(X_i)\right|$ and $\left|\psi^{Down}(X_i)\right|$ are the upward and downward probabilities that are computed using Equations (8) and (9). Parameters φ and θ are set by the user, and $|\psi(\delta_2)\rangle$ is an inverse-Gaussian distribution that is computed using Equation (10), where the values of λ and μ are set by the user, and y is a random number.

$$QRD = \begin{cases} \text{Upward quantum dive} & \left|\psi^{Up}(X_i)\right| > \left|\psi^{Down}(X_i)\right| \\ \text{Downward quantum dive} & \left|\psi^{Up}(X_i)\right| \leq \left|\psi^{Down}(X_i)\right| \end{cases} \tag{7}$$

$$|\psi^{Up}(X_i)\rangle = e^{i\varphi} \cos\theta \times |\psi(\delta_2)\rangle - e^{-i\varphi} \sin\theta \times |\psi(\delta_2)\rangle \tag{8}$$

$$|\psi^{Down}(X_i)\rangle = e^{i\varphi} \sin\theta \times |\psi(\delta_2)\rangle + e^{-i\varphi} \cos\theta \times |\psi(\delta_2)\rangle \tag{9}$$

$$|\psi(\delta_2)\rangle = \sqrt{\frac{\lambda}{2 \times \pi \times y^3}} \times e\left[-\frac{\lambda(y-\mu)^2}{2 \times \mu^2 \times y}\right] \tag{10}$$

The downward and upward quantum dives are computed using Equations (11) and (12), respectively, where $|\psi(R_D)\rangle$ is selected from set R, $|\psi(X_i)\rangle$ is the position of starling s_i in the current iteration, the position of $|\psi(X_r)\rangle$ is randomly selected among flocks assigned for diving strategy, $\left|\psi(X_j)\right\rangle$ is randomly selected from the population S and the best starlings set. $|\psi(\delta_1)\rangle$ is a random position selected from the best starlings set obtained from the first iteration so far and the starling population S.

$$|\psi(t+1,\ X_i)\rangle = |\psi(R_D)\rangle - |\psi^{Down}(X_i)\rangle \times (|\psi(X_i)\rangle - |\psi(X_r)\rangle) \tag{11}$$

$$|\psi(t+1,\ X_i)\rangle = |\psi(R_D)\rangle + |\psi^{Up}(X_i)\rangle \times \left(|\psi(X_i)\rangle - |\psi(X_j)\rangle + |\psi(\delta_1)\rangle\right) \tag{12}$$

3.3. Whirling Search Strategy

Starlings of a flock exploit the search problem using the whirling search strategy when the quality of the flock is more than the average quality of all flocks *(Q_q (t) > μ_Q (t)).* The whirling search strategy is denoted in Equation (13), where X_i *(t+1)* is the next position of starling s_i at iteration t, a position X_{RW} *(t)* is randomly selected from set R of flocks that are considered for the whirling search strategy, X_N *(t)* randomly selected from all flocks that want to use the whirling search strategy. C_i *(t)* is the cohesion operator which is calculated using Equation (14), where ξ *(t)* is a random number between intervals 0 and 1.

$$X_i(t+1) = X_i(t) + C_i(t) \times (X_{RW}(t) - X_N(t)) \tag{13}$$

$$C_i(t) = cos(\xi(t)) \tag{14}$$

The pseudocode of the SMO algorithm is shown in Algorithm 1.

Algorithm 1: Starling Murmuration Optimizer (SMO)

Input: N (Population size), k (Flocks size), and $MaxIt$ (Maximum iterations).
Output: Global best solution.

Begin
Randomly distributed N starlings in the search space.
Set $t = 1$.
While $t \leq MaxIt$
  Separating a portion of starlings with size P_{sep} from the population using Equation (2).
  The rest of the population is flocked into k flocks using the dynamic multi-flock construction.
  Computing the quality of each flock (fq) using Equation (5).
  For $q = 1$: k
    If $Q_q\ (t) \leq \mu_Q\ (t)$
      Moving starlings of the flock fq using the diving strategy.
    Else
      Moving starlings of the flock fq using the whirling strategy.
    End if
  End for
  Update the position of starlings and global best solution.
  $t = t + 1$.
End while
Return position of best starling as a global best solution.
End

4. Binary Starling Murmuration Optimizer (BSMO)

SMO is a new metaheuristic algorithm that effectively solves various engineering and complex problems. However, the ability of the SMO algorithm to solve feature selection problems has not been studied yet, which is the motivation of this study. In this study, a binary starling murmuration optimizer (BSMO) is proposed to select effective features from the datasets of four important targeted diseases consisting Diabetes, Heart problems, Hepatitis, and Coronavirus. The proposed BSMO is developed using two different approaches. The first approach uses S-shaped and V-shaped transfer functions, whereas the second approach maps the continuous search space to 0 or 1 using a threshold value.

Suppose matrix X is to represent the population of starlings in the BSMO, then Figure 1 shows the representation scheme of the proposed BSMO algorithm in solving the feature selection problem. Figure 1a–c show starling S_i, binary vector B_i, and the selected feature set SF_i. Each starling S_i is transformed using different transform functions to the binary vector B_i in which the value of 1 for each element means the corresponding feature should be selected to form the selected feature set SF_i. Accordingly, the BSMO algorithm uses the fitness function defined in Equation (15) [83,88].

$$Fit_i = \alpha E + \beta \frac{|SF_i|}{D} \tag{15}$$

where E determines the error rate of the classification algorithm, $|SF_i|$ and D are the number of the selected feature in a subset of SF_i, and the total features in the dataset, respectively. α and $\beta = 1 - \alpha$ are two constant values to control the significance of the classification accuracy and feature subset reduction, respectively. Since the accuracy is more important of the number of features, usually β is very smaller than α, in this study, $\alpha = 0.99$ and $\beta = 0.01$, according to [89].

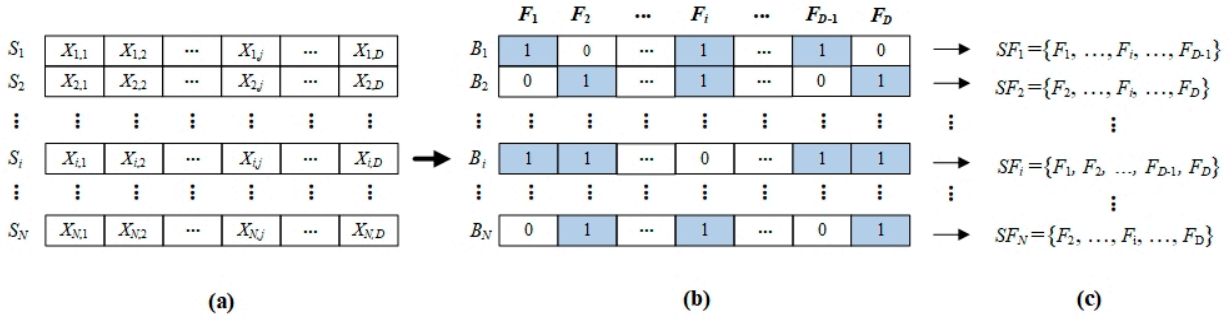

Figure 1. The representation scheme used by BSMO, (**a**) Starling population (Matrix X), (**b**) Binary population (Matrix B), and (**c**) Selected features (SF).

4.1. BSMO Using S-Shaped Transfer Function (S-BSMO)

This method uses the sigmoid transfer function (S-shape) to map the continuous to the binary version of the SMO algorithm. Therefore, updating the position of the starlings by the transfer functions *S* will cause them to be in a binary search space, and their position vector will only take the values of "0" or "1". The sigmoid function *S2* formulated in Equation (16) first used in BPSO to develop a binary PSO [89,90].

$$S\left(x_i^d(t+1)\right) = \frac{1}{1+e^{-x_i^d(t)}} \tag{16}$$

where $x_i^d(t)$ and $S\left(x_i^d(t+1)\right)$ show the position and probability of changing the binary position value of the search agent i^{th} in dimension d in the t^{th} iteration, respectively. Since the calculated value of *S* is still in continuous mode, it must be compared with a threshold value to create binary mode. Therefore, the new position of the search agent is updated using Equation (17), where $b_i^d(t+1)$ is a binary position of i^{th} search agent in dimension d, and r is a random value between 0 and 1.

$$b_i^d(t+1) = \begin{cases} 0 & if \quad r < S\left(x_i^d(t+1)\right) \\ 1 & if \quad r \geq S\left(x_i^d(t+1)\right) \end{cases}, \tag{17}$$

In addition to the transfer function *S2* introduced in Equation (16), three other types of S-shaped transfer functions, including *S1*, *S3*, and *S4* have been used. All four transfer functions are formulated in Table 1. Moreover, all these transfer functions are shown visually in Figure 2. According to the figure, as the slope of the transfer function *S* increases, the probability of changing the position value increases. Therefore, *S1* obtains the highest probability, and *S4* obtains the lowest probability, effectively updating agents' position and finding the optimal solution.

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

Name	S-Shaped Transfer Functions	Name	V-Shaped Transfer Functions
S1-shaped	$T(x) = \frac{1}{1+e^{-2x}}$	V1-shaped	$T(x) = \left\lvert \mathrm{erf}\left(\frac{\sqrt{\pi}}{2}x\right)\right\rvert$
S2-shaped	$T(x) = \frac{1}{1+e^{-x}}$	V2-shaped	$T(x) = \lvert \tan \mathrm{h}(x)\rvert$
S3-shaped	$T(x) = \frac{1}{1+e^{\frac{-x}{2}}}$	V3-shaped	$T(x) = \left\lvert \frac{x}{\sqrt{1+x^2}}\right\rvert$
S4-shaped	$T(x) = \frac{1}{1+e^{\frac{-x}{3}}}$	V4-shaped	$T(x) = \left\lvert \frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right\rvert$

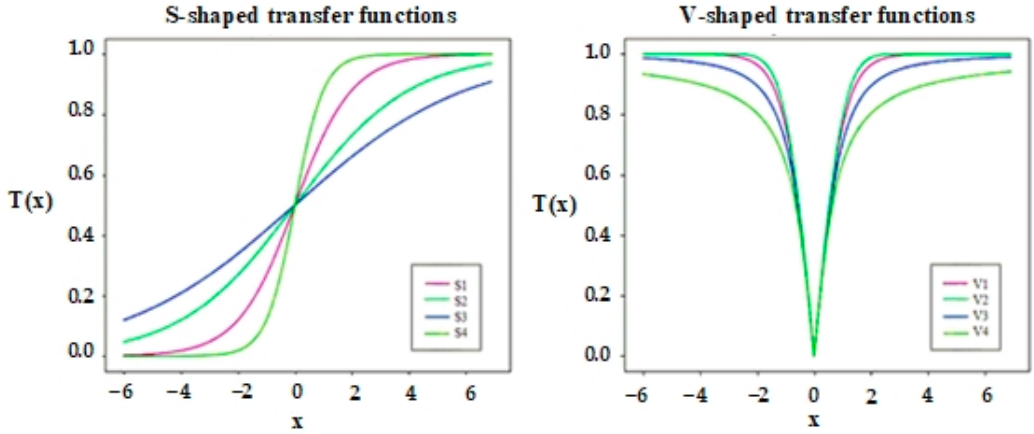

Figure 2. The S-shaped and V-shaped transfer functions [89].

4.2. BSMO Using V-Shaped Transfer Function (V-BSMO)

In this approach, the V-shaped transfer function is used to calculate the probability of changing the position of the agents in the SMO algorithm. Probability values are calculated using the V-shaped (hyperbolic) transfer function by Equation (18) [64], where $x_i^d(t)$ indicates the position value of the i^{th} search agent in dimension d at iteration t.

$$V\left(x_i^d(t+1)\right) = \left|tanh\left(x_i^d(t)\right)\right| \tag{18}$$

Considering that the V-shaped transfer function is different from the S-shaped transfer function, after calculating the probability values, the Equation (19) [64] is used to update the position of each search agent.

$$b_i^d(t+1) = \begin{cases} x_i^d(t)^{-1} & if \quad r < V\left(x_i^d(t+1)\right) \\ x_i^d(t) & if \quad r \geq V\left(x_i^d(t+1)\right) \end{cases} \tag{19}$$

where, $b_i^d(t+1)$ indicates the binary position of the i^{th} search agent at iteration $t+1$ in dimension d. Moreover, $x_i^d(t)^{-1}$ indicates the complement of $x_i^d(t)$. In addition, r is a random number in [0,1]. Unlike the S-shaped transfer function, the V-shaped transfer function does not force the search agents into 0 or 1. According to Equation (19), if the value of V is small and less than the value of r, the binary position of the search agents in dimension d will not change. On the other hand, if the calculated value of the transfer function is greater than or equal to the value r, the position of the search agents is changed to the complement of the current binary position. Table 1 formulates the mathematical equations of transfer functions *V1*, *V2*, *V3*, and *V4*, and Figure 2 represents transfer functions visually. According to Figure 2, *V1* has the highest probability, and *V2*, *V3*, and *V4* have lower probability values for moving the positions of search agents, respectively [89].

4.3. BSMO Using Variable Threshold Method (Threshold-BSMO)

In this section, the SMO transforms the continuous solutions into the binary form using the variable threshold method defined in Equation (20), where $b_i^d(t+1)$ is a new binary position of the i^{th} search agent, and a variable threshold θ is 0.5 that is set by the user.

$$b_i^d(t+1) = \begin{cases} 1 & if \quad x_i^d(t+1) > \theta \\ 0 & if \quad x_i^d(t+1) \leq \theta \end{cases} \tag{20}$$

Figure 3 represents the flowchart of the proposed BSMO algorithm, which is a binary version of the SMO algorithm to solve the feature selection problem. As shown in this figure, the optimization process is started by initializing the input variables, including a maximum number of iterations (*MaxIt*), population size (*N*), problem size (*D*), and flocks size (*k*). First, *N* starlings are randomly distributed in a *D*-dimensional search space. Then, a portion of starlings (P_{sep}) using Equation (2) are randomly selected to separate from the population and explore the search space using the separating strategy defined in Equation (3). The rest of the starlings are partitioned between different flocks to exploit the search space using the whirling strategy defined in Equation (13) or explore using the diving strategy defined in Equation (7). The obtained solutions from such search strategies are mapped to binary using two binarization approaches demonstrated in Table 1 and Equation (20). The obtained solutions are restricted to binary values 0 or 1 using Equations (17), (19), and (20). Finally, the solutions are evaluated using Equation (15). The optimization process is repeated until the termination condition, or MaxIt, is satisfied, and the global best solution is reported as the output variable.

4.4. The Computational Complexity of the BSMO Algorithm

Since BSMO has six distinct phases: initialization, separating search strategy, multi-flock construction, diving or whirling search strategy, mapping, and fitness evaluation, its computational complexity can be computed as follows. The initialization phase's computational complexity is O (*ND*), considering *N* starlings are randomly allocated in a *D*-dimensional search space using Equation (1). Then, a portion of the starlings is randomly selected using Equation (2) to explore the search space with computational complexity O (*ND*). The cost of the multi-flock construction phase to build *k* flocks by partitioning *N* starlings is O (*NlogN* + *k*). In the next phase, the cost of each flock containing *n* subpopulation for determining its quality utilizing Equation (5) is O (*nD*), and for moving by either diving or whirling search strategy is also O (*nD*). Thus, the overall complexity of this phase is O (*knD*) or O (*ND*) in the worst case. In the mapping phase, the continuous solutions are transformed into binary ones based on Table 1 and Equation (20) with computational complexity O (*ND*). Finally, in the fitness evaluation phase, the quality of binary solutions is assessed using Equation (15), consisting of a K-fold cross-validation method, k-NN classifier, and updating. The computational complexity of a K-fold cross-validation method with *M* samples is O (*KM*). Since *K* is a constant value, complexity equals O (*M*). The k-NN classifier with *M* samples and *D* features for training the classifier is O (*MD*), and the complexity of updating is O (*ND*). Since these phases are repeated *T* times, therefor the summation of the computational complexity of BSMO is O (*ND* + *T* (*ND* + (*NlogN*+*k*) + *ND* + *ND* + *M* + *MD* + *ND*)), which is equal to *O* (*TD* (*N*+*M*)).

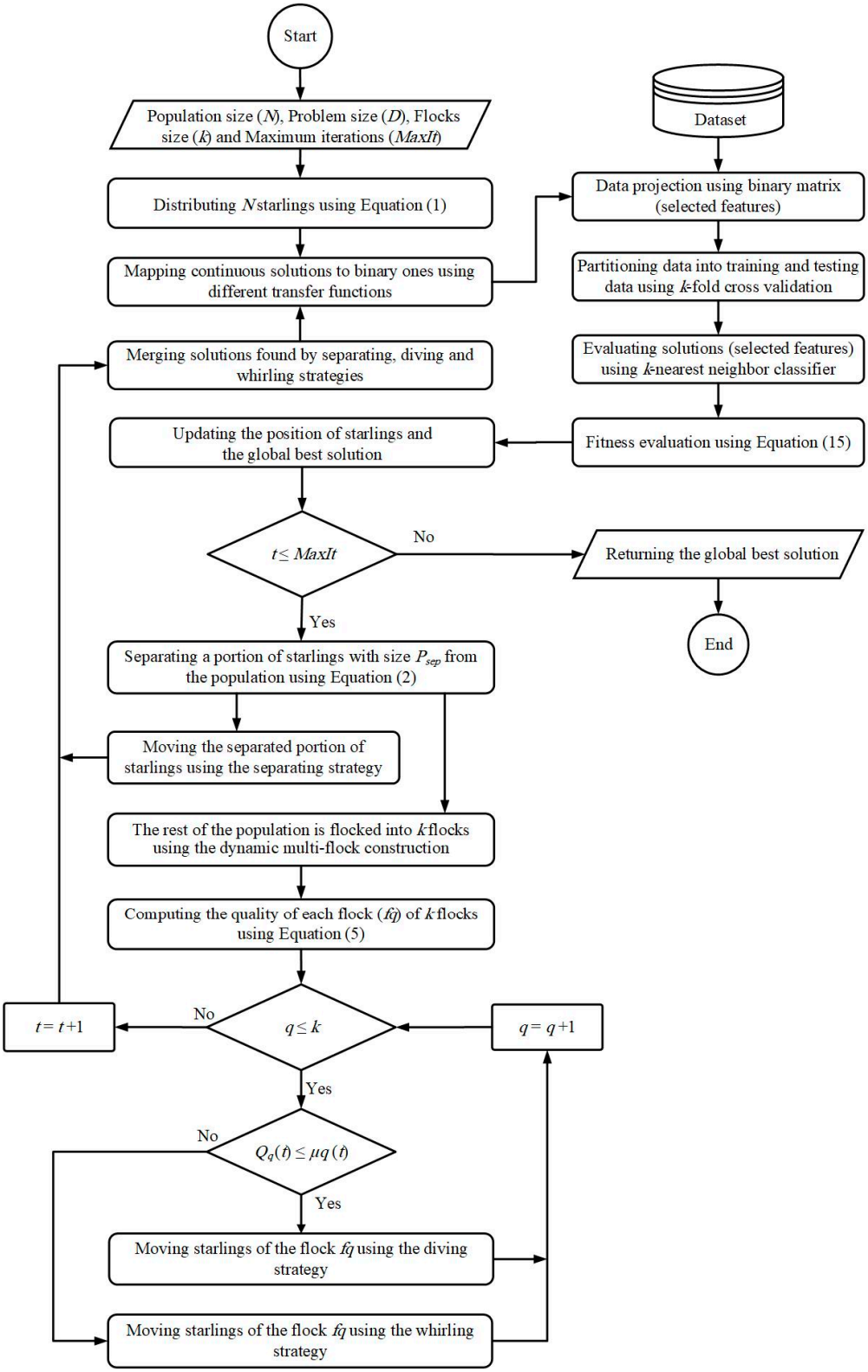

Figure 3. Flowchart of the proposed BSMO algorithm.

5. Experimental Evaluation

The performance of the proposed BSMO algorithm is assessed in finding the optimal feature subset from targeted datasets, Diabetes, Heart, Hepatitis, and Coronavirus diseases 2019, downloaded from [91,92]. Then, the nine BSMO variants' outcomes are then compared with those of competitive algorithms, ACO [28], BBA [42], bGWO [43], and BWOA [39].

All experiments are run under the same experimental conditions. MATLAB R2019b programming language is considered for implementing the BSMO and running all comparative algorithms. All experiments are run using an Intel (R) Core (TM) i5-3770 CPU, 3.4 GHz, 8 GB RAM, and Windows 10 with the 64-bit operating system.

5.1. Parameter Settings of Algorithms and k-NN Classifier

In this study, the k-nearest neighbor (*k-NN*) classifier with $k = 5$ is used to classify the feature subsets in all algorithms [93]. To learn the *k-NN* classifier, each dataset is randomly partitioned using a *K*-fold cross-validation method into training and testing sets, where *K* is a constant value equal to 10. One fold is used for the testing set, and the $K-1$ folds are applied for the training set [94,95].

For a fair comparison, all results were obtained under the same experimental conditions. The common parameters in BSMO and comparative algorithms, such as termination criterion and population size (*N*), are the same. In most optimization algorithms, the termination criterion is defined using the maximum number of iterations (*MaxIt*) or maximum function evaluations (*MaxFEs*), where $MaxIt = MaxFEs/N$ and it is set to 300 and *N* is 30. Due to the stochastic nature of the algorithms, all simulations and obtained results are conducted with 15 independent runs. All results are reported using the standard statistical metrics maximum (*Max*), average (*Avg*), and minimum (*Min*) values. In each table, the best result is highlighted in boldface.

Table 2 shows the values of parameters used for BSMO and other comparative algorithms. The parameter values of all contender algorithms were set as same as their original papers. Moreover, a sensitivity analysis on key parameters of the BSMO algorithm, such as flock size (*k*), and population size (*N*), is performed to tune the values of these parameters using the offline parameter tuning method. The tuning results were reported in Tables A1–A6 of Appendix A in terms of fitness, error, accuracy, sensitivity, specificity, and precision metrics.

Table 2. Parameters setting.

Algorithms	Parameters
ACO	$\tau = 1$, $\eta = 1$, $\rho = 0.2$, $\alpha = 1$, and $\beta = 0.1$
BBA	$Q_{min} = 0$ and $Q_{max} = 2$
bGWO	a linearly decreases from 2 to 0, C_1, C_2, and C_3 are a random numbers
BWOA	a linearly decreases from 2 to 0, $b = 1$, r_1 and $r_2 \in$ rand (0, 1)
BSMO	$k = 5$, $\lambda = 20$, $\mu = 0.5$, θ and $\phi \in (0, 1.8)$

5.2. Evaluation Criteria

The performance of proposed BSMO and contender algorithms are assessed using evaluation criteria such as fitness, accuracy, sensitivity, specificity, precision, and error. The fitness evaluation metric is computed using Equation (15). The accuracy, sensitivity, specificity, precision, and error are calculated using Equations (21)–(25) [96,97]. In these equations, parameters *TP* and *TN* specify the number of positive and negative samples that are correctly classified by the classifier, respectively. *FN* is the number of positive samples incorrectly predicted as negative, and FP is the number of negative samples incorrectly predicted as positive using a classifier [98].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

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

$$\text{Specificity} = \frac{TN}{TP + FN} \tag{23}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{24}$$

The error metric is computed using the mean square error (MSE) denoted in Equation (25), where N is the number of samples, y_i is the observed values and $\hat{y}_i$ is the predicted value. Moreover, evaluating the proposed algorithm does not use any constraint handling methods since no constraints are considered in the feature selection problem.

$$\text{Error} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{25}$$

5.3. Numerical Results and Discussion

In this section, the simulation results of the proposed BSMO algorithm are presented on targeted medical datasets.

5.3.1. Comparison of Algorithms to Detect Diabetes Disease

The Pima Indian Diabetes dataset [91] consists of eight features, 268 samples with diabetes-positive labeling and 500 samples with diabetes-negative. The objective of this dataset is to detect whether or not a patient has diabetes. Table 3 shows that the proposed Threshold-BSMO can achieve the best performance compared to all comparative algorithms.

Table 3. Diabetes disease detection.

Algorithms	Fitness		Accuracy		Sensitivity		Precision		Specificity		Error	
	Avg	Min	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Min
ACO	0.2384	0.2318	76.5109	77.0865	85.2345	86.6173	60.1832	64.0414	79.9663	82.1451	0.2351	0.2291
BBA	0.2331	0.2281	76.9974	77.4675	86.4089	88.748	79.8734	83.4279	59.365	63.0096	0.23	0.2253
bGWO	0.2295	0.2253	77.3573	77.8725	86.2124	89.3135	80.0664	83.5267	59.8209	65.9114	0.2264	0.2213
BWOA	0.2386	0.2344	76.4744	76.825	85.8432	87.8664	79.8754	82.3961	59.5944	64.961	0.2353	0.2317
S1-BSMO	0.2342	0.2266	76.9719	77.7409	88.5142	89.8454	83.2288	84.242	65.9602	68.2916	0.2504	0.2382
S2-BSMO	0.2352	0.2267	76.8537	77.7341	88.2422	89.2631	83.1426	84.2726	65.7932	68.023	0.2516	0.2369
S3-BSMO	0.2373	0.2291	76.6101	77.4897	88.1787	90.1796	82.925	84.4974	65.5806	67.8662	0.2508	0.2397
S4-BSMO	0.2368	0.2291	76.6654	77.4863	88.3085	89.7088	82.764	83.8476	65.0295	66.8104	0.2533	0.2384
V1-BSMO	0.2344	0.2294	76.889	77.3411	88.2848	89.7132	83.1764	86.1848	65.7345	70.5787	0.2552	0.2422
V2-BSMO	0.2343	0.2266	76.8872	77.6128	88.6261	90.0085	82.9072	83.761	65.7846	67.6503	0.2548	0.2345
V3-BSMO	0.2353	0.2306	76.7716	77.2163	88.245	89.6911	83.1812	84.5091	66.0204	69.1626	0.2547	0.2383
V4-BSMO	0.2335	0.2292	76.9639	77.471	88.1009	89.484	83.2658	84.4214	66.2564	69.1896	0.2534	0.2383
Threshold-BSMO	0.2306	0.2229	77.3077	77.9904	89	89.9871	83.5823	84.7376	66.6321	69.2028	0.253	0.2408

5.3.2. Comparison of Algorithms to Detect Heart Disease

The Statlog (Heart) dataset [91] consists of 13 features and 270 samples without no missing values to detect the absence or presence of heart disease. In this dataset 120 of the samples are labeled with the presence of heart disease and 150 samples are labeled with the absence of this disease. The performance of the proposed BSMO with nine variants is assessed and compared with well-known optimizers to diagnose heart disease. The results in Table 4 show that the proposed Threshold-BSMO can obtain a minimum fitness value of 0.1322 and a maximum accuracy of 87.037 than other algorithms.

Table 4. Heart disease detection.

Algorithms	Fitness		Accuracy		Sensitivity		Precision		Specificity		Error	
	Avg	Min	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Min
ACO	0.147	0.1387	85.4815	86.2963	88.8186	94.1537	86.8452	89.665	82.7764	86.6325	0.1452	0.137
BBA	0.1414	0.1380	86.0123	86.2963	94.0096	95.4345	89.7266	91.4855	88.5579	91.1526	0.1959	0.1519
bGWO	0.1383	0.1358	86.4198	86.6667	87.4898	93.0586	85.4259	90.2422	80.7175	87.4738	0.1578	0.1444
BWOA	0.1409	0.1387	86.1728	86.2963	89.4656	91.3609	86.9606	90.1189	82.8787	88.087	0.1383	0.137
S1-BSMO	0.151	0.1432	85.1852	85.9259	89.2216	95.26	83.6512	89.9588	78.67	87.0474	0.1481	0.1407
S2-BSMO	0.146	0.1411	85.8148	86.2963	93.6608	95.0876	89.2841	91.4817	86.0433	88.7512	0.1964	0.1593
S3-BSMO	0.1481	0.1424	85.5185	85.9259	93.3517	95.3351	89.403	91.5718	86.2794	88.2128	0.2015	0.1556
S4-BSMO	0.1495	0.1432	85.3333	85.9259	93.1475	94.4033	89.7136	91.6581	87.0123	89.3531	0.1930	0.1556
V1-BSMO	0.1492	0.1403	85.3704	86.2963	93.2132	95.0297	89.4763	91.4379	86.3764	89.284	0.1907	0.1481
V2-BSMO	0.1423	0.1387	85.9383	86.2963	93.8571	96.2621	89.3417	91.9558	89.0497	91.2747	0.1884	0.1593
V3-BSMO	0.1417	0.1380	86.037	86.2963	94.4918	96.2525	89.3503	91.9198	88.4579	91.1828	0.1911	0.1481
V4-BSMO	0.1411	0.1351	86.0741	86.6667	94.1042	95.6443	89.6908	91.8579	88.5817	90.503	0.1956	0.1667
Threshold-BSMO	0.1371	0.1322	86.5432	87.037	89.8998	93.4192	86.7337	90.5212	82.2366	87.3123	0.1346	0.1296

5.3.3. Comparison of Algorithms to Detect Hepatitis Disease

The Hepatitis disease dataset [91] is complex with many missing values that contain occurrences of hepatitis in people. This dataset consists of 19 features with 155 samples, of which 123 samples are categorized in the live class, and 32 are categorized in the die class. The optimization algorithms try to find the best feature set which can detect Hepatitis disease with high accuracy. In this evaluation, the performance of the proposed algorithm is assessed and reported in Table 5. The results show that the BSMO using the variable threshold can obtain the optimum feature set with a minimum fitness value. Additionally, the Threshold-BSMO achieves the highest classification accuracy compared to the contender algorithm.

Table 5. Hepatitis disease detection.

Algorithms	Fitness		Accuracy		Sensitivity		Precision		Specificity		Error	
	Avg	Min	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Min
ACO	0.1215	0.1074	88.0639	89.625	64.5377	76.411	94.4719	97.8957	75.7176	89.8369	0.1194	0.1037
BBA	0.1116	0.0977	89.1083	90.5	64.4286	80.5122	78.7006	90.214	95.0395	97.9604	0.109	0.095
bGWO	0.1067	0.0932	89.5417	90.9583	63.8564	82.9117	79.1231	85.5983	95.3145	97.5229	0.1046	0.0904
BWOA	0.1209	0.1135	88.1806	88.9583	60.8305	74.3306	78.0184	93.4557	95.2697	98.8117	0.1182	0.1104
S1-BSMO	0.1265	0.1147	87.8319	89	70.6404	80.2298	81.3914	95.0256	99.422	100	0.1659	0.1292
S2-BSMO	0.1218	0.1118	88.1708	89.1667	70.8924	84.532	78.9332	91.5289	99.4674	100	0.1598	0.1171
S3-BSMO	0.1213	0.1051	88.2153	89.9167	71.8705	85.8738	81.3385	96.9048	99.377	100	0.1599	0.1237
S4-BSMO	0.1209	0.1070	88.2306	89.6667	72.851	82.1369	81.9163	93.1111	99.3568	100	0.1603	0.1296
V1-BSMO	0.1109	0.0977	89.1542	90.5	78.8832	85.8624	83.8414	95.5556	99.471	100	0.1587	0.1292
V2-BSMO	0.1106	0.0998	89.2069	90.375	79.3521	87.3972	84.3151	96.3492	99.2964	99.9187	0.1589	0.1342
V3-BSMO	0.1107	0.0994	89.1986	90.375	78.5909	86.1964	85.7139	97.5	99.4433	100	0.1617	0.1412
V4-BSMO	0.1096	0.0990	89.3278	90.375	79.7051	88.4275	84.2503	98.75	99.4127	100	0.1617	0.1425
Threshold-BSMO	0.1081	0.0924	89.5194	91.0417	80.2438	91.3715	85.1981	95.7778	99.4531	100	0.1623	0.1342

5.3.4. Comparison of Algorithms to Detect Coronavirus Disease 2019 (COVID-19)

The COVID-19 pandemic is an infectious disease of severe acute respiratory syndrome Coronavirus 2019 [99] which was initiated in Wuhan, China, in December 2019 and profoundly affected human life [100]. Early detection of Coronavirus disease can reduce the transmission rate and slow the epidemic outbreak. Many optimization algorithms have been developed to alleviate this global crisis [101]. In this section, the performance of the proposed algorithm is evaluated in the Coronavirus disease 2019 (COVID-19) dataset [92]. This dataset consists of two classes, death or recovery, and 13 features, including location, country, gender, age, whether the patients visited Wuhan, whether the patients from Wuhan had fever, cough, cold, fatigue, body pain, malaise, and day's difference between the symptoms being noticed and admission to the hospital. The results reported in Table 6 indicate the proposed Threshold-BSMO outperforms all contender algorithms and BSMO variants to detect COVID-19.

Table 6. Coronavirus disease 2019 (COVID-19) detection.

Algorithms	Fitness		Accuracy		Sensitivity		Precision		Specificity		Error	
	Avg	Min	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg	Min
ACO	0.0521	0.0493	95.2805	95.4825	98.3325	99.0601	96.3844	97.4589	74.0994	78.5774	0.0477	0.0452
BBA	0.0508	0.0494	95.3575	95.4838	98.411	98.9281	96.5039	97.1542	74.7778	79.4731	0.0464	0.0452
bGWO	0.0482	0.0455	95.4915	95.7137	98.6061	99.3678	96.1273	97.5426	73.3757	80.9149	0.0451	0.0429
BWOA	0.0518	0.0493	95.2667	95.7164	98.3045	99.0229	96.4153	97.1998	74.626	82.0579	0.0479	0.0428
S1-BSMO	0.0515	0.0493	95.417	95.5988	99.2906	99.7496	97.7266	98.2616	83.0173	87.3208	0.0511	0.0452
S2-BSMO	0.0516	0.049	95.3861	95.5961	99.3947	100	97.5923	97.9672	82.1743	85.5629	0.052	0.0498
S3-BSMO	0.0517	0.0497	95.3308	95.6001	99.3703	100	97.5576	98.2389	81.7173	87.298	0.0521	0.0487
S4-BSMO	0.0516	0.049	95.3347	95.5948	99.4093	100	97.5954	98.126	82.1407	86.2074	0.0532	0.0498
V1-BSMO	0.051	0.0497	95.2469	95.5974	99.8598	100	97.3384	97.924	80.376	84.761	0.0537	0.0476
V2-BSMO	0.0509	0.0489	95.263	95.4812	99.8182	100	97.364	97.8237	80.6954	84.0749	0.053	0.0474
V3-BSMO	0.051	0.0486	95.2695	95.4838	99.7693	100	97.3319	97.9259	80.477	83.9283	0.053	0.0475
V4-BSMO	0.0506	0.0478	95.2692	95.4892	99.7845	100	97.4058	97.956	80.7991	84.4487	0.0532	0.0452
Threshold-BSMO	0.0488	0.0451	95.537	95.8353	99.3774	100	97.7178	98.0502	83.1011	87.2075	0.0518	0.0487

5.4. Convergence Comparison

In addition, to compare the efficiency of BSMO with other comparative algorithms, convergence curves were drawn for each dataset used in the evolution. Figure 4 shows the convergence curves of all algorithms based on the fitness value. According to the figure, Threshold-BSMO has the highest efficiency in diagnosing Diabetes, Hepatitis, Heart, and Coronavirus 2019 diseases with the lowest fitness value compared to competitive algorithms.

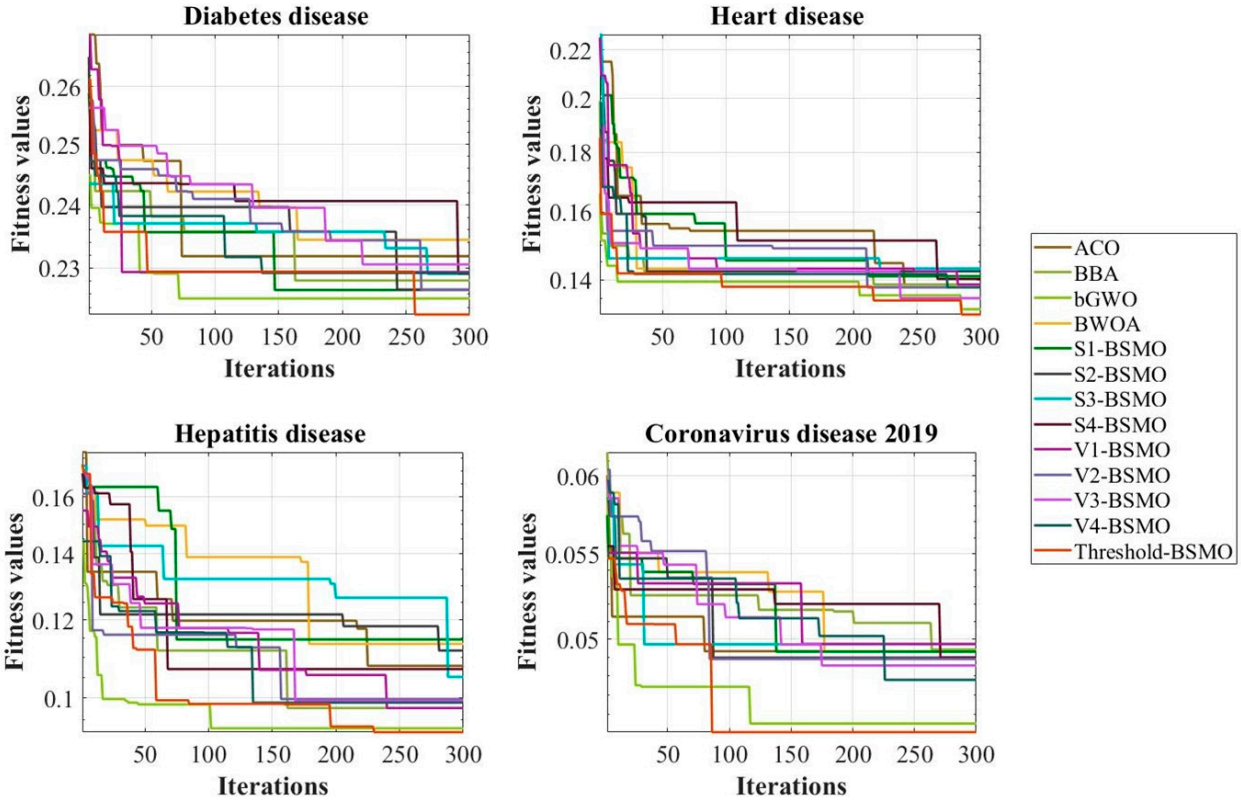

Figure 4. Convergence comparison of the BSMO and comparative algorithms.

5.5. Statistical Analysis

To compare the algorithms fairly and to choose the best transfer function for mapping the continuous solutions to binary ones, Friedman's statistical test was used to rank the algorithms. Table 7 shows the results of Friedman's test according to the fitness values of the algorithms in which the Threshold-BSMO is a great variant to select the effect features from Diabetes, Heart, Hepatic, and Coronavirus diseases.

Table 7. Friedman test.

Algorithms	Medical Problems			
	Diabetes (Rank)	Heart (Rank)	Hepatics (Rank)	COVID-19 (Rank)
ACO	10.37(11)	8.67 (8)	9.23 (8)	9.70 (11)
BBA	10.37 (11)	8.67 (8)	9.23 (8)	9.70 (11)
bGWO	2.80 (2)	2.17 (2)	3.07 (2)	2.27 (2)
BWOA	10.40 (12)	11.23 (12)	9.23 (8)	8.70 (9)
S1-BSMO	5.53 (4)	8.57 (7)	11.87 (11)	7.80 (7)
S2-BSMO	7.43 (8)	9.30 (9)	8.87 (7)	8.97 (10)
S3-BSMO	9.47 (10)	10.73 (11)	9.27 (9)	10.07 (12)
S4-BSMO	9.27 (9)	10.37 (10)	9.43 (10)	8.40 (8)
V1-BSMO	6.13 (5)	5.67 (6)	4.87 (6)	5.53 (3)
V2-BSMO	6.27 (6)	5.13 (5)	4.40 (4)	6.73 (6)
V3-BSMO	6.70 (7)	3.90 (3)	4.60 (5)	6.20 (5)
V4-BSMO	4.67 (3)	4.80 (4)	4.20 (3)	5.60 (4)
Threshold-BSMO	1.60 (1)	1.80 (1)	2.73 (1)	1.33 (1)

6. Conclusions

Many metaheuristic algorithms have been applied in the wrapper-based methods to select effective features from medical data; however, most cannot find those features that can fulfill an acceptable accurate diagnosis of diseases. To deal with this weakness, a new binary metaheuristic algorithm named binary starling murmuration optimization (BSMO) is proposed to select the effective features from different important diseases such as Diabetes, Heart, Hepatitis, and Coronavirus. The proposed BSMO used two different approaches: S-shaped and V-shaped transfer functions and a variable threshold method to convert the continuous solutions to binary ones. Moreover, metrics such as fitness, accuracy, sensitivity, specificity, precision, and error were used to assess the proposed BSMO's performance compared to competing algorithms. Finally, the Friedman non-parametric test was also used to show the proposed algorithm's superiority statistically. The statistical and experimental tests proved that the proposed BSMO algorithm is very competitive in selecting effective features from targeted medical datasets. The proposed Threshold-BSMO can effectively find the optimal feature subset for Diabetes, Heart, Hepatitis, and Coronavirus diseases. Overall, considering the fitness criterion as the main criterion for identifying the most effective binary algorithm in selecting the effective features from the medical datasets targeted in this study, Threshold-BSMO was a superior variant to the contender algorithms.

Although the proposed algorithm can select effective features compared to other comparative algorithms, it was limited to four disease datasets targeted in this study. Therefore, the proposed BSMO algorithm can be applied and improved for other real-world applications. Moreover, a self-adapting parameter tuning method can be applied instead of the try-and-test method used for tuning some parameters of BSMO. The BSMO can be armed by other binarization techniques and transfer functions for selecting effective features in other applications. In addition, the SMO's search strategies can be hybridized with other metaheuristic algorithms to generate better candidate continues solutions.

Author Contributions: Conceptualization, M.H.N.-S., Z.A.V. and H.Z.; methodology, M.H.N.-S., Z.A.V. and H.Z.; software, M.H.N.-S., Z.A.V. and H.Z.; validation, M.H.N.-S., Z.A.V., H.Z. and S.M.; formal analysis, M.H.N.-S., Z.A.V., H.Z. and S.M.; investigation, M.H.N.-S., Z.A.V. and H.Z.; resources, M.H.N.-S. and H.Z.; data curation, M.H.N.-S., Z.A.V., H.Z. and S.M.; writing, original draft preparation, M.H.N.-S., Z.A.V. and H.Z.; writing—review and editing, M.H.N.-S., Z.A.V., H.Z. and S.M.; visualization, M.H.N.-S., Z.A.V. 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

The metaheuristic optimization algorithms' performance is strongly dependent on selecting the proper values for their parameters. Therefore, in this section, the sensitivity on different values for key parameters of the BSMO algorithm, such as flock size (k) and population size (N), are analyzed and tuned using the offline parameter tuning method. The detailed results of pretests and experiments for tuning the BMSO's parameter values to find its best robustness in solving feature selection problems on targeted medical datasets were reported in Tables A1–A6 in terms of fitness, error, accuracy, sensitivity, specificity, and precision. The Friedman rank in Tables A1 and A2 specifies the highest performance of BSMO when k and N are equal to 5 and 30, respectively.

Table A1. Parameters setting of BSMO algorithm in terms of fitness values.

Algorithms	**Metrics**	***k* = 3, *N* = 30**				***k* = 5, *N* = 20**			
		Diabetes	**Heart**	**Hepatics**	**COVID-19**	**Diabetes**	**Heart**	**Hepatics**	**COVID-19**
S1-BSMO	Avg	0.2349	0.1463	0.1235	0.0519	0.2354	0.1482	0.1238	0.0523
	Min	0.2295	0.1382	0.1129	0.0493	0.2305	0.1418	0.1014	0.0490
S2-BSMO	Avg	0.2358	0.1486	0.1222	0.0517	0.2368	0.1492	0.1230	0.0522
	Min	0.2317	0.1411	0.1067	0.0497	0.2292	0.1403	0.1080	0.0505
S3-BSMO	Avg	0.2370	0.1487	0.1196	0.0516	0.2367	0.1506	0.1242	0.0519
	Min	0.2331	0.1403	0.1069	0.0493	0.2241	0.1403	0.1138	0.0485
S4-BSMO	Avg	0.2360	0.1497	0.1225	0.0519	0.2369	0.1517	0.1234	0.0521
	Min	0.2305	0.1432	0.1118	0.0505	0.2319	0.1403	0.1050	0.0505
V1-BSMO	Avg	0.2338	0.1419	0.1103	0.0513	0.2361	0.1418	0.1109	0.0519
	Min	0.2305	0.1358	0.0990	0.0479	0.2319	0.1380	0.0994	0.0510
V2-BSMO	Avg	0.2335	0.1413	0.1096	0.0515	0.2365	0.1428	0.1108	0.0510
	Min	0.2319	0.1380	0.0995	0.0493	0.2345	0.1387	0.1059	0.0497
V3-BSMO	Avg	0.2341	0.1410	0.1091	0.0507	0.2347	0.1423	0.1103	0.0508
	Min	0.2319	0.1351	0.0981	0.0497	0.2305	0.1395	0.1003	0.0475
V4-BSMO	Avg	0.2330	0.1410	0.1092	0.0505	0.2344	0.1422	0.1101	0.0514
	Min	0.2240	0.1380	0.0990	0.0482	0.2319	0.1380	0.0999	0.0486
Threshold-BSMO	Avg	0.2314	0.1375	0.1044	0.0487	0.2324	0.1395	0.1144	0.0497
	Min	0.2268	0.1308	0.0884	0.0463	0.2254	0.1337	0.0978	0.0482
Friedman rank		2				4			
Algorithms	Metrics	*k* = 5, *N* = 30				*k* = 7, *N* = 30			
		Diabetes	Heart	Hepatics	COVID-19	Diabetes	Heart	Hepatics	COVID-19
S1-BSMO	Avg	0.2342	0.1460	0.1265	0.0515	0.2352	0.1463	0.1230	0.0517
	Min	0.2266	0.1411	0.1147	0.0493	0.2330	0.1374	0.1146	0.0497
S2-BSMO	Avg	0.2352	0.1481	0.1218	0.0516	0.2344	0.1462	0.1208	0.0518
	Min	0.2267	0.1424	0.1118	0.0490	0.2293	0.1403	0.1088	0.0509
S3-BSMO	Avg	0.2373	0.1495	0.1213	0.0517	0.2360	0.1484	0.1211	0.0517
	Min	0.2291	0.1432	0.1051	0.0497	0.2331	0.1432	0.1128	0.0505
S4-BSMO	Avg	0.2368	0.1492	0.1209	0.0516	0.2367	0.1497	0.1238	0.0518
	Min	0.2291	0.1403	0.1070	0.0490	0.2331	0.1440	0.1120	0.0501
V1-BSMO	Avg	0.2344	0.1423	0.1109	0.0510	0.2343	0.1427	0.1096	0.0509
	Min	0.2294	0.1387	0.0977	0.0497	0.2293	0.1411	0.0990	0.0489
V2-BSMO	Avg	0.2343	0.1417	0.1106	0.0509	0.2339	0.1410	0.1098	0.0508
	Min	0.2266	0.1380	0.0998	0.0489	0.2294	0.1387	0.1046	0.0497
V3-BSMO	Avg	0.2353	0.1411	0.1107	0.0510	0.2354	0.1413	0.1125	0.0515
	Min	0.2306	0.1351	0.0994	0.0486	0.2320	0.1380	0.1073	0.0496
V4-BSMO	Avg	0.2335	0.1414	0.1096	0.0506	0.2330	0.1425	0.1100	0.0507
	Min	0.2292	0.1380	0.0990	0.0478	0.2293	0.1403	0.1049	0.0490
Threshold-BSMO	Avg	0.2306	0.1378	0.1081	0.0488	0.2302	0.1370	0.0920	0.0491
	Min	0.2229	0.1308	0.0924	0.0451	0.2266	0.1308	0.0920	0.0478
Friedman rank		1				3			

Table A2. Parameters setting of BSMO algorithm in terms of error values.

Algorithms	**Metrics**	***k* = 3, *N* = 30**				***k* = 5, *N* = 20**			
		Diabetes	**Heart**	**Hepatics**	**COVID-19**	**Diabetes**	**Heart**	**Hepatics**	**COVID-19**
S1-BSMO	Avg	0.2505	0.1954	0.1594	0.0523	0.2541	0.2081	0.1692	0.0538
	Min	0.2422	0.1556	0.1212	0.0498	0.2383	0.1593	0.1412	0.0498
S2-BSMO	Avg	0.2492	0.1949	0.1628	0.0518	0.2556	0.2096	0.1693	0.0531
	Min	0.2370	0.1593	0.1358	0.0475	0.2422	0.1704	0.1342	0.0487
S3-BSMO	Avg	0.2517	0.1956	0.1624	0.0523	0.2541	0.2057	0.1672	0.0535
	Min	0.2408	0.1519	0.1429	0.0476	0.2369	0.1519	0.1421	0.0498
S4-BSMO	Avg	0.2498	0.2016	0.1573	0.0529	0.2546	0.2037	0.1664	0.0536
	Min	0.2371	0.1593	0.1225	0.0487	0.2383	0.1667	0.1358	0.0498
V1-BSMO	Avg	0.2551	0.1930	0.1549	0.0530	0.2589	0.2004	0.1623	0.0527
	Min	0.2461	0.1519	0.1162	0.0486	0.2488	0.1519	0.1346	0.0510
V2-BSMO	Avg	0.2578	0.1946	0.1532	0.0528	0.2560	0.2044	0.1611	0.0542
	Min	0.2461	0.1630	0.1096	0.0464	0.2396	0.1482	0.1300	0.0510
V3-BSMO	Avg	0.2504	0.1906	0.1588	0.0536	0.2590	0.2047	0.1669	0.0546
	Min	0.2357	0.1519	0.1171	0.0486	0.2474	0.1704	0.1479	0.0509
V4-BSMO	Avg	0.2557	0.1878	0.1545	0.0528	0.2554	0.2049	0.1635	0.0538
	Min	0.2474	0.1630	0.1279	0.0487	0.2422	0.1556	0.1417	0.0509
Threshold-BSMO	Avg	0.2514	0.1925	0.1640	0.0525	0.2563	0.2163	0.1629	0.0534
	Min	0.2370	0.1481	0.1233	0.0487	0.2448	0.1593	0.1150	0.0497
Friedman rank		2				4			
Algorithms	Metrics	*k* = 5, *N* = 30				*k* = 7, *N* = 30			
		Diabetes	Heart	Hepatics	COVID-19	Diabetes	Heart	Hepatics	COVID-19
S1-BSMO	Avg	0.2504	0.1964	0.1659	0.0511	0.2506	0.2020	0.1612	0.0521
	Min	0.2382	0.1593	0.1292	0.0452	0.2409	0.1630	0.1363	0.0463
S2-BSMO	Avg	0.2516	0.2015	0.1598	0.0520	0.2534	0.1983	0.1702	0.0520
	Min	0.2369	0.1556	0.1171	0.0498	0.2423	0.1593	0.1500	0.0475
S3-BSMO	Avg	0.2508	0.1930	0.1599	0.0521	0.2532	0.1847	0.1598	0.0521
	Min	0.2397	0.1556	0.1237	0.0487	0.2421	0.1630	0.1288	0.0464
S4-BSMO	Avg	0.2533	0.1907	0.1603	0.0532	0.2568	0.1901	0.1650	0.0532
	Min	0.2384	0.1481	0.1296	0.0498	0.2487	0.1519	0.1429	0.0487
V1-BSMO	Avg	0.2552	0.1884	0.1587	0.0537	0.2537	0.1915	0.1572	0.0533
	Min	0.2422	0.1593	0.1292	0.0476	0.2384	0.1593	0.1346	0.0498
V2-BSMO	Avg	0.2548	0.1911	0.1589	0.0530	0.2539	0.1896	0.1637	0.0527
	Min	0.2345	0.1481	0.1342	0.0474	0.2447	0.1630	0.1483	0.0521
V3-BSMO	Avg	0.2547	0.1956	0.1617	0.0530	0.2558	0.2096	0.1531	0.0528
	Min	0.2383	0.1667	0.1412	0.0475	0.2475	0.1889	0.1225	0.0464
V4-BSMO	Avg	0.2534	0.1959	0.1617	0.0532	0.2565	0.1959	0.1472	0.0538
	Min	0.2383	0.1519	0.1425	0.0452	0.2396	0.1556	0.1163	0.0521
Threshold-BSMO	Avg	0.2530	0.1952	0.1623	0.0518	0.2498	0.1986	0.1558	0.0528
	Min	0.2408	0.1519	0.1342	0.0487	0.2395	0.1593	0.1558	0.0487
Friedman rank		1				3			

Table A3. Parameters setting of BSMO algorithm in terms of accuracy values.

Algorithms	**Metrics**	***k* = 3, *N* = 30**				***k* = 5, *N* = 20**			
		Diabetes	**Heart**	**Hepatics**	**COVID-19**	**Diabetes**	**Heart**	**Hepatics**	**COVID-19**
S1-BSMO	Avg	76.9228	85.7901	88.1236	95.4467	76.8885	85.5802	88.1083	95.401
	Max	77.3257	86.6667	89.125	95.5961	77.4761	86.2963	90.2917	95.707
S2-BSMO	Avg	76.8251	85.4568	88.1625	95.364	76.717	85.4444	88.0736	95.3321
	Max	77.3582	86.2963	89.75	95.5921	77.4761	86.2963	89.625	95.4852
S3-BSMO	Avg	76.6528	85.4198	88.3667	95.3389	76.677	85.2222	87.9167	95.2899
	Max	77.0899	86.2963	89.6667	95.4852	77.9973	86.2963	89.0417	95.4878
S4-BSMO	Avg	76.7293	85.3086	88.0444	95.3477	76.6684	85.0988	87.9708	95.3
	Max	77.2198	85.9259	89.125	95.4878	77.0933	86.2963	89.875	95.4838
V1-BSMO	Avg	76.9143	86	89.2708	95.2115	76.707	86	89.158	95.197
	Max	77.218	86.6667	90.375	95.4758	77.088	86.296	90.333	95.362
V2-BSMO	Avg	76.9203	86.0617	89.2958	95.2158	76.644	85.926	89.229	95.238
	Max	77.0779	86.2963	90.375	95.3635	76.822	86.296	89.833	95.364
V3-BSMO	Avg	76.9173	86.0741	89.3458	95.2694	76.834	85.951	89.211	95.268
	Max	77.2095	86.6667	90.4583	95.3729	77.227	86.296	90.292	95.595
V4-BSMO	Avg	76.9872	86.0864	89.3708	95.2895	76.876	85.951	89.289	95.204
	Max	78.0041	86.2963	90.375	95.83	77.081	86.296	90.333	95.481
Threshold-BSMO	Avg	77.3771	86.5309	89.8972	95.5124	77.136	86.37	88.919	95.467
	Max	78.1203	87.4074	91.5	95.715	77.862	87.037	90.542	95.599
Algorithms	Metrics	*k* = 5, *N* = 30				*k* = 7, *N* = 30			
		Diabetes	Heart	Hepatics	COVID-19	Diabetes	Heart	Hepatics	COVID-19
S1-BSMO	Avg	76.9719	85.8148	87.8319	95.417	76.895	85.778	88.189	95.427
	Max	77.7409	86.2963	89	95.5988	77.216	86.667	88.958	95.599
S2-BSMO	Avg	76.8537	85.5185	88.1708	95.3861	76.927	85.704	88.3	95.389
	Max	77.7341	85.9259	89.1667	95.5961	77.344	86.296	89.542	95.482
S3-BSMO	Avg	76.6101	85.3333	88.2153	95.3308	76.763	85.457	88.192	95.336
	Max	77.4897	85.9259	89.9167	95.6001	76.965	85.926	89.083	95.482
S4-BSMO	Avg	76.6654	85.3704	88.2306	95.3347	76.643	85.333	87.922	95.341
	Max	77.4863	86.2963	89.6667	95.5948	76.96	85.926	89.167	95.484
V1-BSMO	Avg	76.889	85.9383	89.1542	95.2469	76.864	85.963	89.313	95.24
	Max	77.3411	86.2963	90.5	95.5974	77.348	86.296	90.375	95.376
V2-BSMO	Avg	76.8872	86.037	89.2069	95.263	76.918	86.148	89.267	95.227
	Max	77.6128	86.2963	90.375	95.4812	77.353	86.296	89.792	95.365
V3-BSMO	Avg	76.7716	86.0741	89.1986	95.2695	76.774	86.037	89.013	95.23
	Max	77.2163	86.6667	90.375	95.4838	77.075	86.296	89.583	95.607
V4-BSMO	Avg	76.9639	86.0123	89.3278	95.2692	76.992	85.926	89.292	95.293
	Max	77.471	86.2963	90.375	95.4892	77.346	86.296	89.833	95.365
Threshold-BSMO	Avg	77.3077	86.5309	89.5194	95.537	77.328	86.6173	91.0833	95.4834
	Max	77.9904	87.4074	91.0417	95.8353	77.6179	87.4074	91.0833	95.7124

Table A4. Parameters setting of BSMO algorithm in terms of sensitivity values.

Algorithms	**Metrics**	**k = 3, N = 30**				**k = 5, N = 20**			
		Diabetes	**Heart**	**Hepatics**	**COVID-19**	**Diabetes**	**Heart**	**Hepatics**	**COVID-19**
S1-BSMO	Avg	88.7609	93.8971	68.6377	99.2675	88.365	93.4924	68.8162	99.2336
	Max	90.2278	96.2467	85.3876	99.7684	89.7842	95.292	78.4811	100
S2-BSMO	Avg	88.3668	93.6035	71.8856	99.3568	88.22	92.9212	69.6659	99.3498
	Max	89.1665	95.3642	84.1441	99.8295	89.7151	95.2571	79.7631	100
S3-BSMO	Avg	88.1191	93.0296	72.7952	99.3912	88.0791	93.0515	69.5805	99.383
	Max	89.0592	95.2235	85.8833	100	89.7059	95.8368	79.1405	100
S4-BSMO	Avg	88.191	93.1775	72.6372	99.4499	88.0248	93.0834	71.6037	99.2982
	Max	89.0733	95.3049	81.5983	100	89.4038	96.6535	88.5142	99.7212
V1-BSMO	Avg	88.3628	94.2365	78.835	99.9088	88.327	94.012	76.679	99.504
	Max	89.4449	95.5444	88.3507	100	89.715	94.952	85.144	100
V2-BSMO	Avg	87.6423	94.536	79.637	99.9252	87.515	93.419	80.879	99.696
	Max	88.1193	96.3431	90.6895	100	88.523	94.47	88.078	100
V3-BSMO	Avg	87.8565	94.0807	79.3312	99.8619	88.05	94.206	75.106	99.811
	Max	88.9304	96.2699	86.7033	100	89.359	95.947	83.572	100
V4-BSMO	Avg	88.4581	94.1059	80.3357	99.8371	88.104	93.948	78.811	99.775
	Max	89.115	95.4339	89.9166	100	89.716	94.769	91.569	100
Threshold-BSMO	Avg	89.0468	94.655	80.116	99.3936	88.503	94.575	75.676	99.348
	Max	91.0607	96.4165	88.838	100	89.44	96.612	84.286	99.548
Algorithms	Metrics	k = 5, N = 30				k = 7, N = 30			
		Diabetes	Heart	Hepatics	COVID-19	Diabetes	Heart	Hepatics	COVID-19
S1-BSMO	Avg	88.5142	93.6608	70.6404	99.2906	88.514	93.831	71.703	99.26
	Max	89.8454	95.0876	80.2298	99.7496	89.311	95.569	88.021	99.609
S2-BSMO	Avg	88.2422	93.3517	70.8924	99.3947	88.387	93.475	67.965	99.381
	Max	89.2631	95.3351	84.532	100	89.719	94.748	75.613	99.822
S3-BSMO	Avg	88.1787	93.1475	71.8705	99.3703	88.325	92.863	69.775	99.512
	Max	90.1796	94.4033	85.8738	100	90.142	95.51	75.39	100
S4-BSMO	Avg	88.3085	93.2132	72.851	99.4093	88.234	93.467	74.89	99.486
	Max	89.7088	95.0297	82.1369	100	88.447	96.508	90.417	100
V1-BSMO	Avg	88.2848	93.8571	78.8832	99.8598	88.244	94.334	78.981	99.861
	Max	89.7132	96.2621	85.8624	100	88.978	96.082	87.181	100
V2-BSMO	Avg	88.6261	94.4918	79.3521	99.8182	88.217	94.237	76.884	99.87
	Max	90.0085	96.2525	87.3972	100	89.536	96.225	85.649	100
V3-BSMO	Avg	88.245	94.1042	78.5909	99.7693	88.328	94.287	78.435	99.707
	Max	89.6911	95.6443	86.1964	100	89.598	96.606	85.285	100
V4-BSMO	Avg	88.1009	94.0096	79.7051	99.7845	88.377	94.327	77.526	99.802
	Max	89.484	95.4345	88.4275	100	89.576	95.545	84.209	100
Threshold-BSMO	Avg	89	94.6804	80.2438	99.3774	89.1512	94.8397	81.5161	99.4531
	Max	89.9871	96.6626	91.3715	100	90.8328	96.8864	81.5161	100

Table A5. Parameters setting of BSMO algorithm in terms of precision values.

Algorithms	**Metrics**	***k* = 3, *N* = 30**				***k* = 5, *N* = 20**			
		Diabetes	**Heart**	**Hepatics**	**COVID-19**	**Diabetes**	**Heart**	**Hepatics**	**COVID-19**
S1-BSMO	Avg	83.3024	89.7759	81.8452	97.6496	82.9358	88.9528	77.9683	97.6013
	Max	84.5112	91.1126	92.5556	98.0808	84.0629	91.3534	92.5714	97.9844
S2-BSMO	Avg	83.1656	89.6198	83.3028	97.5525	82.995	88.9851	77.8085	97.5018
	Max	83.9979	91.6603	96.8889	97.9414	84.226	92.1922	96.6667	98.2263
S3-BSMO	Avg	82.8792	89.3916	80.6788	97.6031	82.7702	89.4225	78.8535	97.4893
	Max	84.0439	92.1316	94	97.9846	84.4498	92.9411	96.6667	98.1287
S4-BSMO	Avg	82.9226	90.0263	82.091	97.5329	82.8662	89.0141	78.5044	97.5208
	Max	83.5297	93.4569	98.3333	98.0334	84.1876	91.7804	92.4643	98.099
V1-BSMO	Avg	83.3034	89.275	82.3873	97.3381	82.916	89.014	82.316	97.303
	Max	84.5946	91.9784	89.9596	97.6875	83.93	90.58	88.611	97.645
V2-BSMO	Avg	82.8698	89.5967	85.3515	97.3886	82.777	90.147	78.368	97.291
	Max	83.2788	91.9132	100	97.8924	83.495	92.77	92.051	97.604
V3-BSMO	Avg	83.0581	89.6372	84.2314	97.4291	82.752	89.063	80.449	97.369
	Max	84.4839	92.3453	94.6429	98.2173	84.24	92.867	96.349	97.735
V4-BSMO	Avg	83.0346	90.0785	84.4764	97.4154	83.013	89.097	81.099	97.38
	Max	83.6545	92.453	100	98.0199	83.788	91.567	93.099	97.932
Threshold-BSMO	Avg	83.7292	91.6759	85.2687	97.7006	83.564	91.301	81.078	97.67
	Max	85.4714	93.7512	97.5	98.3656	84.936	92.869	95.325	98.032
Algorithms	Metrics	*k* = 5, *N* =30				*k* = 7, *N* =30			
		Diabetes	Heart	Hepatics	COVID-19	Diabetes	Heart	Hepatics	COVID-19
S1-BSMO	Avg	83.2288	89.2841	81.3914	97.7266	83.487	89.175	84.47	97.634
	Max	84.242	91.4817	95.0256	98.2616	84.571	89.756	96.5	98.064
S2-BSMO	Avg	83.1426	89.403	78.9332	97.5923	83.224	89.761	81.574	97.574
	Max	84.2726	91.5718	91.5289	97.9672	84.225	92.71	98	97.908
S3-BSMO	Avg	82.925	89.7136	81.3385	97.5576	82.855	89.229	78.523	97.55
	Max	84.4974	91.6581	96.9048	98.2389	84.549	91.133	86.998	97.873
S4-BSMO	Avg	82.764	89.4763	81.9163	97.5954	83.236	89.538	82.034	97.459
	Max	83.8476	91.4379	93.1111	98.126	84.582	90.938	98.75	97.872
V1-BSMO	Avg	83.1764	89.3417	83.8414	97.3384	82.958	89.327	83.84	97.356
	Max	86.1848	91.9558	95.5556	97.924	84.323	90.731	91	97.672
V2-BSMO	Avg	82.9072	89.3503	84.3151	97.364	82.987	89.725	82.101	97.369
	Max	83.761	91.9198	96.3492	97.8237	84.159	92.147	92.372	97.734
V3-BSMO	Avg	83.1812	89.6908	85.7139	97.3319	83.21	89.99	81.337	97.334
	Max	84.5091	91.8579	97.5	97.9259	84.092	91.817	86.738	97.62
V4-BSMO	Avg	83.2658	89.7266	84.2503	97.4058	83.289	89.8	83.036	97.41
	Max	84.4214	91.4855	98.75	97.956	84.474	92.318	95.238	97.716
Threshold-BSMO	Avg	83.5823	91.4408	85.1981	97.7178	83.5777	91.6578	82.5912	97.6967
	Max	84.7376	93.8631	95.7778	98.0502	84.5587	94.5662	82.5912	97.9926

Table A6. Parameters setting of BSMO algorithm in terms of specificity values.

Algorithms	Metrics	k = 3, N = 30				k = 5, N = 20			
		Diabetes	**Heart**	**Hepatics**	**COVID-19**	**Diabetes**	**Heart**	**Hepatics**	**COVID-19**
S1-BSMO	Avg	66.2342	86.542	99.4587	82.4862	65.1298	85.5329	99.2915	82.346
	Max	68.2502	87.9644	100	85.9366	66.4093	88.7883	99.8374	84.4318
S2-BSMO	Avg	65.4575	86.9073	99.5053	81.7373	65.6179	85.9113	99.3399	81.8105
	Max	66.6288	89.656	100	85.0243	67.707	91.2161	99.9187	85.9674
S3-BSMO	Avg	64.8413	86.4002	99.4766	82.3377	65.1382	86.3003	99.2782	81.4987
	Max	66.1969	89.2235	100	84.8969	67.4399	91.9747	100	86.2247
S4-BSMO	Avg	65.7836	87.4361	99.4948	81.9079	65.0415	86.3582	99.4617	81.5022
	Max	66.5935	91.8474	99.9187	84.9453	67.8061	90.3087	100	86.1819
V1-BSMO	Avg	65.75	88.5427	99.368	80.2567	65.413	88.039	99.063	80.501
	Max	67.4069	90.7376	100	83.9446	66.648	89.375	99.399	82.194
V2-BSMO	Avg	66.6065	89.1717	99.4428	80.7396	64.975	89.529	99.286	79.295
	Max	68.4934	91.1563	100	84.5923	66.435	92.307	100	82.191
V3-BSMO	Avg	66.1459	88.7473	99.3145	81.0216	64.927	88.066	99.258	80.392
	Max	67.5985	91.9959	99.8286	85.1101	66.815	90.58	99.829	82.735
V4-BSMO	Avg	65.8974	88.7773	99.4461	81.007	65.451	88.153	99.243	80.932
	Max	67.0376	90.7347	100	85.0112	66.87	91.094	99.837	83.784
Threshold-BSMO	Avg	66.9048	89.1478	99.3309	82.7429	66.572	88.889	99.058	81.727
	Max	69.2527	92.084	100	87.2612	70.241	90.63	100	84.678
Algorithms	Metrics	k = 5, N = 30				k = 7, N = 30			
		Diabetes	Heart	Hepatics	COVID-19	Diabetes	Heart	Hepatics	COVID-19
S1-BSMO	Avg	65.9602	86.0433	99.422	83.0173	66.509	85.883	99.513	82.554
	Max	68.2916	88.7512	100	87.3208	69.423	86.761	100	84.83
S2-BSMO	Avg	65.7932	86.2794	99.4674	82.1743	66.123	87.242	99.451	82.462
	Max	68.023	88.2128	100	85.5629	68.293	90.414	100	84.963
S3-BSMO	Avg	65.5806	87.0123	99.377	81.7173	66.103	86.476	99.369	81.929
	Max	67.8662	89.3531	100	87.298	69.399	88.543	99.837	84.695
S4-BSMO	Avg	65.0295	86.3764	99.3568	82.1407	65.939	86.436	99.48	81.524
	Max	66.8104	89.284	100	86.2074	66.92	88.336	100	85.479
V1-BSMO	Avg	65.7345	89.0497	99.471	80.376	65.889	88.631	99.324	80.379
	Max	70.5787	91.2747	100	84.761	68.174	89.595	99.919	82.705
V2-BSMO	Avg	65.7846	88.4579	99.2964	80.6954	65.45	89.022	99.279	80.59
	Max	67.6503	91.1828	99.9187	84.0749	66.884	90.656	99.919	82.872
V3-BSMO	Avg	66.0204	88.5817	99.4433	80.477	65.038	88.873	99.174	80.363
	Max	69.1626	90.503	100	83.9283	67.188	90.793	99.473	83.175
V4-BSMO	Avg	66.2564	88.5579	99.4127	80.7991	66.366	88.717	99.438	81.74
	Max	69.1896	91.1526	100	84.4487	67.647	91.444	99.666	83.986
Threshold-BSMO	Avg	66.6321	88.8911	99.4531	83.1011	66.562	89.1274	99.6652	82.6134
	Max	69.2028	92.1136	100	87.2075	68.5097	92.9485	99.6652	85.0483

References

1. Nilashi, M.; bin Ibrahim, O.; Ahmadi, H.; Shahmoradi, L. An analytical method for diseases prediction using machine learning techniques. *Comput. Chem. Eng.* **2017**, *106*, 212–223. [CrossRef]
2. Abdulkhaleq, M.T.; Rashid, T.A.; Alsadoon, A.; Hassan, B.A.; Mohammadi, M.; Abdullah, J.M.; Chhabra, A.; Ali, S.L.; Othman, R.N.; Hasan, H.A.; et al. Harmony search: Current studies and uses on healthcare systems. *Artif. Intell. Med.* **2022**, *131*, 102348. [CrossRef]
3. Qader, S.M.; Hassan, B.A.; Rashid, T.A. An improved deep convolutional neural network by using hybrid optimization algorithms to detect and classify brain tumor using augmented MRI images. *Multimedia Tools Appl.* **2022**, *81*, 44059–44086. [CrossRef]

4. Golestan Hashemi, F.S.; Razi Ismail, M.; Rafii Yusop, M.; Golestan Hashemi, M.S.; Nadimi Shahraki, M.H.; Rastegari, H.; Miah, G.; Aslani, F. Intelligent mining of large-scale bio-data: Bioinformatics applications. *Biotechnol. Biotec. Eq.* **2018**, *32*, 10–29. [CrossRef]
5. Van Woensel, W.; Elnenaei, M.; Abidi, S.S.R.; Clarke, D.B.; Imran, S.A. Staged reflexive artificial intelligence driven testing algorithms for early diagnosis of pituitary disorders. *Clin. Biochem.* **2021**, *97*, 48–53. [CrossRef] [PubMed]
6. Shah, D.; Patel, S.; Bharti, S.K. Heart Disease Prediction using Machine Learning Techniques. *SN Comput. Sci.* **2020**, *1*, 1–6. [CrossRef]
7. Sharma, P.; Choudhary, K.; Gupta, K.; Chawla, R.; Gupta, D.; Sharma, A. Artificial plant optimization algorithm to detect heart rate & presence of heart disease using machine learning. *Artif. Intell. Med.* **2019**, *102*, 101752. [CrossRef]
8. Devaraj, J.; Elavarasan, R.M.; Pugazhendhi, R.; Shafiullah, G.; Ganesan, S.; Jeysree, A.K.; Khan, I.A.; Hossain, E. Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant? *Results Phys.* **2021**, *21*, 103817. [CrossRef]
9. Remeseiro, B.; Bolon-Canedo, V. A review of feature selection methods in medical applications. *Comput. Biol. Med.* **2019**, *112*, 103375. [CrossRef]
10. Gokulnath, C.B.; Shantharajah, S.P. An optimized feature selection based on genetic approach and support vector machine for heart disease. *Clust. Comput.* **2018**, *22*, 14777–14787. [CrossRef]
11. Huda, S.; Yearwood, J.; Jelinek, H.F.; Hassan, M.M.; Fortino, G.; Buckland, M. A Hybrid Feature Selection With Ensemble Classification for Imbalanced Healthcare Data: A Case Study for Brain Tumor Diagnosis. *IEEE Access* **2016**, *4*, 9145–9154. [CrossRef]
12. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A survey on evolutionary computation approaches to feature selection. *IEEE Trans. Evol. Comput.* **2015**, *20*, 606–626. [CrossRef]
13. García-Torres, M.; Gómez-Vela, F.; Melián-Batista, B.; Moreno-Vega, J.M. High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach. *Inf. Sci.* **2016**, *326*, 102–118. [CrossRef]
14. Yu, L.; Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 856–863.
15. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
16. Gnana, D.A.A.; Balamurugan, S.A.A.; Leavline, E.J. Literature review on feature selection methods for high-dimensional data. *Int. J. Comput. Appl.* **2016**, *136*, 9–17.
17. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [CrossRef]
18. Karegowda, A.G.; Manjunath, A.S.; Jayaram, M.A. Feature Subset Selection Problem using Wrapper Approach in Supervised Learning. *Int. J. Comput. Appl.* **2010**, *1*, 13–17. [CrossRef]
19. Kabir, M.; Shahjahan; Murase, K. A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing* **2011**, *74*, 2914–2928. [CrossRef]
20. 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, Boston, MA, USA, 13–17 July 2019; pp. 481–489. [CrossRef]
21. 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]
22. Benyamin, A.; Farhad, S.G.; Saeid, B. Discrete farmland fertility optimization algorithm with metropolis acceptance criterion for traveling salesman problems. *Int. J. Intell. Syst.* **2020**, *36*, 1270–1303. [CrossRef]
23. Fard, E.S.; Monfaredi, K.; Nadimi, M.H. An Area-Optimized Chip of Ant Colony Algorithm Design in Hardware Platform Using the Address-Based Method. *Int. J. Electr. Comput. Eng. (IJECE)* **2014**, *4*, 989–998. [CrossRef]
24. Sayadi, M.K.; Hafezalkotob, A.; Naini, S.G.J. Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation. *J. Manuf. Syst.* **2013**, *32*, 78–84. [CrossRef]
25. Gharehchopogh, F.S.; Nadimi-Shahraki, M.H.; Barshandeh, S.; Abdollahzadeh, B.; Zamani, H. CQFFA: A Chaotic Quasi-oppositional Farmland Fertility Algorithm for Solving Engineering Optimization Problems. *J. Bionic Eng.* **2022**, 1–26. [CrossRef]
26. 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]
27. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the MHS'95 Proceedings of the sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
28. 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.
29. Storn, R.; Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Global. Optim.* **1997**, *11*, 341–359. [CrossRef]
30. Rajabioun, R. Cuckoo optimization algorithm. *Appl. Soft. Comput.* **2011**, *11*, 5508–5518. [CrossRef]
31. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [CrossRef]
32. James, J.; Li, V.O. A social spider algorithm for global optimization. *Appl. Soft. Comput.* **2015**, *30*, 614–627. [CrossRef]
33. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [CrossRef]

34. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
35. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. QANA: Quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104314. [CrossRef]
36. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [CrossRef]
37. Abu Khurmaa, R.; Aljarah, I.; Sharieh, A. An intelligent feature selection approach based on moth flame optimization for medical diagnosis. *Neural Comput. Appl.* **2020**, *33*, 7165–7204. [CrossRef]
38. Moorthy, U.; Gandhi, U.D. A novel optimal feature selection technique for medical data classification using ANOVA based whale optimization. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 3527–3538. [CrossRef]
39. 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.
40. 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]
41. 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]
42. Mirjalili, S.; Mirjalili, S.M.; Yang, X.-S. Binary bat algorithm. *Neural Comput. Appl.* **2014**, *25*, 663–681. [CrossRef]
43. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [CrossRef]
44. Gandomi, A.H.; Deb, K.; Averill, R.C.; Rahnamayan, S.; Omidvar, M.N. Using semi-independent variables to enhance optimization search. *Expert Syst. Appl.* **2018**, *120*, 279–297. [CrossRef]
45. Kira, K.; Rendell, L.A. A practical approach to feature selection. In *Machine Learning Proceedings*; Elsevier: Amsterdam, The Netherlands, 1992; pp. 249–256.
46. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature selection: A data perspective. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–45. [CrossRef]
47. Bolón-Canedo, V.; Sánchez-Maroño, N.; Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.* **2013**, *34*, 483–519. [CrossRef]
48. Brezočnik, L.; Fister Jr, I.; Podgorelec, V. Swarm intelligence algorithms for feature selection: A review. *Appl. Sci.* **2018**, *8*, 1521. [CrossRef]
49. Solorio-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **2020**, *53*, 907–948. [CrossRef]
50. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
51. Ambusaidi, M.A.; He, X.; Nanda, P.; Tan, Z. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. *IEEE Trans. Comput.* **2016**, *65*, 2986–2998. [CrossRef]
52. Khater, B.; Wahab, A.A.; Idris, M.; Hussain, M.; Ibrahim, A.; Amin, M.; Shehadeh, H. Classifier Performance Evaluation for Lightweight IDS Using Fog Computing in IoT Security. *Electronics* **2021**, *10*, 1633. [CrossRef]
53. 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*, 1–27. [CrossRef]
54. 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]
55. 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]
56. Lin, F.; Liang, D.; Yeh, C.-C.; Huang, J.-C. Novel feature selection methods to financial distress prediction. *Expert Syst. Appl.* **2014**, *41*, 2472–2483. [CrossRef]
57. Kwak, N.; Choi, C.-H. Input feature selection for classification problems. *IEEE Trans. Neural Networks* **2002**, *13*, 143–159. [CrossRef]
58. 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]
59. Xue, B.; Zhang, M.; Browne, W.N. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Trans. Cybern.* **2012**, *43*, 1656–1671. [CrossRef]
60. Akyol, S.; Alatas, B. Plant intelligence based metaheuristic optimization algorithms. *Artif. Intell. Rev.* **2016**, *47*, 417–462. [CrossRef]
61. Alatas, B. Chaotic bee colony algorithms for global numerical optimization. *Expert Syst. Appl.* **2010**, *37*, 5682–5687. [CrossRef]
62. Alatas, B.; Bingol, H. Comparative Assessment Of Light-based Intelligent Search And Optimization Algorithms. *Light Eng.* **2020**, *6*, 51–59. [CrossRef]
63. Mafarja, M.; Eleyan, D.; Abdullah, S.; Mirjalili, S. S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In Proceedings of the International Conference on Future Networks and Distributed Systems, Cambridge, UK, 19–20 July 2017; pp. 1–7.

64. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. BGSA: Binary gravitational search algorithm. *Nat. Comput.* **2009**, *9*, 727–745. [CrossRef]
65. De Souza, R.C.T.; dos Santos Coelho, L.; De Macedo, C.A.; Pierezan, J. A V-shaped binary crow search algorithm for feature selection. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
66. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl.-Based Syst.* **2018**, *161*, 185–204. [CrossRef]
67. 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. [CrossRef]
68. Ahmed, S.; Ghosh, K.K.; Mirjalili, S.; Sarkar, R. AIEOU: Automata-based improved equilibrium optimizer with U-shaped transfer function for feature selection. *Knowl. Based. Syst.* **2021**, *228*, 107283. [CrossRef]
69. Ghosh, K.K.; Singh, P.K.; Hong, J.; Geem, Z.W.; Sarkar, R. Binary social mimic optimization algorithm with X-shaped transfer function for feature selection. *IEEE Access* **2020**, *8*, 97890–97906. [CrossRef]
70. Guo, S.-S.; Wang, J.-S.; Guo, M.-W. Z-Shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–21. [CrossRef]
71. Ramasamy, A.; Mondry, A.; Holmes, C.; Altman, D.G. Key Issues in Conducting a Meta-Analysis of Gene Expression Microarray Datasets. *PLOS Med.* **2008**, *5*, e184. [CrossRef]
72. Alirezanejad, M.; Enayatifar, R.; Motameni, H.; Nematzadeh, H. Heuristic filter feature selection methods for medical datasets. *Genomics* **2019**, *112*, 1173–1181. [CrossRef]
73. Varzaneh, Z.A.; Orooji, A.; Erfannia, L.; Shanbehzadeh, M. A new COVID-19 intubation prediction strategy using an intelligent feature selection and K-NN method. *Informatics Med. Unlocked* **2021**, *28*, 100825. [CrossRef]
74. Pashaei, E.; Pashaei, E. An efficient binary chimp optimization algorithm for feature selection in biomedical data classification. *Neural Comput. Appl.* **2022**, *34*, 6427–6451. [CrossRef]
75. 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. [CrossRef]
76. Alweshah, M.; Alkhalaileh, S.; Al-Betar, M.A.; Abu Bakar, A. Coronavirus herd immunity optimizer with greedy crossover for feature selection in medical diagnosis. *Knowl.-Based Syst.* **2021**, *235*, 107629. [CrossRef] [PubMed]
77. Anter, A.M.; Ali, M. Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. *Soft Comput.* **2019**, *24*, 1565–1584. [CrossRef]
78. Singh, N.; Singh, P. A hybrid ensemble-filter wrapper feature selection approach for medical data classification. *Chemom. Intell. Lab. Syst.* **2021**, *217*, 104396. [CrossRef]
79. Too, J.; Abdullah, A.R. Binary atom search optimisation approaches for feature selection. *Connect. Sci.* **2020**, *32*, 406–430. [CrossRef]
80. Elgamal, Z.; Sabri, A.Q.M.; Tubishat, M.; Tbaishat, D.; Makhadmeh, S.N.; Alomari, O.A. Improved Reptile Search Optimization Algorithm using Chaotic map and Simulated Annealing for Feature Selection in Medical Filed. *IEEE Access* **2022**, *10*, 51428–51446. [CrossRef]
81. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [CrossRef]
82. Zhang, Y.; Song, X.-F.; Gong, D.-W. A return-cost-based binary firefly algorithm for feature selection. *Inf. Sci.* **2017**, *418*, 561–574. [CrossRef]
83. Sayed, G.I.; Tharwat, A.; Hassanien, A.E. Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection. *Appl. Intell.* **2018**, *49*, 188–205. [CrossRef]
84. Wang, J.; Khishe, M.; Kaveh, M.; Mohammadi, H. Binary Chimp Optimization Algorithm (BChOA): A New Binary Meta-heuristic for Solving Optimization Problems. *Cogn. Comput.* **2021**, *13*, 1297–1316. [CrossRef]
85. Kundu, R.; Chattopadhyay, S.; Cuevas, E.; Sarkar, R. AltWOA: Altruistic Whale Optimization Algorithm for feature selection on microarray datasets. *Comput. Biol. Med.* **2022**, *144*, 105349. [CrossRef]
86. Balakrishnan, K.; Dhanalakshmi, R.; Seetharaman, G. S-shaped and V-shaped binary African vulture optimization algorithm for feature selection. *Expert Syst.* **2022**, *10*, e13079. [CrossRef]
87. Akinola, O.A.; Ezugwu, A.E.; Oyelade, O.N.; Agushaka, J.O. A hybrid binary dwarf mongoose optimization algorithm with simulated annealing for feature selection on high dimensional multi-class datasets. *Sci. Rep.* **2022**, *12*, 1–22. [CrossRef]
88. Huang, C.-L.; Wang, C.-J. A GA-based feature selection and parameters optimizationfor support vector machines. *Expert. Syst. Appl.* **2006**, *31*, 231–240. [CrossRef]
89. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [CrossRef]
90. 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.
91. Blake, C. UCI repository of machine learning databases. Available online: http://www.ics.uci.edu/~mlearn/MLRepository-html-1998 (accessed on 22 July 2021).

92. Iwendi, C.; Bashir, A.K.; Peshkar, A.; Sujatha, R.; Chatterjee, J.M.; Pasupuleti, S.; Mishra, R.; Pillai, S.; Jo, O. COVID-19 patient health prediction using boosted random forest algorithm. *Front. Public Health* **2020**, *8*, 357. [CrossRef] [PubMed]
93. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In Proceedings of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Rhodes, Greece, 22–26 October 2003; pp. 986–996.
94. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Cheng, D. Learning k for knn classification. *ACM Trans. Intell. Syst. Technol. (TIST)* **2017**, *8*, 1–19. [CrossRef]
95. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Wang, R. Efficient kNN Classification With Different Numbers of Nearest Neighbors. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *29*, 1774–1785. [CrossRef] [PubMed]
96. Garcia, S.; Fernandez, A.; Luengo, J.; Herrera, F. A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Comput.* **2008**, *13*, 959–977. [CrossRef]
97. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
98. Glas, A.S.; Lijmer, J.G.; Prins, M.H.; Bonsel, G.J.; Bossuyt, P.M.M. The diagnostic odds ratio: A single indicator of test performance. *J. Clin. Epidemiol.* **2003**, *56*, 1129–1135. [CrossRef]
99. Ciotti, M.; Ciccozzi, M.; Terrinoni, A.; Jiang, W.-C.; Wang, C.-B.; Bernardini, S. The COVID-19 pandemic. *Crit. Rev. Clin. Lab. Sci.* **2020**, *57*, 365–388. [CrossRef]
100. Chakraborty, I.; Maity, P. COVID-19 outbreak: Migration, effects on society, global environment and prevention. *Sci. Total Environ.* **2020**, *728*, 138882. [CrossRef]
101. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [CrossRef]