Journal of Ambient Intelligence and Humanized Computing (2021) 12:7637–7665
https://doi.org/10.1007/s12652-020-02484-z

ORIGINAL RESEARCH

Binary β-hill climbing optimizer with S-shape transfer function for feature selection

Mohammed Azmi Al-Betar[1,2] · Abdelaziz I. Hammouri[3] · Mohammed A. Awadallah[4] · Iyad Abu Doush[5,6]

Received: 26 September 2019 / Accepted: 17 August 2020 / Published online: 28 August 2020

Abstract
Feature selection is an essential stage in many data mining and machine learning and applications that find the proper subset of features from a set of irrelevant, redundant, noisy and high dimensional data. This dimensional reduction is a vital task to increase classification accuracy and thus reduce the processing time. An optimization algorithm can be applied to tackle the feature selection problem. In this paper, a β-hill climbing optimizer is applied to solve the feature selection problem. β-hill climbing is recently introduced as a local-search based algorithm that can obtain pleasing solutions for different optimization problems. In order to tailor β-hill climbing for feature selection, it has to be adapted to work in a binary context. The S-shaped transfer function is used to transform the data into the binary representation. A set of 22 *de facto* benchmark real-world datasets are used to evaluate the proposed algorithm. The effect of the β-hill climbing parameters on the convergence rate is studied in terms of accuracy, the number of features, fitness values, and computational time. Furthermore, the proposed method is compared against three local search methods and ten metaheuristics methods. The obtained results show that the proposed binary β-hill climbing optimizer outperforms other comparative local search methods in terms of classification accuracy on 16 out of 22 datasets. Furthermore, it overcomes other comparative metaheuristics approaches in terms of classification accuracy in 7 out of 22 datasets. The obtained results prove the efficiency of the proposed binary β-hill climbing optimizer.

Keywords Feature selection · β-hill climbing optimizer · S-shape transfer function · Optimization · Dimensionality reduction

1 Introduction

Data mining research community works on designing and improving techniques for data classifications (Mashrgy et al. 2014; Al-Abdallah et al. 2017), pattern recognition (Ma and Xia 2017), and machine learning (Lee and Lee 2006; Doush and Sahar 2017; Sawalha and Doush 2012). Some data mining problems contain huge data with thousands of features. In many cases, only a set of proper features is needed while the others are redundant, irrelevant or noisy. Picking a subset of these features to accurately represent the entire set of features can largely affect the performance of machine learning algorithms in different properties such as time complexity and classification accuracy (Hu et al. 2006).

✉ Mohammed A. Awadallah
ma.awadallah@alaqsa.edu.ps

Mohammed Azmi Al-Betar
m.albetar@ajman.ac.ae

Abdelaziz I. Hammouri
Aziz@bau.edu.jo

Iyad Abu Doush
idoush@auk.edu.kw

1 Department of Information Technology - MSAI, College of Engineering and Information Technology, Ajman University, Ajman, UAE

2 Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, P.O. Box 50, Al-Huson, Irbid, Jordan

3 Department of Computer Information Systems, Al-Balqa Applied University, Al-Salt, Jordan

4 Department of Computer Science, Al-Aqsa University, P. O. Box 4051, Gaza, Palestine

5 Computer Science Department, American University of Kuwait, Salmiya, Kuwait

6 Computer Science Department, Yarmouk University, Irbid, Jordan

Feature selection (FS) is choosing a relevant set of features from a large group of features to represent a record in a dataset. The feature selection technique is applied in many applications like text classification (Forman 2003; Deng et al. 2019), text mining (Jing et al. 2002; Ravisankar et al. 2011), image recognition (Goltsev and Gritsenko 2012), image retrieval (Rashedi et al. 2013; Dubey et al. 2015), bioinformatic data mining (Saeys et al. 2007; Urbanowicz et al. 2018), and many others reported in (Bolón-Canedo and Alonso-Betanzos 2019).

There are three types of feature selection techniques based on the evaluation criteria: filter-based, wrapper-based, and embedded-based methods (Li et al. 2017). Firstly, filter-based feature selection methods give a score for each feature in the dataset using a statistical measure [e.g., information gain (Shang et al. 2013), Chi-squared test (Liu and Setiono 1995), or ReliefF (Robnik-Šikonja and Kononenko 2003)]. Then these features are ranked based on their score. As a result, the features with the higher ranking are kept, while the features with lower rank are removed. Secondly, wrapper-based feature selection methods use search algorithms (e.g., genetic algorithm or particle swarm optimization) to evaluate the generated subset of features. After completing the search process, one of the classifiers (e.g., k-nearest neighbor (Park and Kim 2015), naive bayes (Bermejo et al. 2014), decision trees (Sindhu et al. 2012), etc.) is used to evaluate the quality of the chosen subset of features in term of accuracy. Finally, the integration of a wrapper-based and a filter-based method is known as an embedded-based method in which the searching algorithm is embedded in the classifier such as the k-nearest neighbor algorithm (kNN). Then it guides the classifier to pick some features that can achieve higher accuracy.

In the context of optimization, the FS is considered not easy to solve combinatorial optimization problem (Gheyas and Smith 2010). The complexity of the FS problem comes from selecting the relevant set of features from a plenty possible subsets. For example, the power-set of the set $\mathcal{A}$ of size N features contains $2^N - 1$ possible subsets of features. Therefore, as the number of features increase the number of solutions to look for the problem increases exponentially. The picked set of features is modeled using a function which is guided by the accuracy of the classification and the number of used feature. The FS solution is conventionally expressed as a binary array of the selected features.

The brute-force method can be used to solve the FS problem where all possible subsets generated and evaluated, then the relevant subset is identified (Lai et al. 2006). This type of algorithm cannot be used when we have a large number of features. Heuristic algorithms can be utilized to obtain the optimal set for the FS problem (Zhong et al. 2001). This type of algorithms can find efficiently the subset of relevant features. However, the quality of this acceptable subset is not necessarily guaranteed (Talbi 2009). Therefore, researchers use metaheuristic-based algorithms to find an optimal portion of relevant features in a feasible time with high classification accuracy.

Metaheuristic-based algorithms can be used to solve different kinds of optimization problems using self-learning operators that is configured with operators to efficiently explore and exploit possible solutions, hoping to attain the best solution (Blum and Roli 2003). Metaheuristic-based algorithms can be classified into population-based and local search-based algorithms (Blum and Roli 2003). Population-based algorithms examine several search space regions concurrently and improve them iteratively wishing to obtain the optimal solution. Examples of population-based algorithm for FS include genetic algorithm (Ghareb et al. 2016), differential evolution (Mlakar et al. 2017), ant lion optimizer (Emary et al. 2016), grey wolf optimization (Emary et al. 2016), ant colony optimization (Kabir et al. 2012), competitive swarm optimizer (Gu et al. 2018), firefly algorithm (Zhang et al. 2018; Al-Abdallah et al. 2017), grasshopper optimization algorithm (Mafarja et al. 2018b, 2019), bat algorithm (Mafarja et al. 2018b), whale optimization algorithm (Mafarja and Mirjalili 2018), dragonfly optimization (Mafarja et al. 2018a), crow search algorithm (Sayed et al. 2019), gravitational search algorithm (Taradeh et al. 2019), and harmony search algorithm (Moayedikia et al. 2017)

Local search-based algorithms, the focal point of this paper, consider one solution at a time. Let's call this the initial solution. It will be modified repetitively using an operator which allow visiting nearby values until a peak local value is found. A local search-based algorithm is capable of thoroughly investigate a specific region of the initial solution and find the local optima. Such algorithms have a limitation of not exploring multi-search space regions concurrently. Therefore, some random strategies are employed to empower the local search-based approach. In the literature, FS is tackled by several local search-based algorithms such as tabu search (Zhang and Sun 2002), GRASP (Bermejo et al. 2011), iterated local search (Marinaki and Marinakis 2015), variable neighborhood search (Marinaki and Marinakis 2015), and stochastic local search method (Boughaci and Alkhawaldeh 2018).

Due to the complex nature of FS problems, most FS-based algorithms are either a modification of a metaheuristic algorithm or a hybridization of two or more metaheuristic algorithms. Examples of modified metaheuristic algorithms that are used to solve FS problems are binary ant lion optimizer using S-shaped function and V-shaped function (Emary et al. 2016), binary grey wolf optimization using crossover and sigmoidal function (Emary et al. 2016), binary dragonfly optimization using time-varying transfer functions (Mafarja et al. 2018a), and chaotic crow search algorithm (Sayed et al. 2019). Examples of hybrid metaheuristics are the hybridization of the ant colony optimization with the wrapper and filter approaches (Kabir et al. 2012), and the integration of the gravitational search algorithm with evolutionary crossover and mutation operators (Taradeh et al. 2019).

β-hill climbing is a local search-based algorithm that is recently introduced Al-Betar (2017). It is an improved version of the hill-climbing algorithm with a β operator that is governed by the β parameter to diversify the search as well as a neighboring operator called $\mathcal{N}$ to intensify the search. The β operator empowers the β-hill climbing to intelligently escape the local optima by searching different regions and digging deeply into the regions of the search space. Due to its simplicity, the algorithm is adapted in a broad range optimization problems such as ECG and EEG signal denoising (Alyasseri et al. 2017a, b, 2018), generating substitution-boxes (Alzaidi et al. 2018), gene selection (Alomari et al. 2018b), economic load dispatch problem (Al-Betar et al. 2018), mathematical optimization functions (Abed-alguni and Alkhateeb 2018; Abed-alguni and Klaib 2018), multiple-reservoir scheduling (Alsukni et al. 2017), sudoku game (Al-Betar et al. 2017), text document clustering (Abualigah et al. 2017a, b), and cancer classification (Alomari et al. 2018a). β-hill climbing produces successful outcomes for a broad range of optimization problems. It produces pleasing results for many real-world problems, and such algorithm can be used to tackle the FS problem.

In this paper, a binary version of β-hill climbing optimizer is developed to tackle the problem of feature selection. The algorithm is evaluated using 22 UCI machine learning benchmark datasets that are used widely in the literature. The evaluation is discussed using four measurement criteria which are the fitness function, the classification accuracy, the number of relevant features, and the elapsed CPU time. The impact of the parameter settings (β and $\mathcal{N}$) for the binary β-hill climbing optimizer is carefully analyzed and studied. Furthermore, the effect of using different transfer functions as well as the different classifiers on the efficiency of the binary β-hill climbing is studied. The comparative evaluation is conducted against three local search methods using the same datasets on the four evaluation criteria. Interestingly, the proposed binary β-hill climbing optimizer excel other comparative local search methods 16 out of 22 datasets. On the other hand, it overcomes other comparative metaheuristics methods approaches in 7 out of 22 datasets and very-closed to the best results for the remaining 15 datasets. The binary β-hill climbing optimizer can be considered as a very important addition to the body of knowledge in the machine learning and classifications domain as it produces very promising outcomes when compared against other methods.

The rest of the paper is organized as follows: in sect. 2, the procedural steps of the proposed binary β-hill climbing algorithm is provided. The parameter setting analysis and comparative evaluations of the work are discussed in sect. 3. Finally, the conclusion is presented and the possible future research directions are shown in sect. 4.

2 Binary β-hill climbing optimizer for feature selection

Metaheuristic-based methods can be local search-based or population-based. Most of the techniques applied to resolve the feature selection problem are population-based methods such as swarm-based algorithms or evolutionary-based algorithm.

β-hill climbing optimizer is an enhanced variant of the basic hill climbing algorithm. It is a local search-based method proposed in Al-Betar (2017) to escape from being stuck into a local optima. β-hill climbing optimizer can be initiated with a random or heuristic solution [say $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$]. In each step, the current solution can be improved using three operators: (i) $\mathcal{N}$-operator which is controlled by $\mathcal{N}$ parameter to exploit a specific region on the search space, (ii) β-operator which is controlled by β parameter to examine the solution space, and (iii) $\mathcal{S}$-operator that utilizes the principle of survival-of-the-fittest. Normally, the search process of β-hill climbing optimizer is halted once the maximum number of iterations/time is reached.

The β-hill climbing optimizer can be applied for discrete or continuous search spaces. The variables of the feature selection problem are assigned by binary values (i.e., being selected or not). Therefore, a new operator called $\mathcal{T}$-operator (i.e., transfer operator) is proposed to transfer the variable into binary using the sinusoidal function. Consequently, the new version of β-hill climbing optimizer is called binary β-hill climbing optimizer. The algorithm pseudo-code is shown in Algorithm 1 and it is flow-charted in Fig. 1. The steps and the four operators are described as follows:

Step 1: Initialize binary β-hill climbing and feature selection parameters— The parameters of the binary β -hill climbing for feature selection are set in this step. The problem of feature selection is known to have a binary search space. Therefore, the solution is modeled as a binary vector $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ of n features. The value of $x_i = 1$ means that the feature i is considered. The parameters of the binary β-Hill climbing optimizer are $\mathcal{N}$ and β. The parameter $\mathcal{N} \in [0, 1]$ controls the neighboring operator ($\mathcal{N}$-operator) which is responsible for determin-

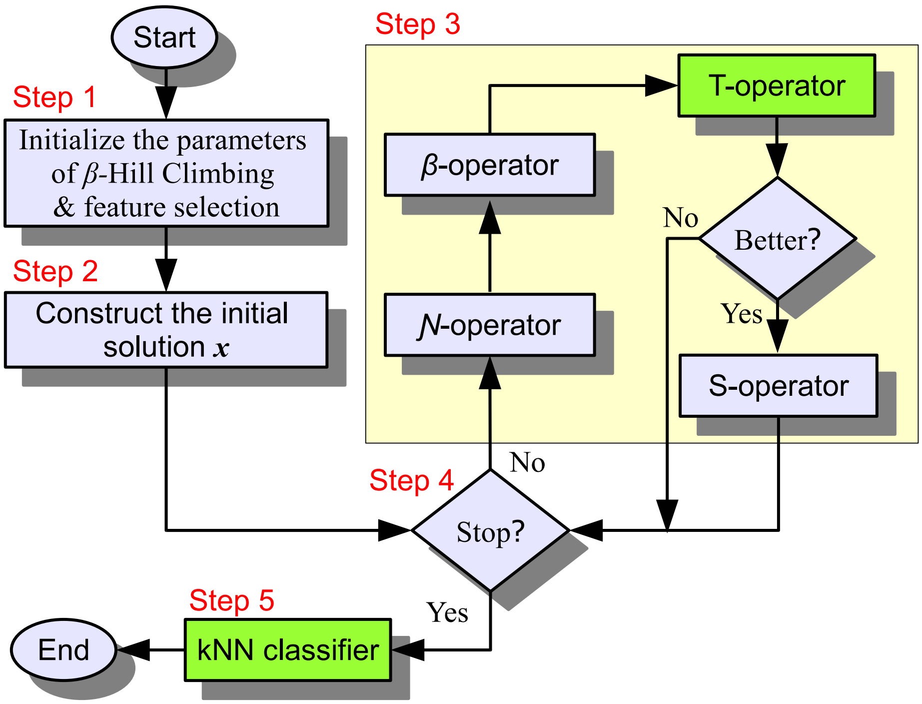

Fig. 1 Binary β-hill climbing optimizer for feature selection

ing the adjustment bandwidth to modify current solution to a neighboring solution. The β parameter controls the β-operator which determines the intensity of utilizing exploration in the neighboring solution. The last parameter of the proposed binary β-hill climbing optimizer is the maximum number of iterations which is Max_itr.
Step 2: Construct the initial solution— The initial solution $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ is randomly constructed from the binary domain as follows:

$$x_i \leftarrow \begin{cases} 1 & U[0,1] \geq 0.5 \\ 0 & otherwise. \end{cases}$$

where $U[0, 1]$ generates a random number between 0 and 1. In order to evaluate the initial solution, the set of features in the current solution are evaluated using the objective function formulated in Eq. (1) (Emary et al. 2016).

$$f(\boldsymbol{x}) = \alpha\gamma_R(D) + (1-\alpha)\frac{|R|}{|N|} \tag{1}$$

where the classification error rate is expressed as $\gamma_R(D)$. In this study, the kNN classifier is used to find the classification error rate (Liao and Vemuri 2002). Note that $|R|$ is how many features are selected, $|N|$ is the count of the entire features, α refers to the role of classification rate and the length of feature subset, $\alpha \in [0, 1]$.
Step 3: Improvement loop— The enhancement of the current feature selection solution ($\boldsymbol{x}$) is achieved by using four operators which are used to yield a neighboring solution (i.e., $\boldsymbol{x}'$).

$\mathcal{N}$-operator— This operator is responsible for moving the present solution $\boldsymbol{x}$ to the near by solution $\boldsymbol{x}'$ using Eq. (2). This operator is governed by the likeliness of picking $\mathcal{N}$ parameter where $\mathcal{N} \in [0, 1]$. The probability determines the adjustment of the decision variables (features) in the current solution. A greater value of $\mathcal{N}$ aid in a furthest movement from the neighboring solution $\boldsymbol{x}'$. The pseudo-code for $\mathcal{N}$-operator is shown in line 5 of Algorithm 1. Formally, let x_i be given the value of $v_i(k)$ of k^{th} position, then the following present how to allocate the value of x_i':

$$x_i' = x_{i,k} \pm \mathcal{N} \tag{2}$$

where $x_{i,k} \pm \mathcal{N}$ is the neighboring value of $x_{i,k}$.
β-operator— This operator is utilized for increasing the regions covered from the search space. It utilizes the idea of invariable mutation to the present solution. As shown in Eq. (3), $\forall i \in (1, 2, \dots, n)$, the x_i decision variable is picked at random to be adjusted using the β parameter. This is pseudo-coded in lines from 6 to 10 in Algorithm 1.

$$x_i' \leftarrow \begin{cases} x_r & U[0,1] \leq \beta \\ x_i' & otherwise. \end{cases} \tag{3}$$

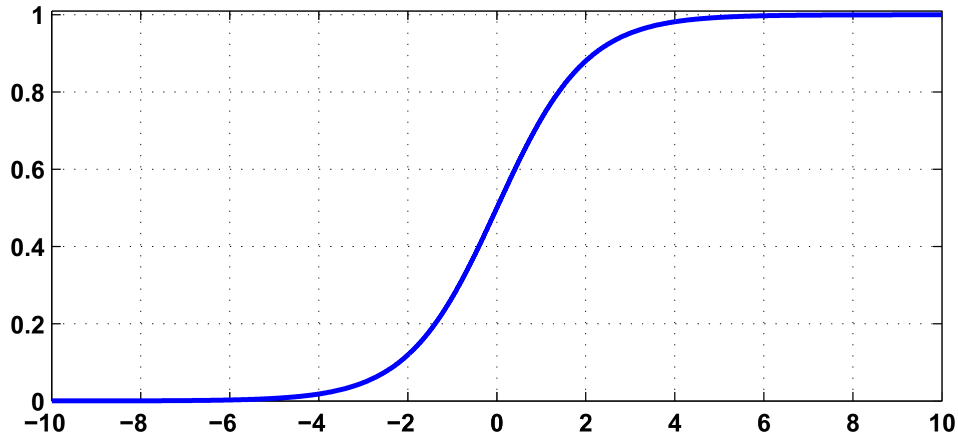

Fig. 2 S-shape function

Note that the β parameter determines how often the uniform mutation is used. Also, x_r is a random value which is either 0 or 1.

$\mathcal{T}$-operator— As aforementioned, the feature selection problem deals with a binary values for the decision variables. Therefore, the sinusoidal function (or S-shape function as shown in Fig. 2) is adapted to transform continuous solutions into binary. To elaborate, the Sigmoidal function (Kennedy and Eberhart 1997) is formulated in Eq. (4):

$$T\left(x_i'\right) = \frac{1}{1+e^{-x_i'}} \quad \forall i = (1, 2, \dots, n) \tag{4}$$

The value of the decision variables in the neighboring solution is re-assigned a binary value using Eq. (5). Let r be function that generates at random a value bounded by 0 and 1 (i.e., $r \in [0, 1]$), the value of x_i' of the feature i will be re-assigned as follows:

$$x_i' = \begin{cases} 1 & r < T\left(x_i'\right) \\ 0 & Otherwise) \end{cases} \tag{5}$$

$\mathcal{S}$-operator— The quality of the neighboring solution $\boldsymbol{x}'$ is assessed by applying the objective function $f(\boldsymbol{x}')$ which is formulated in 1. The neighboring solution $\boldsymbol{x}'$ is interchanged by the current one $\boldsymbol{x}$, if it is better (i.e., $f(\boldsymbol{x}') \leq f(\boldsymbol{x})$). The pseudo-code for the $\mathcal{S}$-operator is presented in the lines from 20 to 22 of Algorithm 1.

Step 4: Stop criterion— The proposed binary β-hill climbing is iterated until a stop criterion is reached. The stop criterion used in this study is based on the number of iteration Max_Itr defined at Step 1.

Step 5: kNN classifier— The accuracy of the obtained solution by binary β-hill climbing is evaluated using a kNN classifier. kNN is an effective non-parametric method that is utilized for classification and regression. The kNN starts by storing all the training data instances. After that, a pairwise computation is applied to calculate the similarity between the training instances when compared against the unseen instances (Chen et al. 2009; Weinberger and Saul 2009). Then selecting the k-closest instances. This operation is done repetitively for all the unseen instances.

In order to compute the classification accuracy and error rate measurements, Eqs. (6) and (7) are used. Classification accuracy is a statistical measure which defines the ability of the classifier to correctly use the picked features to precisely label a given tuple into a class. It can be computed using Eq.(6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where TP (true positive) denotes identifying correctly the class using a precise set of features. TN (true negative) denotes identifying correctly that it is not the class using a correct set of features. FP (false positive) denotes identifying incorrectly that it is the class. Finally, FN (false negative) denotes identifying incorrectly that it is not the class.

The classification error rate calculated in Eq.(7) is used to determine the percentage of features that are incorrectly assigned. It will be part of the formulated objective function.

$$\gamma_R(D) = 1 - Accuracy \tag{7}$$

Algorithm 1 Binary β–hill climbing pseudo-code

1: Initialize the parameters of β-Hill climbing and feature selection
2: $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ { Generate the initial solution x}
3: Calculate $f(x)$
4: $itr = 0$
5: **while** $(itr \leq \text{Max\_Itr})$ **do**
6: $\quad x'_i = v_i \pm \mathcal{N}$ $\{\mathcal{N}$-operator$\}$
7: $\quad$ **for** $i = 1, \cdots, n$ **do**
8: $\quad\quad$ **if** $(U[0,1] \leq \beta)$ **then**
9: $\quad\quad\quad x'_i = v_i(r)$ $\{\beta$-operator$\}$
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: $\quad$ **for** $i = 1, \cdots, n$ **do**
13: $\quad\quad T(x'_i) = \frac{1}{1+e^{-x'_i}}$ $\{\mathcal{T}$-operator$\}$
14: $\quad\quad$ **if** $r < T(x'_i)$ **then**
15: $\quad\quad\quad x'_i = 1$
16: $\quad\quad$ **else**
17: $\quad\quad\quad x'_i = 0$
18: $\quad\quad$ **end if**
19: $\quad$ **end for**
20: $\quad$ **if** $f(\boldsymbol{x}') \leq f(\boldsymbol{x})$ **then**
21: $\quad\quad x = x'$ $\{\mathcal{S}$-operator$\}$
22: $\quad$ **end if**
23: $\quad itr = itr + 1$
24: **end while**
25: kNN classifier(x)

2.1 Computational complexity of the proposed method

The time complexity required for the proposed binary β-hill climbing algorithm is measured by analyzing the pseudo-code given in Algorithm 1 in terms of big-$\mathcal{O}$ notation. The binary β -hill climbing pseudo-code can be divided into three parts: (i) The initial phase (from line 1 to line 3 in Algorithm 1); (ii) The improvement phase (from line 5 to line 24 in Algorithm 1); and (iii)The classifier phase (the line 25 in Algorithm 1).

The time complexity of the *initial phase* is $\mathcal{O}(n)$ for the construction of the initial solution. The *f(x)* calculation is based on the kNN classifier which computes the classification error rate and thus its complexity is $\mathcal{O}(n^2)$. Therefore, the time complexity for the initial phase is $\mathcal{O}(n^2)$.

The time complexity of the second phase (i.e., improvement phase) rely upon number of iterations (Max_Itr) and the time required for $\mathcal{N}$-operator, β-operator, $\mathcal{T}$-operator, and $\mathcal{S}$-operator. The time complexity of the $\mathcal{N}$-operator is $\mathcal{O}(1)$ while the complexity of the β-operator is $\mathcal{O}(n)$. Furthermore, the time complexity of the $\mathcal{T}$-operator is also $\mathcal{O}(n)$ while $\mathcal{S}$-operator requires $\mathcal{O}(n^2)$. In brief, the time complexity of the second phase is $\mathcal{O}(\text{Max\_Itr} \cdot n^2)$.

The time complexity of the classifier phase is $\mathcal{O}(n^2)$ which is the time required to execute the kNN classifier. as a wrap-up, the time complexity required to execute the developed binary β-hill climbing is $\mathcal{O}(\text{Max\_Itr} \cdot n^2)$.

3 Experiments and results

A comprehensive experimental analysis is conducted in this section to investigate the proposed binary βHC algorithm efficiency when solving the problem of feature selection. The experiments are divided as follows: (i) the effect of two parameters of binary βHC (i.e., $\mathcal{N}$ and β) on the algorithm performance is studied in sect. 3.2.1 and 3.2.2; (ii) the influence of different transfer functions on the efficiency of the proposed binary βHC algorithm is presented in sect. 3.2.3; (iii) the performance of the proposed binary βHC algorithm using different classifiers is summarized in sect. 3.2.4; (iv) The effect of training/testing against k-fold cross validation models on the performance of the proposed algorithm is provided in Sec.3.2.5 ; and (v) the efficiency of the binary βHC algorithm is compared against other local search-based algorithms in sect. 3.3, it is compared against recent metaheuristic methods in sect. 3.4, and it also compared against filter-based approach in sect. 3.5. It should be noted that the attributes of the datasets utilized in the algorithm assessment are summarized in sect. 3.1.

The binary βHC algorithm was implemented using MATLAB (R2014a) and tested on a laptop with 2.80 Intel Core i7 with 16 GB RAM. The operating system installed on the laptop is Microsoft Windows 10. In all the experiments each dataset is splitted at random into two portions: training which is 80% of the instances and testing which is the remaining 20%. This split is used as it has been widely adapted by several state-of-the-art methods (Mafarja et al. 2019; Alsaafin and Elnagar 2017; Li et al. 2011; Wieland and Pittore 2014)

3.1 Dataset

The proposed binary βHC algorithm is evaluated using twenty-two datasets collected from the UCI data repository. A brief of the datasets characteristics is presented in Table 1.

Table 1 The characteristics of the datasets

Dataset	No. of features	No. of instances	
Small	Tic-tac-toe	9	958
	Breastcancer	9	699
	HeartEW	13	270
	Exactly2	13	1000
	Exactly	13	1000
	M-of-n	13	1000
	WineEW	13	178
	CongressEW	16	435
	Vote	16	300
	Zoo	16	101
	Lymphography	18	148
	SpectEW	22	267
	BreastEW	30	596
	IonosphereEW	34	351
	KrvskpEW	36	3196
	WaveformEW	40	5000
	SonarEW	60	208
Medium	Clean1	166	476
	Semeion	265	1593
	PenglungEW	325	73
Large	Colon	2000	62
	Leukemia	7129	72

The summary shows for each dataset the FS problem name, the number of features, and the instances count.

3.2 Evaluation of the proposed binary βHC algorithm

A sensitivity analysis for the proposed binary βHC algorithm is performed to investigate the effect of different operators on the convergence of the algorithm. Twenty-three experimental scenarios are designed as shown in Table 2. In this table, these scenarios are divided to four groups as follows: firstly, five scenarios (Sen1–Sen5) are designed to study the influence of the $\mathcal{N}$ parameter on the performance of the binary βHC algorithm. The next five experimental scenarios (Sen6–Sen10) investigate the convergence behavior of the binary βHC algorithm by tunning the β parameter. The third group of experimental scenarios (Sen11–Sen18) are designed in order to investigate the influence of the different transfer functions on the the proposed binary βHC algorithm efficiency. The effect of the classifiers on the behavior of the developed binary βHC algorithm is investigated in the last three scenarios (Sen19–Sen21). Finally, the last two scenarios (i.e., Sen22, and Sen23) are designed in order to study the influence of data splitting techniques

Table 2 Twenty three experimental scenarios to evaluate the sensitivity of binary βHC algorithm

Experimental scenario	$\mathcal{N}$	β	Trans. fun.	Classifiers	Notes
Sen1	0.005	0.05	S2	kNN	
Sen2	0.05				
Sen3	0.1				
Sen4	0.5				
Sen5	0.9				
Sen6	0.9	0	S2	kNN	
Sen7		0.005			
Sen8		0.05			Sen8 = Sen5
Sen9		0.1			
Sen10		0.5			
Sen11	0.9	0.5	S1	kNN	
Sen12			S2		Sen12 = Sen8
Sen13			S3		
Sen14			S4		
Sen15			V1		
Sen16			V2		
Sen17			V3		
Sen18			V4		
Sen19	0.9	0.5	S2	kNN	Sen19 = Sen12
Sen20				SVM	
Sen21				decision tree	
Sen22	0.9	0.5	S2	kNN+traninig	Sen22 = Sen19
Sen23				kNN+fold	

Table 3 The classification accuracy results obtained by binary βHC algorithm with varies $\mathcal{N}$ values

Dataset	Sen1 ($\mathcal{N}$ =0.005)		Sen2 ($\mathcal{N}$=0.05)		Sen3 ($\mathcal{N}$=0.1)		Sen4 ($\mathcal{N}$=0.5)		Sen5 ($\mathcal{N}$=0.9)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	0.780	0.012	0.786	0.024	0.781	0.017	**0.800**	0.009	0.776	0.006
Breastcancer	0.974	0.005	0.969	0.007	0.961	0.002	0.957	0.006	**0.967**	0.003
HeartEW	0.786	0.015	0.824	0.012	0.798	0.023	**0.859**	0.017	0.821	0.012
Exactly2	**0.719**	0.018	0.709	0.018	0.717	0.017	0.703	0.019	0.707	0.011
Exactly	0.917	0.147	0.966	0.105	**1.000**	0.002	0.969	0.095	0.967	0.100
M-of-n	0.988	0	**1.000**	0	**1.000**	0	**1.000**	0	**1.000**	0
WineEW	0.970	0.009	**0.989**	0.008	0.972	0.007	0.976	0.016	0.976	0.015
CongressEW	0.959	0.007	0.956	0.005	**0.975**	0.012	0.954	0.006	0.964	0.006
Vote	0.961	0.009	0.953	0.009	0.955	0.006	**0.965**	0.008	0.948	0.006
Zoo	0.915	0.019	**0.977**	0.038	0.926	0.029	0.874	0.015	0.947	0.009
Lymphography	0.801	0.030	0.829	0.022	0.841	0.028	0.872	0.038	**0.886**	0.029
SpectEW	**0.859**	0.010	0.838	0.013	0.769	0.009	0.838	0.015	0.854	0.013
BreastEW	0.955	0.007	0.955	0.011	0.959	0.008	0.958	0.006	**0.960**	0.007
IonosphereEW	0.933	0.013	0.940	0.013	0.947	0.011	**0.949**	0.010	0.928	0.013
KrvskpEW	0.981	0.001	0.983	0.004	**0.985**	0.002	0.974	0.007	0.984	0.003
WaveformEW	0.756	0.007	0.760	0.011	0.763	0.010	0.764	0.007	**0.772**	0.008
SonarEW	0.931	0.021	0.913	0.022	**0.953**	0.010	0.951	0.017	0.928	0.017
clean1	0.912	0.017	0.887	0.018	0.911	0.016	0.906	0.010	**0.919**	0.014
semeion	0.979	0.003	0.979	0.005	**0.989**	0.003	0.984	0.004	0.986	0.003
PenglungEW	**0.909**	0.018	0.897	0.019	0.872	0.026	0.884	0.040	0.873	0.025
Colon	**0.868**	0.027	0.785	0.028	0.735	0.031	0.721	0.016	0.810	0.021
Leukemia	0.894	0.026	**0.971**	0.006	0.869	0.020	0.936	0.022	0.965	0.012

(i.e., training/testing against k-fold cross-validation) on the performance of the proposed βHC algorithm. It should be noted that 20 independent replications is conducted for each experimental scenario, and the maximum number of iterations is 500. As suggested in (Emary et al. 2016), the value of the k parameter in the kNN algorithm is set to 5. Note that the bold results obtained in all result tables refers to the best results obtained.

3.2.1 Study the effect of the $\mathcal{N}$ parameter

The parameter $\mathcal{N}$ influence on the performance of the binary βHC algorithm is investigated in this section. Five experimental scenarios are designed with different settings of $\mathcal{N}$ (i.e., Sen1 ($\mathcal{N}$=0.005), Sen2 ($\mathcal{N}$=0.05), Sen3 ($\mathcal{N}$=0.1), Sen4 ($\mathcal{N}$=0.5), and Sen5 ($\mathcal{N}$=0.9)). In general, a higher value for the $\mathcal{N}$ parameter leads to a higher exploitation and makes the algorithm dig deeper in the searched region. Tables 3, 4, 5, and 6 provide the average (Avg) and the standard deviation (Stdv) of the results obtained by running Sen1 to Sen5 in terms of the classification accuracy, the fitness value, the selected features, and the elapsed CPU time. Note that the best results in these tables are highlighted in **bold** font.

Table 3 shows the behavior of the proposed binary βHC algorithm with different values of the parameter $\mathcal{N}$ in terms of classification accuracy. The highest average values are the best. As shown in Table 3, the two scenarios (i.e., Sen3 and Sen5) obtained the best results on 6 datasets. While the three other scenarios (i.e., Sen1, Sen2, and Sen4) obtain the best results on 4, 5, and 5 datasets respectively. Based on the above findings, it is not clear which scenario configuration is the most efficient. In other words, no real impact of the parameter $\mathcal{N}$ on the performance of binary βHC. In addition, the standard deviation values recorded in Table 3 show the robustness of the proposed algorithm. Clearly, the five experimental scenarios has low standard deviation values in all datasets, with a better performance for Sen5.

Table 4 present the influence of different values of the parameter $\mathcal{N}$ on the performance of the binary βHC algorithm in terms of fitness value. The best outcomes achieved are highlighted in bold font (lowest is the best).Clearly, Table 4 shows that the efficiency of the binary βHC algorithm in the five experimental scenarios (Sen1 to Sen5) is almost the same. The two experimental scenarios Sen3 and Sen4 achieved the best results in 6 datasets, while Sen1 get the utmost results in 5 datasets. In addition, Sen1 and Sen5 achieved the best results in 4 datasets. The lowest standard derivation values in all datasets happened in Sen1 to Sen 5.

Table 4 The fitness values obtained by binary βHC algorithm with varies $\mathcal{N}$ values

Dataset	Sen1 ($\mathcal{N}$ =0.005)		Sen2 ($\mathcal{N}$=0.05)		Sen3 ($\mathcal{N}$=0.1)		Sen4 ($\mathcal{N}$=0.5)		Sen5 ($\mathcal{N}$=0.9)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	0.224	0.011	0.219	0.023	0.224	0.017	**0.205**	0.010	0.228	0.006
Breastcancer	**0.031**	0.004	0.036	0.007	0.044	0.003	0.047	0.006	0.038	0.003
HeartEW	0.216	0.015	0.180	0.012	0.206	0.021	**0.145**	0.017	0.182	0.013
Exactly2	**0.283**	0.017	0.294	0.017	0.285	0.018	0.301	0.018	0.295	0.011
Exactly	0.086	0.146	0.038	0.103	**0.005**	0.002	0.035	0.094	0.037	0.099
M-of-n	0.016	0	**0.005**	0	**0.005**	0	**0.005**	0	**0.005**	0
WineEW	0.035	0.009	**0.016**	0.007	0.033	0.007	0.028	0.016	0.028	0.015
CongressEW	0.043	0.007	0.046	0.005	**0.028**	0.013	0.048	0.006	0.038	0.006
Vote	0.042	0.009	0.050	0.010	0.048	0.007	**0.037**	0.008	0.054	0.006
Zoo	0.088	0.018	**0.027**	0.038	0.076	0.029	0.129	0.015	0.056	0.009
Lymphography	0.201	0.030	0.173	0.022	0.161	0.028	**0.117**	0.037	0.130	0.029
SpectEW	**0.144**	0.010	0.164	0.013	0.231	0.008	0.164	0.015	0.148	0.013
BreastEW	0.049	0.007	0.048	0.011	0.045	0.008	0.046	0.006	**0.043**	0.007
IonosphereEW	0.070	0.013	0.063	0.013	0.056	0.011	**0.054**	0.010	0.074	0.014
KrvskpEW	0.023	0.002	0.021	0.004	**0.018**	0.001	0.029	0.007	0.021	0.003
WaveformEW	0.246	0.008	0.243	0.011	0.240	0.011	0.238	0.007	**0.230**	0.008
SonarEW	0.073	0.020	0.090	0.022	**0.050**	0.010	0.053	0.017	0.075	0.017
clean1	0.091	0.017	0.116	0.018	0.092	0.016	0.097	0.010	**0.085**	0.014
semeion	0.026	0.003	0.026	0.005	**0.015**	0.003	0.021	0.004	0.019	0.004
PenglungEW	**0.093**	0.018	0.105	0.019	0.130	0.026	0.118	0.040	0.129	0.025
Colon	**0.135**	0.027	0.216	0.028	0.266	0.030	0.280	0.016	0.192	0.020
Leukemia	0.109	0.026	**0.033**	0.006	0.134	0.020	0.068	0.022	0.039	0.012

These experimental scenarios obtain almost the same results over 20 runs.

The results of Sen1 to Sen5 on the proposed algorithm in terms of the selected features are outlined in Table 5. Again lowest values (i.e., best) are point up using bold font. The developed binary βHC algorithm efficiency using Sen5 outperforms the other four scenarios (Sen1–Sen4) as it achieves the best results on 7 datasets. While the other scenarios Sen1, Sen2, Sen3, and Sen4 get the best results on 3, 4, 6, and 2 datasets respectively.

Similarly, the impact of the parameter $\mathcal{N}$ by utilizing different configurations on the efficiency of the proposed binary βHC algorithm in terms of the elapsed CPU time is recorded in Table 6. Clearly, higher values of the parameter $\mathcal{N}$ leads to a lower CPU time. In other words, the two experimental scenarios (Sen4 and Sen5) achieved the lowest CPU time on 8 datasets, while the other three scenarios Sen1, Sen2, and Sen3 obtained the lowest CPU time on 3, 1, and 2 datasets respectively.

In summary, the best results for the proposed βHC algorithm for most datasets in terms of the classification accuracy, , the selected features, and the elapsed CPU time is happened when $\mathcal{N} = 0.9$. Furthermore, the performance of the proposed βHC algorithm using different configurations of the parameter $\mathcal{N}$ is almost the same on all datasets in terms of the fitness value. As a result, in the next experiments the parameter $\mathcal{N}$ value is set to 0.9.

3.2.2 Study the effect of parameter β

The impact of the β parameter on the performance of the binary βHC algorithm is studied in this section using five various settings (β=0, β=0.005, β=0.05, β=0.1, and β=0.5). As a result, five empirical scenarios (Sen6–Sen10) are devised. The value of the parameter $\mathcal{N}$ is set to 0.9 based on the previous experiment conducted in sect. 3.2.1. Generally speaking, a larger β value results in a greater exploration rate. Tables 7, 8, 9, and 10 present the mean and the standard deviation of the results by running the scenarios from Sen6 to Sen10 in terms of the classification accuracy, the fitness value, the selected features,and the elapsed CPU time. The obtained finest outcomes are highlighted in **bold** font.

Table 7 summarizes the experimental results of running Sen6 to Sen10 in terms of the classification accuracy. Table 7 show that the efficiency of the binary βHC algorithm is enhanced by increasing the the parameter β value. In other words, Sen10 attained the best results on 10 out of 20 datasets, while the other scenarios Sen9, Sen8, and Sen7 achieved the finest outcomes on the datasets 7, 3, and 2 respectively. However, the performance of Sen6 is the worst

Table 5 The selected features obtained by binary βHC algorithm with varies $\mathcal{N}$ values

Dataset	Sen1 ($\mathcal{N}$=0.005)		Sen2 ($\mathcal{N}$=0.05)		Sen3 ($\mathcal{N}$=0.1)		Sen4 ($\mathcal{N}$=0.5)		Sen5 ($\mathcal{N}$=0.9)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	6.2	0.616	5.8	0.894	5.950	0.224	6.0	1.026	**5.550**	0.686
Breastcancer	5.1	0.718	**4.550**	0.999	5.1	1.252	4.8	0.696	4.650	1.089
HeartEW	**5.150**	1.309	7.5	1.433	7.5	2.090	5.850	1.309	6.350	2.254
Exactly2	6.050	2.605	7.250	1.482	**5.850**	2.084	8.550	0.999	6.950	2.438
Exactly	5.9	1.294	5.8	0.696	6.050	0.224	5.850	1.226	**5.350**	1.268
M-of-n	6.0	0	6.0	0	6.0	0	6.0	0	**5.850**	0.875
WineEW	6.550	0.887	6.650	1.565	6.5	1.638	**5.550**	1.276	6.250	1.482
CongressEW	4.550	1.605	4.650	1.268	5.8	1.281	5.150	2.346	**3.450**	2.328
Vote	4.950	1.356	4.5	1.638	**4.4**	1.314	4.550	1.276	4.7	2.473
Zoo	**5.2**	1.542	6.750	1.209	5.3	0.865	5.3	1.342	5.350	1.137
Lymphography	6.9	1.804	7.1	1.651	7.450	1.731	7.750	1.618	**6.7**	1.949
SpectEW	8.6	2.113	7.3	1.895	**4.150**	3.329	9.0	1.974	8.350	2.7
BreastEW	12.250	2.023	**10.550**	2.139	12.550	2.350	11.250	1.333	11.050	3.546
IonosphereEW	11.9	2.490	11.650	2.815	11.7	3.011	11.6	2.415	**10.350**	3.313
KrvskpEW	14.350	2.815	17.050	2.212	**12.650**	3.345	12.7	3.097	16.2	5.238
WaveformEW	21.1	2.845	20.150	2.961	20.750	2.936	20.4	2.542	**19.9**	3.144
SonarEW	24.1	3.024	**22.550**	2.685	23.2	3.622	25.250	2.807	25.3	3.840
clean1	**68.050**	5.216	69.050	6.992	68.6	5.642	70.7	4.390	71.850	7.043
semeion	123.550	8.185	122.250	10.492	**120.450**	7.749	125.750	9.695	123.4	8.075
PenglungEW	95.7	5.283	91.950	6.629	**90.6**	6.557	98.7	8.208	98.5	8.121
Colon	822.250	19.553	807.550	19.041	800.2	19.718	**778.1**	14.090	806.8	16.513
Leukemia	3220.050	48.621	**3179.2**	37.954	3194.950	39.349	3191.750	46.362	3191.550	49.618

when it is compared with other scenarios (i.e., Sen7, Sen8, and Sen10). This is because the parameter β is neglected in this scenario, and thus the source of exploration is not used. Furthermore, the standard derivation values are recorded in Table 7. It can be observed that Sen10 is more robust than the other scenarios on almost all the datasets as it obtains the same results over 20 runs.

Table 8 presents the results of examining the performance of the binary βHC algorithm by utilizing various values of the parameter β in terms of fitness values are recorded in . Table 8 provides clear evidence that the efficiency of the proposed binary βHC algorithm is enhanced with a larger value of the parameter β. This is for the reason that higher values of β lead to a higher rate of exploration, and thus avoid from being stuck in local optima. Clearly, the performance of Sen10 outperforms the other four scenarios (Sen6–Sen9) as it achieves the finest outcomes on 10 datasets. Furthermore, Sen9 achieved the best results in 8 datasets, while Sen8, Sen7, and Sen6 get the best results on 4, 3, and 1 datasets respectively. Nonetheless, the performance of the proposed binary βHC algorithm using Sen10 is more robust than other scenarios based on the standard derivation outcomes listed in Table 8.

Table 9 illustrates the results when running Sen6 to Sen10 in terms of the selected features by the proposed algorithm. Apparently, Sen8 outperform other scenarios, as it obtains the finest outcomes on 9 datasets. Furthermore, Sen7 and Sen9 obtained the finest outcomes on 5 datasets, while Sen10 achieved the finest outcomes on 2 datasets. Note that Sen6 did not obtain any best results for any of the datasets. This is because parameter β value is zero, and this makes the search process of the proposed algorithm to get suck in local optima.

Finally, the outcomes of the proposed binary βHC algorithm efficiency when tunning the parameter β in terms of the elapsed CPU time are outlined in Table 10. According to the results in Table 10, it can be seen that Sen8 obtained the minimum CPU time on 8 datasets. While the other scenarios Sen6, Sen7, Sen9, and Sen10 achieved the minimum CPU time on 1, 4, 5, and 4 datasets respectively.

In a nutshell, the proposed binary βHC algorithm with β =0.5 is superior than the other versions of the binary βHC algorithm in terms of the classification accuracy and the obtained fitness value. On the other hand, the efficiency of the proposed binary βHC algorithm with β=0.05 outperforms the other versions of the binary βHC algorithm in terms of the selected features and the elapsed CPU time. As a result, based on the classification accuracy results the value 0.5 is used to set the parameter β in the upcoming experiments.

Table 6 The CPU time (in Seconds) obtained by binary βHC algorithm with varies $\mathcal{N}$ values

Dataset	Sen1 ($\mathcal{N}$ =0.005)		Sen2 ($\mathcal{N}$=0.05)		Sen3 ($\mathcal{N}$=0.1)		Sen4 ($\mathcal{N}$=0.5)		Sen5 ($\mathcal{N}$=0.9)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	5.989	0.268	5.988	0.200	5.668	0.256	**5.513**	0.404	6.079	0.515
Breastcancer	5.276	0.104	4.970	0.102	5.036	0.142	4.987	0.118	**4.905**	0.103
HeartEW	4.101	0.142	4.230	0.128	4.420	0.137	4.164	0.133	**4.094**	0.244
Exactly2	6.422	0.669	7.028	0.603	6.574	0.682	7.131	0.384	**6.401**	0.435
Exactly	6.338	0.282	6.378	0.304	6.167	0.241	**5.948**	0.299	6.421	0.242
M-of-n	6.058	0.155	6.026	0.066	6.422	0.122	5.673	0.231	**5.606**	0.142
WineEW	4.236	0.109	4.135	0.048	4.129	0.053	**4.076**	0.075	4.157	0.048
CongressEW	4.714	0.096	4.471	0.158	4.582	0.213	4.753	0.088	**4.308**	0.138
Vote	4.401	0.042	4.366	0.092	4.275	0.059	**4.175**	0.116	4.230	0.162
Zoo	3.919	0.178	4.084	0.106	3.935	0.222	**3.707**	0.172	3.744	0.144
Lymphography	3.973	0.352	**3.941**	0.116	3.975	0.153	4.165	0.052	4.145	0.033
SpectEW	**3.912**	0.173	4.241	0.128	4.048	0.132	4.403	0.048	4.095	0.189
BreastEW	5.195	0.081	5.168	0.091	5.075	0.085	5.063	0.107	**4.994**	0.112
IonosphereEW	4.388	0.088	4.634	0.085	4.495	0.081	**4.350**	0.273	4.413	0.061
KrvskpEW	36.189	3.252	37.828	2.478	35.144	4.220	**35.124**	3.961	37.356	2.790
WaveformEW	94.514	9.550	93.414	8.396	92.964	10.179	93.403	8.502	**92.013**	7.604
SonarEW	4.246	0.036	4.245	0.037	4.285	0.070	4.240	0.039	**4.201**	0.066
clean1	6.652	0.160	6.684	0.262	**6.629**	0.176	6.701	0.164	6.720	0.238
semeion	51.893	3.184	52.008	5.155	**51.012**	2.599	53.120	4.209	51.890	3.519
PenglungEW	4.370	0.046	4.378	0.050	4.369	0.090	**4.315**	0.050	4.348	0.044
Colon	**4.940**	0.106	5.031	0.096	4.980	0.060	4.951	0.085	5.064	0.077
Leukemia	**12.052**	1.343	12.284	1.363	12.221	1.225	12.410	1.482	12.728	1.614

Table 7 The classification accuracy results obtained by binary βHC algorithm with varies β values

Dataset	Sen6 (β=0)		Sen7 (β=0.005)		Sen8 (β=0.05)		Sen9 (β=0.1)		Sen10 (β=0.5)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	0.777	0.011	0.771	0.008	0.776	0.006	0.800	0	**0.816**	0
Breastcancer	0.961	0.006	0.969	0.007	0.967	0.003	0.962	0.002	**0.974**	0.001
HeartEW	0.813	0.017	0.803	0.021	0.821	0.012	0.819	0.012	**0.836**	0.011
Exactly2	0.741	0.028	0.731	0.019	0.707	0.011	0.728	0.011	**0.750**	0.006
Exactly	0.950	0.122	0.885	0.161	0.967	0.100	0.820	0.167	**0.999**	0.003
M-of-n	**1.000**	0.001	**1.000**	0	**1.000**	0	**1.000**	0	**1.000**	0
WineEW	0.960	0.021	0.974	0.009	0.976	0.015	0.987	0.010	**0.996**	0.005
CongressEW	0.952	0.006	0.964	0.007	0.964	0.006	0.963	0.005	**0.974**	0.004
Vote	0.949	0.011	**0.978**	0.007	0.948	0.006	0.959	0.004	0.957	0.004
Zoo	0.995	0.009	0.996	0.008	0.947	0.009	0.962	0.029	**1.000**	0
Lymphography	0.837	0.030	0.853	0.036	0.886	0.029	**0.907**	0.029	0.877	0.014
SpectEW	0.820	0.015	**0.866**	0.026	0.854	0.013	0.860	0.015	0.859	0.009
BreastEW	0.954	0.006	0.954	0.010	0.960	0.007	**0.969**	0.006	0.964	0.005
IonosphereEW	0.921	0.014	0.910	0.014	0.928	0.013	**0.942**	0.012	0.937	0.008
KrvskpEW	0.974	0.008	0.981	0.005	**0.984**	0.003	0.976	0.002	0.964	0.003
WaveformEW	0.748	0.015	0.756	0.010	**0.772**	0.008	0.767	0.007	0.748	0.006
SonarEW	0.928	0.021	0.912	0.021	0.928	0.017	**0.949**	0.019	0.894	0.011
clean1	0.881	0.008	0.886	0.017	0.919	0.014	**0.941**	0.013	0.880	0.007
semeion	0.969	0.005	0.973	0.005	0.986	0.003	**0.987**	0.003	0.976	0.002
PenglungEW	0.820	0.033	0.769	0.058	0.873	0.025	**0.912**	0.019	0.781	0.019
Colon	0.673	0.030	0.756	0.029	0.810	0.021	0.763	0.028	**0.835**	0.010
Leukemia	0.793	0.028	0.856	0.029	**0.965**	0.012	0.856	0.011	0.903	0.014

Table 8 The fitness values obtained by binary βHC algorithm with varies β values

Dataset	Sen6 (β=0)		Sen7 (β=0.005)		Sen8 (β=0.05)		Sen9 (β=0.1)		Sen10 (β=0.5)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	0.227	0.011	0.233	0.009	0.228	0.006	0.205	0	**0.189**	0
Breastcancer	0.045	0.006	0.037	0.007	0.038	0.003	0.043	0.002	**0.030**	0.001
HeartEW	0.191	0.017	0.201	0.021	0.182	0.013	0.184	0.012	**0.167**	0.010
Exactly2	0.260	0.030	0.270	0.021	0.295	0.011	0.274	0.010	**0.249**	0.007
Exactly	0.054	0.121	0.117	0.158	0.037	0.099	0.181	0.164	**0.005**	0.003
M-of-n	**0.005**	0.002	**0.005**	0	**0.005**	0	**0.005**	0	**0.005**	0
WineEW	0.045	0.021	0.032	0.008	0.028	0.015	0.018	0.010	**0.009**	0.004
CongressEW	0.050	0.007	0.039	0.007	0.038	0.006	0.040	0.006	**0.029**	0.005
Vote	0.054	0.011	**0.024**	0.008	0.054	0.006	0.044	0.004	0.046	0.004
Zoo	0.008	0.009	0.008	0.008	0.056	0.009	0.042	0.028	**0.003**	0
Lymphography	0.166	0.029	0.150	0.036	0.130	0.029	**0.095**	0.029	0.126	0.015
SpectEW	0.182	0.015	**0.137**	0.026	0.148	0.013	0.143	0.014	0.145	0.008
BreastEW	0.050	0.006	0.049	0.010	0.043	0.007	**0.034**	0.007	0.041	0.005
IonosphereEW	0.082	0.014	0.093	0.014	0.074	0.014	**0.061**	0.012	0.067	0.008
KrvskpEW	0.031	0.008	0.023	0.006	**0.021**	0.003	0.029	0.002	0.041	0.004
WaveformEW	0.255	0.015	0.246	0.011	**0.230**	0.008	0.236	0.007	0.255	0.006
SonarEW	0.075	0.021	0.092	0.021	0.075	0.017	**0.055**	0.018	0.110	0.011
clean1	0.122	0.008	0.116	0.016	0.085	0.014	**0.063**	0.013	0.123	0.006
semeion	0.035	0.005	0.031	0.005	0.019	0.004	**0.018**	0.002	0.029	0.002
PenglungEW	0.182	0.033	0.231	0.057	0.129	0.025	**0.090**	0.019	0.221	0.019
Colon	0.329	0.030	0.245	0.028	0.192	0.020	0.239	0.028	**0.168**	0.010
Leukemia	0.210	0.027	0.147	0.029	**0.039**	0.012	0.148	0.011	0.101	0.014

Table 9 The selected features obtained by binary βHC algorithm with varies β values

Dataset	Sen6 (β=0)		Sen7 (β=0.005)		Sen8 (β=0.05)		Sen9 (β=0.1)		Sen10 (β=0.5)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	**5.350**	0.489	5.9	1.071	5.550	0.686	5.9	0.553	6.0	0.459
Breastcancer	5.3	1.129	5.150	0.587	4.650	1.089	4.6	1.231	**4.050**	0.394
HeartEW	7.350	1.631	6.6	1.635	**6.350**	2.254	6.6	1.903	6.550	1.932
Exactly2	4.5	3.220	5.350	3.360	6.950	2.438	5.6	2.583	**1.550**	2.438
Exactly	5.750	1.372	5.050	2.164	5.350	1.268	**3.8**	2.707	6.1	0.641
M-of-n	6.050	0.224	6.2	0.410	**5.850**	0.875	6.1	0.553	6.2	0.616
WineEW	6.8	1.196	7.350	1.565	**6.250**	1.482	7.0	1.892	6.8	2.285
CongressEW	4.150	2.110	6.0	2.596	**3.450**	2.328	4.850	3.133	4.8	2.142
Vote	6.350	1.843	4.7	2.408	4.7	2.473	**3.7**	2.557	4.7	1.129
Zoo	5.750	0.786	5.8	0.951	**5.350**	1.137	5.8	1.989	5.350	0.671
Lymphography	7.9	1.774	8.150	1.899	**6.7**	1.949	6.850	2.870	7.7	1.689
SpectEW	8.550	2.438	8.350	2.033	**8.350**	2.007	8.550	2.064	11.350	1.631
BreastEW	11.550	2.235	11.850	2.159	**11.050**	3.546	12.050	3.316	15.5	3.103
IonosphereEW	13.950	3.069	13.350	2.581	**10.350**	3.313	12.250	2.149	13.950	2.417
KrvskpEW	17.950	3.332	16.9	3.726	16.2	5.238	**15.7**	3.063	18.8	2.913
WaveformEW	20.550	3.379	21.1	3.538	19.9	3.144	**19.350**	2.681	22.050	3.034
SonarEW	24.550	3.546	23.750	3.508	25.3	3.840	**22.9**	3.878	28.9	2.989
clean1	71.4	6.969	**60.350**	7.365	71.850	7.043	77.050	5.763	78.250	7.973
semeion	123.350	7.876	**109.650**	7.686	123.4	8.075	128.850	7.184	128.450	9.423
PenglungEW	123.3	10.043	**85.250**	11.964	98.5	8.121	102.150	9.664	148.0	7.518
Colon	954.750	21.210	**740.9**	21.706	806.8	16.513	854.250	13.447	944.450	20.213
Leukemia	3522.150	42.794	**3092.6**	46.311	3191.550	49.618	3290.750	30.133	3473.8	38.030

Table 10 The CPU time (in Seconds) obtained by binary βHC algorithm with varies β values

Dataset	Sen6 (β=0)		Sen7 (β=0.005)		Sen8 (β=0.05)		Sen9 (β=0.1)		Sen10 (β=0.5)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	**5.666**	0.268	6.081	0.912	6.079	0.515	5.958	0.084	5.708	0.193
Breastcancer	5.318	0.178	4.951	0.093	**4.905**	0.103	4.907	0.197	4.934	0.075
HeartEW	4.345	0.087	4.202	0.232	**4.094**	0.244	4.295	0.080	4.304	0.061
Exactly2	6.358	1.134	6.218	0.834	6.401	0.435	6.349	0.701	**5.752**	0.392
Exactly	6.063	0.255	6.207	0.444	6.421	0.242	**5.783**	0.652	6.077	0.085
M-of-n	5.871	0.216	6.404	0.130	**5.606**	0.142	5.756	0.189	6.109	0.116
WineEW	4.065	0.153	4.255	0.103	4.157	0.048	**3.947**	0.300	4.126	0.052
CongressEW	4.671	0.116	4.646	0.291	**4.308**	0.138	4.444	0.221	4.311	0.300
Vote	4.323	0.057	4.341	0.069	4.230	0.162	3.962	0.204	**3.919**	0.312
Zoo	4.153	0.088	4.098	0.143	3.744	0.144	4.096	0.137	**3.650**	0.272
Lymphography	4.159	0.064	4.140	0.074	4.145	0.033	**3.968**	0.168	4.124	0.043
SpectEW	4.295	0.116	**3.954**	0.215	4.095	0.189	4.015	0.223	4.041	0.126
BreastEW	5.216	0.115	5.108	0.124	**4.994**	0.112	5.100	0.089	4.996	0.088
IonosphereEW	4.356	0.123	**4.348**	0.237	4.413	0.061	4.456	0.091	4.402	0.181
KrvskpEW	38.501	3.457	38.519	4.423	37.356	2.790	**36.151**	1.992	37.718	1.238
WaveformEW	92.639	12.676	96.081	9.831	92.013	7.604	**91.124**	5.148	93.915	3.411
SonarEW	4.235	0.067	4.258	0.075	**4.201**	0.066	4.263	0.074	4.261	0.061
clean1	6.774	0.228	**6.402**	0.220	6.720	0.238	6.678	0.226	6.959	0.144
semeion	52.388	3.390	**49.365**	2.466	51.890	3.519	52.972	2.795	53.574	1.450
PenglungEW	4.368	0.046	4.368	0.066	4.348	0.044	4.424	0.093	**4.286**	0.146
Colon	5.146	0.119	5.103	0.106	**5.064**	0.077	5.082	0.121	5.181	0.108
Leukemia	13.183	1.799	12.839	1.789	**12.728**	1.614	12.808	1.712	13.860	1.993

3.2.3 Study the effect of the different transfer functions

The the proposed binary βHC algorithm performance when using various transfer functions is studied in this section. Eight experimental scenarios (i.e., Sen11–Sen18) are tailored with different eight transfer functions (i.e., S1–S4 as a different versions of S-Shaped, and V1–V4 as a different versions of V-Shaped). These transfer functions are borrowed from (Mafarja et al. 2018a). The value of the parameter $\mathcal{N}$ is set to 0.9 and the parameter β is set to 0.5 based on the previous experiments conducted in sects. 3.2.1 and 3.2.2. Tables 11, 12, 13, and 14 illustrate the mean and the standard deviation of the results by running Sen11 to Sen18 in terms of the classification accuracy, the fitness value, the selected features, and the elapsed CPU time. The **bold** font is used to point up the best outcomes.

Table 11 shows the influence of using different transfer functions on the performance of the proposed binary βHC algorithm in terms of the classification accuracy. Table 11 demonstrate that the performance of the proposed algorithm using Sen12 outperforms the seven other scenarios (Sen11, and Sen13–Sen18) as it obtained the finest outcomes on 7 datasets. In addition, Sen11, Sen13, and Sen16 have the best results on 3 datasets. While Sen14, Sen15, and Sen17 obtained the finest outcomes on 2 datasets. Finally, Sen18 achieved better results than other scenarios in one dataset. Table 11 shows the robustness of Sen11 to Sen18 based on the standard derivations of the results. It is not clear which experimental scenario to pick, as all scenarios achieved almost similar results.

Similarly, Table 12 demonstrates the proposed binary βHC algorithm efficiency in terms of the fitness value by utilizing various transfer functions. Clearly, Sen12 outperforms other scenarios as it obtained the finest outcomes on 6 datasets. In addition, Sen11, Sen13, and Sen16 achieved the best results on 3 datasets. While Sen15, Sen17, and Sen18 obtained the best results on 2 datasets. Finally, Sen14 achieved the finest outcomes on one dataset. Based on the standard derivation results recorded in Table 12, it can be observed that Sen11 to Sen18 obtained almost similar standard derivations results. This reflects the robustness of the proposed binary βHC algorithm in all cases.

Table 13 summarizes the results of running the experimental scenarios from Sen11 to Sen18 in terms of the selected features. Clearly, Sen11 to Sen14 which study different versions of the S-Shaped transfer function did not obtain any of the best results. On the other hand, Sen16 achieved the best results in 7 datasets. While the scenarios Sen15, Sen17, and Sen18 get the best results in 5 datasets. In conclusion, the performance of the proposed binary βHC algorithm in terms of the selected features using the

Table 11 The classification accuracy results obtained by binary βHC algorithm with varies transfer functions

Dataset		Sen11 S1	Sen12 S2	Sen13 S3	Sen14 S4	Sen15 V1	Sen16 V2	Sen17 V3	Sen18 V4
Tic-tac-toe	Avg	0.795	**0.816**	0.808	0.799	0.776	0.788	0.802	0.776
	Stdv	0.005	0	0	0.002	0.010	0.007	0.003	0.012
Breastcancer	Avg	**0.985**	0.974	0.973	0.977	0.979	0.970	0.966	0.975
	Stdv	0.001	0.001	0.002	0.001	0.002	0.002	0	0.003
HeartEW	Avg	0.860	0.836	0.827	**0.870**	0.859	0.843	0.844	0.838
	Stdv	0.006	0.011	0.008	0.004	0.008	0.006	0.007	0.004
Exactly2	Avg	0.760	0.750	**0.774**	0.766	0.766	0.750	0.766	0.767
	Stdv	0.005	0.006	0.001	0.007	0	0.005	0.003	0.002
Exactly	Avg	0.817	**0.999**	0.902	0.818	0.731	0.735	0.757	0.847
	Stdv	0.062	0.003	0.064	0.090	0.066	0.075	0.099	0.123
M-of-n	Avg	0.965	**1.000**	0.967	0.957	0.914	0.902	0.940	0.969
	Stdv	0.024	0	0.025	0.032	0.057	0.057	0.054	0.052
WineEW	Avg	0.978	0.996	0.988	**1.000**	0.959	**1.000**	0.971	0.971
	Stdv	0	0.003	0.005	0	0.005	0	0.009	0.008
CongressEW	Avg	0.966	0.974	0.959	0.963	0.962	0.973	**0.977**	0.972
	Stdv	0.005	0.004	0.004	0.005	0.004	0.004	0.002	0.001
Vote	Avg	0.957	0.957	0.943	0.962	0.939	0.958	**0.975**	0.963
	Stdv	0.005	0.004	0.008	0.004	0.003	0.004	0.003	0.005
Zoo	Avg	0.886	**1.000**	0.989	0.786	0.909	0.844	0.971	0.929
	Stdv	0.023	0.016	0.012	0.006	0.020	0.008	0.010	0.018
Lymphography	Avg	0.836	0.877	**0.913**	0.878	0.868	0.849	0.905	0.870
	Stdv	0.013	0.014	0.011	0.015	0.006	0.014	0.021	0.011
SpectEW	Avg	0.851	0.859	**0.869**	0.835	0.841	0.868	0.846	0.852
	Stdv	0.012	0.009	0.006	0.007	0.009	0.005	0.007	0.008
BreastEW	Avg	**0.979**	0.964	0.946	0.962	0.953	0.969	0.959	0.964
	Stdv	0.003	0.005	0.003	0.004	0.004	0.004	0.005	0.003
IonosphereEW	Avg	0.861	**0.937**	0.895	0.895	0.918	0.901	0.902	0.907
	Stdv	0.007	0.008	0.004	0.005	0.008	0.006	0.010	0.008
KrvskpEW	Avg	0.957	**0.964**	0.954	0.961	0.950	0.947	0.956	0.953
	Stdv	0.002	0.003	0.006	0.005	0.006	0.009	0.010	0.010
WaveformEW	Avg	**0.789**	0.748	0.779	0.786	0.771	0.775	0.771	0.777
	Stdv	0.003	0.006	0.006	0.006	0.008	0.008	0.008	0.009
SonarEW	Avg	0.806	**0.894**	0.843	0.855	0.850	0.838	0.823	0.868
	Stdv	0.012	0.011	0.012	0.012	0.011	0.012	0.016	0.015
clean1	Avg	0.880	0.880	0.882	0.875	0.879	**0.895**	0.881	0.868
	Stdv	0.005	0.007	0.006	0.008	0.006	0.008	0.005	0.008
semeion	Avg	0.971	0.976	0.965	0.977	**0.978**	0.975	0.975	0.968
	Stdv	0.001	0.002	0.002	0.002	0.003	0.002	0.002	0.002
PenglungEW	Avg	0.773	0.781	0.869	0.818	0.888	**0.905**	0.882	0.827
	Stdv	0.014	0.019	0.010	0.012	0.018	0.019	0.013	0.014
Colon	Avg	0.805	0.0.835	0.710	0.789	**0.842**	0.787	0.840	0.803
	Stdv	0.007	0.010	0	0.016	0.014	0.016	0.016	0.010
Leukemia	Avg	0.861	0.903	0.883	0.836	0.967	0.915	0.828	**0.972**
	Stdv	0.009	0.014	0.011	0.009	0.011	0.006	0.011	0

V-Shaped transfer function achieved superior outcomes than the S-Shaped transfer function for all cases.

Finally, the results of running Sen11 to Se18 in terms of the elapsed CPU time are summarized in Table 14. Apparently, the performance of Sen16 outperforms the other scenarios as it obtained the minimum of the average CPU time on 9 datasets. Furthermore, Sen15 gets the minimum of the average CPU time on 5 datasets. While Sen12 and

Table 12 The fitness values obtained by binary βHC algorithm with varies transfer functions

Dataset		Sen11 S1	Sen12 S2	Sen13 S3	Sen14 S4	Sen15 V1	Sen16 V2	Sen17 V3	Sen18 V4
Tic-tac-toe	Avg	0.209	**0.189**	0.200	0.209	0.227	0.216	0.202	0.229
	Stdv	0.005	0	0	0.001	0.009	0.007	0.003	0.010
Breastcancer	Avg	**0.020**	0.030	0.031	0.030	0.025	0.035	0.037	0.030
	Stdv	0.002	0.001	0.002	0.001	0.001	0.002	0	0.003
HeartEW	Avg	0.145	0.167	0.177	**0.136**	0.144	0.160	0.159	0.163
	Stdv	0.006	0.010	0.008	0.004	0.008	0.006	0.006	0.003
Exactly2	Avg	0.245	0.249	**0.225**	0.234	0.232	0.249	0.233	0.232
	Stdv	0.006	0.007	0.001	0.004	0	0.003	0.002	0.001
Exactly	Avg	0.187	**0.005**	0.102	0.187	0.270	0.266	0.245	0.157
	Stdv	0.062	0.003	0.064	0.090	0.064	0.073	0.097	0.121
M-of-n	Avg	0.040	**0.005**	0.039	0.048	0.091	0.102	0.065	0.035
	Stdv	0.024	0	0.025	0.033	0.056	0.057	0.053	0.051
WineEW	Avg	0.029	0.009	0.017	0.005	0.044	**0.003**	0.033	0.033
	Stdv	0.001	0.004	0.007	0.001	0.005	0.001	0.009	0.007
CongressEW	Avg	0.040	0.029	0.044	0.042	0.039	0.029	**0.025**	0.029
	Stdv	0.005	0.005	0.004	0.005	0.003	0.004	0.001	0.001
Vote	Avg	0.050	0.046	0.060	0.042	0.062	0.044	**0.027**	0.040
	Stdv	0.005	0.004	0.008	0.004	0.003	0.004	0.003	0.005
Zoo	Avg	0.119	**0.003**	0.016	0.216	0.094	0.157	0.033	0.073
	Stdv	0.022	0	0.012	0.006	0.020	0.008	0.009	0.018
Lymphography	Avg	0.169	0.126	**0.092**	0.126	0.133	0.153	0.098	0.132
	Stdv	0.012	0.015	0.011	0.014	0.005	0.013	0.021	0.011
SpectEW	Avg	0.153	0.145	**0.134**	0.168	0.160	0.135	0.155	0.150
	Stdv	0.013	0.008	0.006	0.007	0.009	0.005	0.007	0.008
BreastEW	Avg	**0.029**	0.041	0.059	0.043	0.050	0.035	0.044	0.039
	Stdv	0.002	0.005	0.003	0.004	0.003	0.003	0.004	0.003
IonosphereEW	Avg	0.144	**0.067**	0.110	0.109	0.083	0.100	0.099	0.094
	Stdv	0.007	0.008	0.004	0.005	0.008	0.006	0.010	0.008
KrvskpEW	Avg	0.049	**0.041**	0.052	0.045	0.053	0.056	0.047	0.050
	Stdv	0.002	0.004	0.005	0.005	0.005	0.009	0.010	0.009
WaveformEW	Avg	**0.217**	0.255	0.224	0.218	0.230	0.227	0.231	0.224
	Stdv	0.003	0.006	0.006	0.006	0.008	0.008	0.008	0.009
SonarEW	Avg	0.199	0.110	0.161	0.148	0.151	0.163	0.178	**0.133**
	Stdv	0.012	0.011	0.012	0.012	0.011	0.012	0.016	0.014
clean1	Avg	0.127	0.123	0.123	0.130	0.123	**0.106**	0.121	0.134
	Stdv	0.005	0.006	0.006	0.007	0.006	0.008	0.005	0.008
semeion	Avg	0.036	0.029	0.041	0.028	**0.024**	0.028	0.028	0.035
	Stdv	0.001	0.002	0.002	0.002	0.003	0.002	0.002	0.002
PenglungEW	Avg	0.231	0.221	0.135	0.186	0.114	**0.096**	0.119	0.174
	Stdv	0.013	0.019	0.010	0.012	0.018	0.018	0.013	0.013
Colon	Avg	0.200	0.168	0.293	0.214	**0.159**	0.213	0.161	0.197
	Stdv	0.007	0.010	0	0.016	0.014	0.016	0.016	0.010
Leukemia	Avg	0.145	0.101	0.121	0.167	0.036	0.086	0.173	**0.030**
	Stdv	0.009	0.014	0.011	0.008	0.011	0.006	0.011	0

Sen17 achieved the minimum of the average CPU time on 4 datasets. Also, Sen14 and Sen18 obtained the minimum CPU time on one dataset. Furthermore, Sen11 and Sen13 did not obtain any of the best results in terms of CPU time.

Based on the above discussions, the performance of the proposed algorithm using the S-Shaped transfer function is better than the V-Shaped transfer function in terms of the classification accuracy and the fitness values. On the

Table 13 The selected features obtained by binary βHC algorithm with varies transfer functions

Dataset		Sen11 S1	Sen12 S2	Sen13 S3	Sen14 S4	Sen15 V1	Sen16 V2	Sen17 V3	Sen18 V4
Tic-tac-toe	Avg	6.0	6.0	9.0	8.7	**4.7**	5.050	5.0	5.9
	Stdv	0	0.459	0	0.733	1.174	0.510	0	1.210
Breastcancer	Avg	4.7	4.050	3.9	5.950	4.2	5.1	**3.0**	5.3
	Stdv	1.031	0.394	0.553	0.224	1.152	0.447	0	0.470
HeartEW	Avg	8.1	6.550	7.650	9.6	5.6	6.2	4.9	**3.8**
	Stdv	1.071	1.932	1.814	1.759	1.698	1.609	1.021	1.399
Exactly2	Avg	9.9	1.550	2.7	3.0	**1.0**	2.0	2.050	1.750
	Stdv	1.832	2.438	1.081	3.449	0	2.052	1.669	1.860
Exactly	Avg	7.8	6.1	7.2	7.9	4.3	**4.150**	5.5	5.950
	Stdv	0.696	0.641	0.696	1.119	2.993	2.560	2.544	1.877
M-of-n	Avg	7.350	6.2	7.450	7.450	6.850	6.5	7.1	**6.0**
	Stdv	0.933	0.616	0.686	1.191	1.268	1.433	0.968	1.026
WineEW	Avg	9.050	6.8	5.550	6.250	4.450	**3.7**	5.2	6.050
	Stdv	1.099	2.285	0.945	1.118	1.504	1.261	0.894	0.999
CongressEW	Avg	9.2	4.8	5.450	7.6	3.150	3.650	3.5	**3.050**
	Stdv	0.834	2.142	1.638	1.875	2.254	0.933	0.827	0.394
Vote	Avg	10.750	4.7	6.2	6.650	**2.4**	4.450	3.9	4.4
	Stdv	1.164	1.129	1.881	1.872	0.681	1.761	0.718	1.353
Zoo	Avg	10.750	5.350	8.3	6.550	5.2	**4.2**	6.3	5.4
	Stdv	1.020	0.671	1.302	1.356	0.834	0.410	1.490	0.681
Lymphography	Avg	12.1	7.7	9.550	9.350	**4.250**	6.8	6.3	6.650
	Stdv	1.294	1.689	1.791	1.899	1.293	1.473	1.218	1.309
SpectEW	Avg	12.7	11.350	9.950	9.7	6.4	7.450	**6.2**	7.2
	Stdv	1.780	1.631	1.572	2.203	2.162	1.731	1.642	1.322
BreastEW	Avg	22.8	15.5	16.2	15.850	10.6	11.0	**9.850**	10.5
	Stdv	1.881	3.103	2.093	2.681	2.088	2.596	2.621	2.090
IonosphereEW	Avg	21.6	13.950	17.550	15.250	6.350	6.150	6.1	**5.150**
	Stdv	2.741	2.417	3.034	2.9	2.159	1.899	1.683	1.631
KrvskpEW	Avg	24.7	18.8	21.850	21.550	12.850	**11.6**	12.350	14.0
	Stdv	2.364	2.913	3.133	1.959	2.323	1.789	2.519	1.919
WaveformEW	Avg	29.4	22.050	24.250	24.6	14.750	**14.7**	14.9	14.950
	Stdv	2.393	3.034	2.954	1.818	2.291	2.515	2.713	2.605
SonarEW	Avg	41.1	28.9	31.650	30.150	17.550	16.350	**15.850**	17.050
	Stdv	3.401	2.989	3.407	4.043	2.946	2.477	3.150	2.665
clean1	Avg	121.650	78.250	90.8	93.550	45.550	45.4	**42.5**	44.5
	Stdv	4.392	7.973	6.486	5.735	5.104	4.430	5.960	5.405
semeion	Avg	194.750	128.450	146.450	142.650	75.550	**72.4**	73.8	74.5
	Stdv	6.365	9.423	7.060	10.194	6.395	8.457	5.970	7.302
PenglungEW	Avg	212.450	148.0	174.850	168.250	**80.450**	80.650	80.950	80.5
	Stdv	24.479	7.518	7.250	7.759	9.064	8.324	7.126	6.613
Colon	Avg	1443.0	944.450	1110.050	1061.250	513.250	**503.3**	523.650	529.550
	Stdv	45.473	20.213	16.002	23.402	19.732	17.125	17.024	19.351
Leukemia	Avg	5059.8	3473.8	3956.5	3600.3	1858.650	1843.2	1872.950	**1833.8**
	Stdv	367.432	38.030	92.789	107.594	33.078	37.769	39.409	17.225

other hand, the performance of the proposed algorithm using the V-Shaped transfer function is superior than the S-Shaped transfer function in terms of the selected attributes and the CPU time. However, it is hard to decide which transfer function is better for the feature selection problem. We decide to use the S-Shaped (S3) transfer function in the subsequent experimentation, as the

Table 14 The CPU time (in seconds) obtained by binary βHC algorithm with varies transfer functions

Dataset		Sen11 S1	Sen12 S2	Sen13 S3	Sen14 S4	Sen15 V1	Sen16 V2	Sen17 V3	Sen18 V4
Tic-tac-toe	Avg	7.375	5.708	6.308	6.125	**4.890**	5.164	5.138	5.556
	Stdv	0.198	0.193	0.265	0.433	0.312	0.406	0.195	0.356
Breastcancer	Avg	5.795	4.934	4.648	4.827	4.745	**4.296**	4.822	4.482
	Stdv	0.223	0.075	0.166	0.304	0.306	0.277	0.188	0.157
HeartEW	Avg	3.862	4.304	3.8	3.679	3.635	**3.522**	3.661	3.602
	Stdv	0.22	0.061	0.149	0.074	0.11	0.06	0.267	0.122
Exactly2	Avg	8.19	5.752	6.882	6.949	5.526	5.248	**5.159**	5.935
	Stdv	0.456	0.392	0.287	0.228	0.284	0.222	0.161	0.580
Exactly	Avg	8.096	6.077	6.962	7.286	5.858	5.329	**5.185**	5.974
	Stdv	0.298	0.085	0.226	0.461	0.569	0.19	0.266	0.448
M-of-n	Avg	8.197	6.109	6.775	6.826	5.71	**5.506**	5.84	5.917
	Stdv	0.433	0.116	0.121	0.316	0.209	0.156	0.178	0.305
WineEW	Avg	3.749	4.126	3.768	**3.485**	3.736	3.636	3.914	3.698
	Stdv	0.144	0.052	0.398	0.158	0.344	0.062	0.308	0.192
CongressEW	Avg	4.525	4.311	4.547	4.573	4.185	4.156	**4.016**	4.564
	Stdv	0.184	0.3	0.285	0.488	0.193	0.139	0.108	0.201
Vote	Avg	3.962	3.919	4.273	3.969	**3.701**	3.792	3.899	3.858
	Stdv	0.203	0.312	0.207	0.31	0.119	0.282	0.331	0.201
Zoo	Avg	3.923	**3.65**	3.704	3.866	4.031	3.824	3.793	3.902
	Stdv	0.156	0.272	0.082	0.245	0.506	0.171	0.196	0.178
Lymphography	Avg	3.76	4.124	3.659	3.582	3.721	**3.513**	4.012	3.726
	Stdv	0.316	0.043	0.322	0.098	0.26	0.073	0.304	0.092
SpectEW	Avg	3.762	4.041	3.818	3.744	**3.733**	3.765	3.739	3.738
	Stdv	0.127	0.126	0.145	0.234	0.085	0.198	0.174	0.112
BreastEW	Avg	5.639	4.996	5.007	4.919	5.175	**4.775**	5.509	5.268
	Stdv	0.366	0.088	0.208	0.208	0.222	0.18	0.244	0.394
IonosphereEW	Avg	4.318	4.402	4.186	4.134	4.469	4.22	4.058	**3.986**
	Stdv	0.233	0.181	0.226	0.266	0.352	0.225	0.158	0.081
KrvskpEW	Avg	54.578	37.718	43.131	42.369	32.235	**31.170**	32.034	34.695
	Stdv	1.344	1.238	0.817	0.759	1.087	1.071	1.165	1.179
WaveformEW	Avg	139.126	93.915	109.977	106.688	83.706	**82.663**	85.529	87.815
	Stdv	2.37	3.411	1.023	1.568	1.172	1.684	2.254	2.637
SonarEW	Avg	4.116	4.261	3.974	4.177	4.116	3.714	**3.564**	3.766
	Stdv	0.258	0.061	0.272	0.297	0.189	0.131	0.1	0.236
clean1	Avg	9.564	6.959	7.806	7.583	6.315	**5.636**	6.075	5.984
	Stdv	0.559	0.144	0.376	0.289	0.186	0.105	0.307	0.305
semeion	Avg	88.636	53.574	64.733	62.797	35.247	**33.753**	34.289	36.491
	Stdv	1.485	1.45	0.815	0.772	0.86	0.694	0.884	1.696
PenglungEW	Avg	5.143	**4.286**	5.204	5.343	6.238	5.172	5.119	5.067
	Stdv	0.273	0.146	0.213	0.435	0.44	0.224	0.453	0.349
Colon	Avg	13.162	**5.181**	12.026	11.998	16.503	11.325	8.486	8.457
	Stdv	0.599	0.108	0.496	0.515	0.672	0.486	0.148	0.250
Leukemia	Avg	38.952	**13.86**	36.163	34.945	51.453	32.187	22.233	22.727
	Stdv	0.928	1.993	1.278	1.03	2.639	1.404	0.264	0.816

performance of the proposed algorithm using S-Shaped (S3) is superior than others in terms of classification accuracy.

3.2.4 Study the effect of the different classifiers

In this section, the influence of using three distinct classifiers (i.e., kNN, SVM, and decision tree (DT)) on the

Table 15 The classification accuracy results obtained by binary βHC algorithm with different classifiers

Dataset	Sen19 (kNN)		Sen20 (SVM)		Sen21 (DT)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	**0.816**	0	0.616	0.029	0.739	0
Breastcancer	**0.974**	0.001	0.961	0.003	0.971	0.002
HeartEW	**0.836**	0.011	0.809	0.02	0.829	0.02
Exactly2	**0.75**	0.006	0.74	0.058	0.701	0.057
Exactly	**0.999**	0.003	0.594	0.078	0.852	0.089
M-of-n	**1.000**	0	0.943	0.033	0.956	0.029
WineEW	**0.996**	0.003	0.967	0.014	0.881	0.014
CongressEW	**0.974**	0.004	0.916	0.011	0.938	0.015
Vote	**0.957**	0.004	0.93	0.011	0.902	0.018
Zoo	**1.000**	0.016	0.876	0.035	0.767	0.051
Lymphography	**0.877**	0.014	0.767	0.044	0.699	0.046
SpectEW	**0.859**	0.009	0.836	0.023	0.807	0.027
BreastEW	**0.964**	0.005	0.933	0.009	0.95	0.011
IonosphereEW	**0.937**	0.008	0.797	0.015	0.816	0.014
KrvskpEW	**0.964**	0.003	0.934	0.011	0.942	0.007
WaveformEW	0.748	0.006	**0.777**	0.006	0.769	0.011
SonarEW	**0.894**	0.011	0.73	0.026	0.726	0.024
clean1	**0.88**	0.007	0.801	0.025	0.832	0.02
semeion	**0.976**	0.002	0.953	0.005	0.958	0.004
PenglungEW	0.781	0.019	**0.884**	0.023	0.742	0.016
Colon	**0.835**	0.01	0.826	0.028	0.752	0.018
Leukemia	**0.903**	0.014	0.858	0.022	0.819	0.014

Table 16 The fitness values obtained by binary βHC algorithm with different classifiers

Dataset	Sen19 (kNN)		Sen20 (SVM)		Sen21 (DT)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	0.189	0	0.365	0	**0.171**	0
Breastcancer	0.03	0.001	0.03	0.001	**0.027**	0.001
HeartEW	0.167	0.01	0.165	0.004	**0.162**	0.005
Exactly2	0.249	0.007	**0.225**	0.001	0.24	0.003
Exactly	**0.005**	0.003	0.291	0.001	0.107	0.034
M-of-n	**0.005**	0	0.006	0.001	0.016	0.001
WineEW	0.009	0.004	**0.006**	0.001	0.031	0.005
CongressEW	**0.029**	0.005	0.054	0.002	0.036	0.001
Vote	0.046	0.004	0.049	0.003	**0.039**	0.001
Zoo	**0.003**	0	0.006	0.001	0.14	0.015
Lymphography	**0.126**	0.015	0.134	0.008	0.169	0.006
SpectEW	0.145	0.008	**0.121**	0.006	0.135	0.011
BreastEW	0.041	0.005	0.041	0.002	**0.039**	0.004
IonosphereEW	0.067	0.008	0.111	0.008	**0.039**	0
KrvskpEW	0.041	0.004	0.054	0.003	**0.022**	0.001
WaveformEW	0.255	0.006	**0.169**	0.002	0.247	0.003
SonarEW	**0.11**	0.011	0.168	0.008	0.194	0.022
clean1	**0.123**	0.006	0.155	0.008	0.178	0.01
semeion	0.029	0.002	**0.027**	0.002	0.052	0.003
PenglungEW	0.221	0.019	**0.032**	0	0.419	0.024
Colon	0.168	0.01	0.166	0	**0.146**	0.016
Leukemia	0.101	0.014	**0.005**	0	0.112	0.013

Table 17 The selected attributes obtained by binary βHC algorithm with different classifiers

Dataset	Sen19 (kNN)		Sen20 (SVM)		Sen21 (DT)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	6	0	**1.0**	0	7	0
Breastcancer	4.05	0.394	4.4	0.598	**3.850**	0.366
HeartEW	6.55	1.932	9.5	1.504	**5.950**	0.826
Exactly2	**1.550**	2.438	2.1	0.788	3.25	1.713
Exactly	6.1	0.641	**2.2**	0.696	7.5	0.827
M-of-n	**6.2**	0	7.45	0.759	7.4	0.681
WineEW	6.8	2.285	7.3	0.865	**5.9**	1.165
CongressEW	**4.80**	2.142	8.85	1.268	6.4	0.821
Vote	**4.7**	1.129	9.95	1.504	8.9	0.968
Zoo	**5.350**	0.671	9.55	1.276	7.85	1.137
Lymphography	**7.7**	1.689	10.7	1.949	9.9	1.41
SpectEW	**11.350**	1.631	13.8	1.704	11.7	1.525
BreastEW	**15.5**	3.103	18.1	2.1	16.9	1.683
IonosphereEW	**13.950**	2.417	20.65	2.7	18.85	1.531
KrvskpEW	**18.8**	2.913	23.05	2.502	27.6	2.722
WaveformEW	**22.050**	3.034	28.3	2.43	26.3	2.319
SonarEW	**28.9**	2.989	35.8	3.381	34.65	4.716
clean1	**78.250**	7.973	109.55	5.735	98.9	5.884
semeion	**128.450**	9.423	165.1	9.952	166.05	6.074
PenglungEW	**148**	7.518	185.35	8.4	193.55	8.262
Colon	**944.450**	20.213	1211.55	57.746	1179.3	100.034
Leukemia	**3473.8**	38.03	3635.8	255.435	4397.3	213.633

Table 18 The CPU times (in Seconds) obtained by binary βHC algorithm with different classifiers

Dataset	Sen19 (kNN)		Sen20 (SVM)		Sen21 (DT)	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	**5.708**	0	54.244	0.964	11.816	0
Breastcancer	**4.934**	0.075	60.109	2.003	13.048	0.549
HeartEW	**4.304**	0.061	30.271	0.692	9.96	0.255
Exactly2	**5.752**	0.392	52.767	0.789	11.466	0.532
Exactly	**6.077**	0.085	53.485	0.917	11.711	0.709
M-of-n	**6.109**	0	54.205	1.101	11.068	0.408
WineEW	**4.126**	0.052	43.886	0.836	9.872	0.363
CongressEW	**4.311**	0.3	36.289	1.114	10.115	0.466
Vote	**3.919**	0.312	31.526	0.632	9.997	0.516
Zoo	**3.650**	0.272	186.012	2.094	9.271	0.312
Lymphography	**4.124**	0.043	42.778	0.941	10.051	0.373
SpectEW	**4.041**	0.126	30.592	0.865	10.141	0.511
BreastEW	**4.996**	0.088	43.81	1.107	11.182	0.357
IonosphereEW	**4.402**	0.181	44.402	1.088	11.462	0.676
KrvskpEW	37.718	1.238	177.604	4.812	**18.673**	0.504
WaveformEW	93.915	3.411	1648.72	33.8	**45.895**	0.971
SonarEW	**4.261**	0.061	36.831	1.095	11.018	0.674
clean1	**6.959**	0.144	338.619	24.538	32.977	1.098
semeion	53.574	1.45	101.036	1.189	**32.017**	0.655
PenglungEW	**4.286**	0.146	192.841	2.804	11.433	0.354
Colon	**5.181**	0.108	30.472	0.947	20.535	0.584
Leukemia	**13.860**	1.993	56.946	2.552	52.397	3.414

performance of the proposed binary βHC algorithm for FS problem is studied. Three experimental scenarios Sen19 to Sen21 are designed, where each scenario utilized to study one of the three classifiers. The average (Avg) and the standard derivation (Stdv) are summarized based on the classification accuracy, the fitness value, the selected features, and the elapsed CPU time in Tables 15, 16, 17, and 18. The **bold** font is used to point out the finest outcomes.

Table 16 summarizes the experimental outcomes of investigating the effect of using three distinct classifiers on the performance of the proposed binary βHC algorithm in terms of the classification accuracy. The performance of the proposed algorithm using the kNN classifier (Sen19) outperforms the two other classifiers as it realized the finest outcomes on 20 out of 22 datasets. Furthermore, the proposed algorithm using the SVM classifier achieved the best results on two datasets. While the proposed algorithm with decision tree classifier did not get any of the best results for any dataset. Furthermore, it can be seen that the proposed algorithm with the kNN classifier is more robust than the other versions of this algorithm as it achieves almost the same results over 20 times of run as the results of the standard derivation show.

The impact of using three different classifiers on the performance of the proposed binary βHC algorithm in terms of the fitness values is summarized in Table 16. As the table shows, it is hard to decide which classifier is better to integrate within the proposed algorithm for tackling the problem of feature selection. This is because the performance of the proposed algorithm with decision tree classifier outperforms the performance of the other two classifiers on eight datasets. While the proposed algorithm efficiency when compared with the two other classifiers is almost the same as both were able to realize the finest outcomes for seven datasets.

Table 16 shows the results for Sen19 to Sen21 in terms of the selected features. Table 16 demonstrates that the results produced by Sen19 outperform those produced by other scenarios (Sen20 and Sen21) as it obtained the minimum number of the selected features on 17 out of 22 datasets. Furthermore, the experimental scenarios of Sen20 and Sen21 get the minimum number of the features in 2 and 3 datasets, respectively. This proves that the integration of the proposed binary βHC algorithm and the kNN classifier is better than the other two classifiers for the problem of feature selection in terms of the number of the selected features.

Finally, the results of the elapsed CPU time are investigated from Sen19 to Sen21, and it is recorded in Table 18. In this table, we can see that Sen19 obtained the minimum CPU time on 19 out of 22 datasets. While Sen21 achieved the minimum CPU time on 3 datasets. In addition, it can be seen that Sen20 is slower than the other scenarios (i.e., Sen19, and Sen21) as it could not obtain any minimum CPU time for any dataset.

To wrap-up, the kNN as a classifier is capable to obtain superior outcomes than the other two classifiers in almost all cases. As a result, the kNN classifier will be used in the following experiments.

3.2.5 The effect of training/testing against k-fold cross validation models

In this section, we investigate the effect of the used data splitting techniques (i.e., training/testing against k-fold cross-validation) on the convergence behaviour of the proposed method. The evaluation is conducted using kNN with training/testing (binary βHC(kNN-TT)) and using kNN with k-fold (binary βHC(kNN-KF)) cross-validation. In k-fold we divide a set of features data into complementary subsets, performing the analysis on one subset (i.e., training set), and validating the outcomes using the other subset (i.e., validation set or testing set) Delen et al. (2005). This technique is used for feature selection methods by other researchers Shao et al. (2013); Zhang et al. (2014). Note that the value of $k = 10$ is used in the k-fold cross-validation method. Again, four main measurements are used in the comparison which are classification accuracy, fitness function value, number of informative features, and the CPU time. Each experiment is replicated using 10 runs and the average (Avg) and standard deviation (Stdv) are recorded. The best Avg is highlighted in **bold** (lowest is best).

As can be noticed from Table 19, the binary βHC(kNN-TT) can outperform the binary βHC(kNN-KF) in 17 out of 22 datasets in terms of classification accuracy, and 18 out of 22 datasets in terms of the number of features reduced. However, βHC(kNN-KF) outperforms the binary βHC(kNN-TT) in 16 out of 22 datasets in terms of fitness function values. Furthermore, it excels in all datasets in terms of CPU time required.

In conclusion, binary βHC(kNN-TT) is a powerful algorithm in terms of classification accuracy and the number of features reduced. Therefore, it is recommended to use binary βHC(kNN-TT) for feature selection problems.

3.3 Comparison with local search method

In order to investigate the effectiveness of the proposed binary βHC algorithm for the feature selection problem,the algorithm is compared against other popular local search-based algorithms in this section. The comparative methods are hill climbing (HC), simulated annealing (SA), and variable neighborhood search (VNS). It should be noted that these algorithms are executed under the same conditions of the proposed algorithm in order to ensure fairness. Table 20 shows the parameter settings of these algorithms.

Table 19 The performance of binary βHC(kNN-TT) against binary βHC(kNN-KF)

Dataset		Accuracy		Fitness function		No. features		CPU time	
		Sen22	Sen23	Sen22	Sen23	Sen22	Sen23	Sen22	Sen23
	βHC	kNN-TT	kNN-KF	kNN-TT	kNN-KF	kNN-TT	kNN-KF	kNN-TT	kNN-KF
Tic-tac-toe	Avg	**0.816**	0.785	0.189	**0.131**	6	**5.7**	5.708	**1.104**
	Stdv	0	0.021	0	0.011	0.459	0.733	0.193	0.065
Breastcancer	Avg	**0.974**	0.967	0.030	**0.020**	**4.050**	4.5	4.934	**1.254**
	Stdv	0.001	0.007	0.001	0.003	0.394	0.827	0.075	0.045
HeartEW	Avg	**0.836**	0.709	**0.167**	0.215	**6.550**	7.55	4.304	**0.913**
	Stdv	0.011	0.050	0.010	0.027	1.932	1.099	0.061	0.035
Exactly2	Avg	**0.75**	0.7	0.249	**0.245**	**1.550**	8.6	5.752	**1.180**
	Stdv	0.006	0.045	0.007	0.019	2.438	1.635	0.392	0.084
Exactly	Avg	**0.999**	0.850	**0.005**	0.042	**6.1**	8.6	6.077	**1.174**
	Stdv	0.003	0.076	0.003	0.029	0.641	0.754	0.085	0.070
M-of-n	Avg	**1**	0.971	**0.005**	0.028	**6.2**	8.6	6.109	**1.167**
	Stdv	0	0.028	0	0.023	0.616	0.883	0.116	0.088
WineEW	Avg	**0.996**	0.935	0.009	**0.005**	6.8	**5.9**	4.126	**0.826**
	Stdv	0.003	0.046	0.004	0.001	2.285	1.021	0.052	0.052
CongressEW	Avg	**0.974**	0.913	0.029	**0.026**	**4.8**	8.15	4.311	**0.880**
	Stdv	0.004	0.020	0.005	0.014	2.142	1.309	0.3	0.082
Vote	Avg	**0.957**	0.9	**0.046**	0.052	**4.7**	6.85	3.919	**0.840**
	Stdv	0.004	0.047	0.004	0.017	1.129	1.814	0.312	0.067
Zoo	Avg	**1**	0.989	**0.003**	**0.003**	5.350	**4.85**	3.650	**0.784**
	Stdv	0.016	0.033	0	0.001	0.671	0.933	0.272	0.067
Lymphography	Avg	**0.877**	0.818	0.126	**0.088**	**7.7**	12.2	4.124	**0.791**
	Stdv	0.014	0.067	0.015	0.034	1.689	1.673	0.043	0.081
SpectEW	Avg	**0.859**	0.790	0.145	0.184	11.350	**10.35**	4.041	**0.801**
	Stdv	0.009	0.047	0.008	0.022	1.631	1.725	0.126	0.067
BreastEW	Avg	**0.964**	0.929	**0.041**	0.050	**15.5**	16.2	4.996	**1.046**
	Stdv	0.005	0.027	0.005	0.011	3.103	2.419	0.088	0.087
IonosphereEW	Avg	**0.937**	0.831	0.067	**0.038**	**13.950**	18.65	4.402	**0.856**
	Stdv	0.008	0.053	0.008	0.019	2.417	3.376	0.181	0.079
KrvskpEW	Avg	0.964	**0.968**	0.041	**0.039**	**18.8**	22.85	37.718	**3.922**
	Stdv	0.003	0.009	0.004	0.006	2.913	2.3	1.238	0.092
WaveformEW	Avg	0.748	**0.781**	0.255	**0.253**	**22.050**	26.95	93.915	**8.809**
	Stdv	0.006	0.021	0.006	0.010	3.034	3.052	3.411	0.153
SonarEW	Avg	**0.894**	0.768	0.110	**0.053**	**28.9**	34.4	4.261	0.838
	Stdv	0.011	0.047	0.011	0.011	2.989	3.872	0.061	0.086
clean1	Avg	**0.88**	0.869	0.123	**0.065**	**78.250**	100.4	6.959	**1.265**
	Stdv	0.007	0.031	0.006	0.009	7.973	5.471	0.144	0.028
semeion	Avg	**0.976**	0.972	0.029	**0.012**	**128.450**	157.35	53.574	**6.453**
	Stdv	0.002	0.005	0.002	0.002	9.423	10.917	1.450	0.151
PenglungEW	Avg	0.781	**0.836**	0.221	**0.147**	**148**	173.2	4.286	**0.987**
	Stdv	0.019	0.052	0.019	0	7.518	15.316	0.146	0.025
Colon	Avg	0.835	**0.992**	0.168	**0.154**	**944.45**	1070.8	5.181	**1.831**
	Stdv	0.01	0.037	0.010	0.051	20.213	112.571	0.108	0.035
Leukemia	Avg	0.903	**1**	0.101	**0.005**	**3473.8**	3666	13.860	**5.446**
	Stdv	0.014	0	0.014	0	38.030	254.699	1.993	0.445

The comparison results of the proposed βHC algorithm against other local search-based algorithms are reported in Tables 21, 22, 23, and 24. These tables show the average (Avg) and the standard deviation (Stdv) achieved by each algorithm in terms of the classification accuracy, the fitness values, the selected features, and the elapsed CPU time,

Table 20 The parameter settings of the local search-based algorithms

Algorithm	Parameter settings
βHC	$\mathcal{N}$=0.9, β=0.5
HC	bw = ($\mathcal{N}$ in βHC) = 0.9
SA	T_0=100, and α=0.85 (Corana et al. 1987)

respectively. The **bold** font is used to point out the finest outcomes.

Table 21 shows that the proposed binary βHC outperforms other comparative methods on 16 out of 22 datasets. On the other hand, the performance of the HC algorithm outperforms other comparative algorithms on 3 datasets. Also, the performance of the VNS algorithm is superior than other comparative algorithms on 3 datasets. While the SA algorithm did not achieve any best result for any dataset. Consequently, the average accuracy of the proposed binary βHC algorithm reached 100% on M-of-n and Zoo datasets and 99.9% for the Exactly dataset. This supports the claim that the proposed algorithm has successful trials to reach the proper stability between exploration and exploitation and thus avoid the local optimal problem.

Similarly, the experimental results obtained by the proposed binary βHC algorithm as well as the other comparative local search-based algorithms in terms of the fitness values are recorded in Table 22. This table shows that βHC, HC, and VNS algorithms are able to obtain the same average of the fitness values on the M-of-n dataset. Interestingly, the proposed binary βHC outperforms the other comparative methods on 13 out of 22 datasets. The HC and VNS algorithms obtained the finest outcomes on 4 datasets, although the SA algorithm did not realize any best result for any dataset.

Table 23 demonstrates the average of the selected features obtained by the proposed binary βHC algorithm and the other comparative local search-based algorithms. The SA algorithm obtains the finest outcomes on 13 datasets, although the VNS algorithm achieves the finest outcomes on 9 datasets. Furthermore, the proposed binary βHC and HC algorithms did not realize the finest outcomes for any dataset.

Finally, Table 24 demonstrates the average (Avg) and the standard deviation (Stdv) of the results produced by the proposed βHC algorithm as well as the other comparative local search-based algorithms in terms of the elapsed CPU time. The results summarized in Table 24 are in line with the results recorded in Table 23, where the SA algorithm achieved the finest outcomes on 13 datasets. Moreover, the VNS algorithm realize the finest outcomes on 9 datasets,

Table 21 The classification accuracy results of the binary βHC compared to other local search-based method

Dataset	βHC		HC		SA		VNS	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	**0.816**	0	0.779	0	0.686	0	0.765	0
Breastcancer	**0.974**	0.001	0.96	0.008	0.946	0.01	0.967	0.005
HeartEW	**0.836**	0.011	0.792	0.037	0.766	0.039	0.775	0.017
Exactly2	0.75	0.006	0.708	0.054	0.72	0.025	**0.775**	0.025
Exactly	**0.999**	0.003	0.856	0.147	0.619	0.039	0.899	0.111
M-of-n	**1.000**	0	0.984	0	0.744	0	0.99	0.018
WineEW	**0.996**	0.003	0.931	0.026	0.883	0.054	0.969	0.012
CongressEW	**0.974**	0.004	0.93	0.011	0.894	0.033	0.923	0.018
Vote	0.957	0.004	0.919	0.027	0.892	0.038	**0.959**	0.012
Zoo	**1.000**	0.016	0.837	0.041	0.879	0.065	0.831	0.055
Lymphography	**0.877**	0.014	0.753	0.056	0.691	0.066	0.801	0.029
SpectEW	**0.859**	0.009	0.789	0.036	0.799	0.03	0.821	0.02
BreastEW	**0.964**	0.005	0.925	0.012	0.927	0.009	0.945	0.006
IonosphereEW	**0.937**	0.008	0.868	0.02	0.81	0.017	0.856	0.021
KrvskpEW	0.964	0.003	**0.971**	0.011	0.773	0.084	0.965	0.007
WaveformEW	0.748	0.006	**0.788**	0.01	0.703	0.036	0.783	0.011
SonarEW	**0.894**	0.011	0.74	0.048	0.707	0.028	0.787	0.039
clean1	**0.88**	0.007	0.82	0.019	0.779	0.025	0.817	0.02
semeion	**0.976**	0.002	0.96	0.006	0.953	0.006	0.96	0.005
PenglungEW	0.781	0.019	**0.796**	0.032	0.786	0.026	0.611	0.031
Colon	**0.835**	0.01	0.581	0.036	0.634	0.035	0.731	0.026
Leukemia	0.903	0.014	0.879	0.021	0.886	0.022	**0.95**	0.021

Table 22 The fitness values of the binary βHC compared to other local search-based method

Dataset	βHC		HC		SA		VNS	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	**0.189**	0	0.206	0	0.339	0	0.226	0
Breastcancer	**0.030**	0.001	0.042	0.006	0.068	0.019	0.043	0.002
HeartEW	**0.167**	0.01	0.213	0.025	0.287	0.04	0.212	0.018
Exactly2	0.249	0.007	0.288	0.028	0.317	0.039	**0.233**	0.035
Exactly	**0.005**	0.003	0.137	0.151	0.435	0.047	0.059	0.115
M-of-n	**0.005**	0	**0.005**	0	0.309	0	**0.005**	0.002
WineEW	**0.009**	0.004	0.049	0.022	0.118	0.066	0.016	0.007
CongressEW	**0.029**	0.005	0.053	0.011	0.108	0.05	0.039	0.01
Vote	0.046	0.004	0.05	0.017	0.133	0.051	**0.034**	0.006
Zoo	**0.003**	0	0.059	0.015	0.121	0.092	0.039	0.007
Lymphography	**0.126**	0.015	0.177	0.027	0.322	0.082	0.149	0.018
SpectEW	**0.145**	0.008	0.187	0.023	0.276	0.044	0.158	0.008
BreastEW	**0.041**	0.005	0.06	0.007	0.087	0.016	0.044	0.004
IonosphereEW	**0.067**	0.008	0.07	0.016	0.169	0.024	0.081	0.011
KrvskpEW	0.041	0.004	**0.029**	0.007	0.24	0.08	0.031	0.007
WaveformEW	0.255	0.006	**0.252**	0.009	0.357	0.037	0.255	0.013
SonarEW	0.11	0.011	0.105	0.025	0.274	0.033	**0.081**	0.021
clean1	0.123	0.006	**0.087**	0.019	0.18	0.014	0.115	0.013
semeion	0.029	0.002	0.03	0.005	0.051	0.007	**0.028**	0.005
PenglungEW	0.221	0.019	**0.099**	0.02	0.171	0.025	0.21	0.036
Colon	**0.168**	0.01	0.282	0.028	0.423	0.052	0.271	0.028
Leukemia	**0.101**	0.014	0.152	0.03	0.177	0.031	0.155	0.023

Table 23 The selected features of the binary βHC compared to other local search-based method

Dataset	βHC		HC		SA		VNS	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	5.708	0	5.993	0	**1.354**	0	1.452	0
Breastcancer	4.934	0.075	5.186	0.34	**1.482**	0.098	2.301	0.097
HeartEW	4.304	0.061	3.819	0.216	1.019	0.065	**0.988**	0.042
Exactly2	5.752	0.392	6.747	1.529	**1.522**	0.189	1.551	0.176
Exactly	6.077	0.085	6.101	0.912	**1.432**	0.147	1.545	0.145
M-of-n	6.109	0	5.544	0	1.539	0	**1.455**	0.185
WineEW	4.126	0.052	3.619	0.178	0.91	0.047	**0.891**	0.051
CongressEW	4.311	0.3	4.3	0.322	1.093	0.084	**1.077**	0.087
Vote	3.919	0.312	3.842	0.14	**0.936**	0.042	1.031	0.057
Zoo	3.65	0.272	3.818	0.179	0.95	0.072	**0.916**	0.064
Lymphography	4.124	0.043	3.386	0.082	**0.894**	0.084	0.978	0.046
SpectEW	4.041	0.126	3.641	0.137	0.955	0.062	**0.950**	0.089
BreastEW	4.996	0.088	4.961	0.343	**1.323**	0.102	1.502	0.17
IonosphereEW	4.402	0.181	4.019	0.155	1.061	0.087	**0.950**	0.034
KrvskpEW	37.718	1.238	43.358	4.145	**8.753**	0.927	9.377	0.886
WaveformEW	93.915	3.411	117.174	15.298	**21.976**	2.425	23.691	2.974
SonarEW	4.261	0.061	3.464	0.152	**0.941**	0.04	0.969	0.057
clean1	6.959	0.144	6.454	0.521	**1.588**	0.173	1.712	0.163
semeion	53.574	1.45	51.981	3.194	**12.662**	0.814	12.929	1.489
PenglungEW	4.286	0.146	3.755	0.187	1.019	0.065	**0.975**	0.056
Colon	5.181	0.108	4.607	0.162	**1.119**	0.035	1.191	0.08
Leukemia	13.86	1.993	15.023	0.518	2.755	0.299	**2.682**	0.292

Table 24 The CPU time (in Seconds) of the binary βHC compared to other local search-based method

Dataset	βHC		HC		SA		VNS	
	Avg	Stdv	Avg	Stdv	Avg	Stdv	Avg	Stdv
Tic-tac-toe	5.708	0	5.993	0	**1.354**	0	1.452	0
Breastcancer	4.934	0.075	5.186	0.34	**1.482**	0.098	2.301	0.097
HeartEW	4.304	0.061	3.819	0.216	1.019	0.065	**0.988**	0.042
Exactly2	5.752	0.392	6.747	1.529	**1.522**	0.189	1.551	0.176
Exactly	6.077	0.085	6.101	0.912	**1.432**	0.147	1.545	0.145
M-of-n	6.109	0	5.544	0	1.539	0	**1.455**	0.185
WineEW	4.126	0.052	3.619	0.178	0.91	0.047	**0.891**	0.051
CongressEW	4.311	0.3	4.3	0.322	1.093	0.084	**1.077**	0.087
Vote	3.919	0.312	3.842	0.14	**0.936**	0.042	1.031	0.057
Zoo	3.65	0.272	3.818	0.179	0.95	0.072	**0.916**	0.064
Lymphography	4.124	0.043	3.386	0.082	**0.894**	0.084	0.978	0.046
SpectEW	4.041	0.126	3.641	0.137	0.955	0.062	**0.95**	0.089
BreastEW	4.996	0.088	4.961	0.343	**1.323**	0.102	1.502	0.17
IonosphereEW	4.402	0.181	4.019	0.155	1.061	0.087	**0.95**	0.034
KrvskpEW	37.718	1.238	43.358	4.145	**8.753**	0.927	9.377	0.886
WaveformEW	93.915	3.411	117.174	15.298	**21.976**	2.425	23.691	2.974
SonarEW	4.261	0.061	3.464	0.152	**0.941**	0.04	0.969	0.057
clean1	6.959	0.144	6.454	0.521	**1.588**	0.173	1.712	0.163
semeion	53.574	1.45	51.981	3.194	**12.662**	0.814	12.929	1.489
PenglungEW	4.286	0.146	3.755	0.187	1.019	0.065	**0.975**	0.056
Colon	5.181	0.108	4.607	0.162	**1.119**	0.035	1.191	0.08
Leukemia	13.86	1.993	15.023	0.518	2.755	0.299	**2.682**	0.292

while the proposed βHC and HC algorithms did not get any of the best results.

3.4 Comparison with other metaheuristics

The proposed binary βHC efficiency is verified by comparing it against ten other metaheuristics methods from the literature. These methods include: binary grasshopper optimization algorithm (BGOA) (Mafarja et al. 2018b), binary grey wolf optimizer (BGWO) (Mafarja et al. 2018b), binary gravitational search algorithm (BGSA) (Mafarja et al. 2018b), binary bat algorithm (BBA) (Mafarja et al. 2018b), binary salp swarm algorithm (BSSA) (Aljarah et al. 2018), hybrid gravitational search algorithm (HGSA) (Taradeh et al. 2019), whale optimization algorithm (WOA) (Mafarja and Mirjalili 2018), binary dragonfly optimization (BDA) (Mafarja et al. 2018a), genetic algorithm (GA) (Kashef and Nezamabadi-pour 2015), and particle swarm optimization (PSO) (Kashef and Nezamabadi-pour 2015).

The average classification accuracy achieved by all comparative methods is summarized in Table 25. The finest outcomes are highlighted in boldface. The results of the proposed binary βHC algorithm are collected from Table 3 and Table 7. The results in Table 25 demonstrates that the BDA algorithm achieves the best performance as it outperforms other comparative methods in 14 datasets. Remarkably, the binary βHC algorithm ranked second, as it outperforms other comparative methods in 7 datasets. On the other hand, six of the comparative methods did not obtain any best results for any dataset.

Table 26 illustrates the average ranking of the proposed binary βHC algorithm using Friedman statistical test when compared against the comparative methods. Note that the average ranking of the comparative methods is computed using the results in Table 25. The Significant level α is set to 0.05 as suggested in (García et al. 2010). Interestingly, the binary βHC algorithm outperforms other comparative methods by getting the first rank.

Thereafter, the Holm and Hochberg as a post-hoc statistical test are utilized to calculate the adjusted ρ-values between the first rank method (i.e., controlled method) identified by Friedman test and other methods. Again, the proposed binary βHC algorithm is ranked first as demonstrated in Table 26. Table 27 reveals that the performance of the binary βHC algorithm is statistically superior than 7 of the other comparative methods (i.e., BSSA, WOA, BGWO, BGSA, PSO, GA, and BBA) using the significant level α/Order. Also, the statistical test expose that there is no significant difference between the binary βHC algorithm and three of the comparative methods (BDA, BGOA, and HGSA).

Table 25 The classification accuracy results of the binary βHC compared to other metaheuristics

Dataset	βHC	BGOA	BGWO	BGSA	BBA	BSSA	HGSA	WOA	BDA	GA	PSO
Tic-tac-toe	0.816	0.808	0.754	0.753	0.665	0.797	0.788	0.785	**0.847**	0.764	0.750
Breastcancer	0.974	0.980	0.968	0.957	0.937	–	0.974	0.968	**0.993**	0.957	0.949
HeartEW	0.859	0.833	0.792	0.777	0.754	0.833	0.856	0.807	**0.876**	0.732	0.745
Exactly2	0.750	**0.780**	0.743	0.706	0.628	0.767	0.770	0.742	0.773	0.677	0.666
Exactly	**1.000**	0.999	0.809	0.697	0.610	0.997	**1.000**	**1.000**	**1.000**	0.822	0.973
M-of-n	**1.000**	**1.000**	0.894	0.835	0.722	0.999	**1.000**	0.991	**1.000**	0.916	0.996
WineEW	0.996	0.989	0.960	0.951	0.919	0.998	0.989	0.959	**1.000**	0.947	0.937
CongressEW	0.975	0.964	0.948	0.951	0.872	0.970	0.966	0.956	**0.987**	0.898	0.937
Vote	0.978	0.966	0.944	0.931	0.851	0.955	0.973	0.939	**0.989**	0.808	0.888
Zoo	**1.000**	0.993	0.975	0.939	0.874	0.993	0.932	0.980	**1.000**	0.946	0.963
Lymphography	0.907	0.868	0.813	0.781	0.701	0.844	0.892	0.852	**0.992**	0.758	0.759
SpectEW	0.866	0.826	0.810	0.783	0.800	0.833	**0.919**	0.866	0.852	0.756	0.738
BreastEW	0.969	0.947	0.954	0.942	0.931	–	0.971	0.971	**0.979**	0.923	0.933
IonosphereEW	0.949	0.899	0.885	0.881	0.877	0.938	0.934	0.926	**0.991**	0.863	0.876
KrvskpEW	**0.985**	0.968	0.934	0.908	0.816	0.969	0.978	0.972	0.979	0.940	0.949
WaveformEW	0.772	0.737	0.723	0.695	0.669	0.736	**0.815**	0.753	0.758	0.712	0.732
SonarEW	0.953	0.912	0.836	0.888	0.844	0.948	0.958	0.919	**0.984**	0.833	0.804
clean1	**0.941**	0.863	0.908	0.898	0.826	–	–	–	–	0.862	0.845
semeion	**0.989**	0.976	0.972	0.971	0.962	–	–	–	–	0.963	0.967
PenglungEW	0.912	0.927	0.850	0.919	0.795	0.907	0.956	0.792	**1.000**	0.672	0.879
Colon	0.868	**0.870**	0.661	0.766	0.682	–	–	–	–	0.682	0.624
Leukemia	**0.971**	0.931	0.884	0.844	0.877	–	–	–	–	0.705	0.862

Table 26 Average rankings of the algorithms calculated using Friedman test

Order	Algorithm	Ranking
1	βHC	2.386
2	BDA	3.028
3	BGOA	3.909
4	HGSA	4.432
5	BSSA	6.000
6	WOA	6.136
7	BGWO	6.568
8	BGSA	7.432
9	PSO	8.182
10	GA	8.636
11	BBA	9.295

Table 27 Holm/Hochberg results between the binary βHC algorithm and other methods

Order	Algorithm	Adjusted ρ -value	Holm/Hochberg	Null hypotheses
1	BDA	5.245E-01	0.0500	Not Rejected
2	BGOA	1.278E-01	0.0250	Not Rejected
3	HGSA	4.081E-02	0.0167	Not Rejected
4	BSSA	3.019E-04	0.0125	Rejected
5	WOA	1.768E-04	0.0100	Rejected
6	BGWO	2.892E-05	0.0083	Rejected
7	BGSA	4.524E-07	0.0071	Rejected
8	PSO	6.814E-09	0.0063	Rejected
9	GA	4.105E-10	0.0056	Rejected
10	BBA	4.878E-12	0.0050	Rejected

3.5 Comparison with filter-based techniques

For further validations, the proposed binary βHC algorithm efficiency is compared against relevant filter-based techniques in terms of the average classification accuracy. The comparative outcomes are recorded in Table 28. Note that the filter-based techniques outcomes are taken from Mafarja et al. (2018b) which uses similar configurations as the proposed method. The best outcomes are point out using bold font. To elaborate, the results of the binary βHC algorithm are extracted from Tables 3 and 7. While the comparative filter-based techniques are: correlation-based feature selection from Hall and Smith (1999), fast correlation-based filter (FCBF) from Yu and Liu (2003), fisher score (F-score) from Duda et al. (2012), IG from Cover and Thomas (2012), and wavelet power spectrum (WPS) from Zhao and Liu (2007). Table 28 demonstrates that the binary βHC algorithm outperforms the other techniques in 20 out of 22 datasets. This proves that the binary βHC algorithm is capable to examine the search space efficiently and obtain good results when compared to other comparative methods.

Table 28 The accuracy results of the binary βHC compared to all filter-based methods

Dataset	βHC	CFS	FCBF	F-Score	IG	WPS
Tic-tac-toe	**0.816**	0.000	0.000	0.010	0.010	0.167
Breastcancer	0.974	0.957	**0.986**	0.979	0.957	0.957
HeartEW	**0.859**	0.648	0.648	0.759	0.759	0.796
Exactly2	**0.750**	0.705	0.545	0.680	0.620	0.660
Exactly	**1.000**	0.670	0.440	0.600	0.615	0.575
M-of-n	**1.000**	0.785	0.815	0.815	0.815	0.580
WineEW	**0.996**	0.778	0.889	0.861	0.889	0.889
CongressEW	**0.975**	0.793	0.793	0.908	0.828	0.828
Vote	**0.978**	0.950	0.950	0.933	0.967	0.850
Zoo	**1.000**	0.800	0.900	0.650	0.850	0.600
Lymphography	**0.907**	0.500	0.567	0.667	0.667	0.767
SpectEW	**0.866**	0.736	0.774	0.793	0.793	0.736
BreastEW	**0.969**	0.825	0.798	0.930	0.930	0.772
IonosphereEW	**0.949**	0.857	0.857	0.729	0.800	0.829
KrvskpEW	**0.985**	0.768	0.934	0.959	0.934	0.377
WaveformEW	**0.772**	0.620	0.710	0.662	0.662	0.292
SonarEW	**0.953**	0.310	0.214	0.048	0.191	0.048
clean1	**0.941**	0.716	0.642	0.632	0.547	0.611
semeion	**0.989**	0.875	0.875	0.875	0.868	0.875
PenglungEW	**0.912**	0.600	0.667	0.800	0.667	0.400
Colon	**0.868**	0.750	0.667	0.667	0.667	0.500
Leukemia	0.971	0.929	0.857	**0.980**	**0.980**	0.357

Table 29 Average rankings of the algorithms calculated using Friedman test

Order	Algorithm	Ranking
1	binary βHC	1.364
2	F-Score	3.432
3	FCBF	3.818
4	CFS	4.068
5	WPS	4.727
6	IG	3.591

Table 30 Holm/Hochberg results between the binary βHC algorithm and filter-based techniques

Order	Algorithm	Adjusted ρ -value	Holm/Hoch-berg	Null hypotheses
1	F-Score	2.46E-04	0.0500	Rejected
2	IG	7.86E-05	0.0250	Rejected
3	FCBF	1.35E-05	0.0167	Rejected
4	CFS	1.63E-06	0.0125	Rejected
5	WPS	2.48E-09	0.0100	Rejected

Table 29 shows the average ranking of the proposed binary βHC algorithm and the five filter-based techniques using the Friedman test. These ranking are computed using the outcomes presented in Table 28. Again, the significant level of α is 0.05. Remarkably, the binary βHC algorithm is ranked first, while the F-score technique shows that it is ranked second. In addition, the FCBF, CFS, WPS, and IG techniques reserved the third position until the last one respectively. Eventually, the Holm and Hochberg statistical test is utilized to calculate the ρ-values between the controlled algorithm (i.e., binary βHC algorithm) and the other comparative techniques. Interestingly, there are significant differences between the binary βHC algorithm and the five filter-based techniques as demonstrated in Table 30.

After heavy experiments conducted to prove the viability of the proposed method, we can conclude that the proposed method very efficient algorithm for feature selection problems. The proposed method is able is work well when the value of parameters $\mathcal{N}$ is large and β is small. This is because the $\mathcal{N}$ parameter determines to what extend the proposed method can be make use of accumulative search while the β parameter is like mutation rate which determines the size of randomization in the search.

Apparently, the proposed method competes very well against other local search-based methods and with other advanced metaheuristic techniques. This is can be borne out by the results obtained. Although, the proposed method belongs to the local search-based algorithm which is normally simple and easy-to-use, it outperforms other methods in 7 out of 22 datasets of various sizes and complexities. This is because that the proposed method is able to achieve the right balance between exploration through $\mathcal{N}$ operator and exploitation through β operator. Interestingly, the computational time required to achieve the final results for the proposed method is very small. It is conventionally known that the time required for local search-based algorithm required less computational time than other population-based algorithms.

4 Conclusion and future works

In this paper, β-hill climbing which is a new version of a local search-based algorithm is proposed to solve the feature selection problem. The proposed algorithm is called binary β-hill climbing optimizer. A new operator called $\mathcal{T}$-operator is added to the β-hill climbing algorithm operators ($\mathcal{N}$ -operator, β-operator, and $\mathcal{S}$-operator) to transfer the continuous values of the produced feature solution into binary using the S-shape strategy.

To evaluate the proposed binary β-hill climbing optimizer, Four measurements are used: fitness function, classification accuracy, number of relevant features, and elapsed CPU time consumption. In order to evaluate the proposed algorithm, a commonly used 22 problem instances are picked from the UCI datasets with various sizes and complexities. Different evaluation experiments are conducted which include: parameter configurations, transfer functions effect, the used classifier effect, and comparative evaluations against other local search-based methods as well as other population-based algorithms using the same UCI datasets.

The influence of the main parameters (i.e., β and $\mathcal{N}$) of binary β-hill climbing optimizer on the algorithm convergence behavior is studied. In conclusion, for β operator, the higher value can achieve superior results. This means that the exploration is much beneficial for feature selection problem search space. Also, larger $\mathcal{N}$ parameter value obtains better results in most cases. Furthermore, eight different transfer functions are experimented with which include S-shaped and V-Shaped transfer functions. In summary, the outcomes produced by the proposed binary βHC with S-shape can almost excel all other results produced by other transfer functions. Furthermore, three classifiers are utilized to find the classification accuracy (i.e., kNN, SVM, and decision tree (DT)). The kNN is adapted for the proposed method since it has the best performance. However, the main limitation of the proposed method is the parameter configurations and the trajectory-based search which might result in being stuck in local minima very quickly. Furthermore, the convergence behavior of the proposed method might be varied from feature selection problem to another due to the No-Free-Lunch theorem in optimization (Wolpert et al. 1997).

The proposed binary β-hill climbing is compared against 13 other comparative methods (3 local-search-based algorithms and 10 metaheuristics algorithms) using the same dataset. The proposed binary β-hill climbing optimizer can excel other comparative local search-based approaches in 16 out of 22 datasets. This means that the proposed method is very competitive when compared to other local search-based approaches. On the other hand, the binary β-hill climbing can outperform other comparative metaheuristic approaches in 7 out of 22 datasets. These results prove the effectiveness of the proposed binary β-hill climbing optimizer as an important addition to the body of knowledge in the machine learning and classifications domain.

The Friedman statistical test at a significant level α set to 0.05 shows that the binary βHC algorithm outperforms other comparative metaheuristic methods by getting the first rank. In addition, by applying Holm and Hochberg as a post-hoc statistical test the performance of the binary βHC algorithm is statistically better than 7 of the other comparative metaheuristic methods (i.e., BSSA, WOA, BGWO, BGSA, PSO, GA, and BBA) using the significant level α/Order.

As the proposed binary β-hill climbing is shown to be very successful when used to solve the feature selection problem, we plan to address the following issues in the future:

Feature selection applications: other feature selection applications such as gene selection which might be more complex can be tackled using the proposed binary β-hill climbing optimizer.

adaptiveβ-HC: the adaptive version of β hill climbing optimizer can be utilized for feature selection applications to simplify the adaptations.

Hybridization with population-based algorithm: hybridize binary β hill climbing optimizer with other swarm-based algorithms to empower the exploration capabilities of the algorithm.

Using other evaluating measurements in the experiments: The performance of the proposed method can be also evaluated using other evaluating measurements such as sensitivity, specificity, area under the curve (AUC), and others.

Experimenting using large-scaled datasets: Large-scaled datasets can be experimented with to evaluate the efficiency and scalability of the proposed method.

References

Abed-alguni B, Klaib A (2018) Hybrid whale optimization and β-hill climbing algorithm. Int J CompuT Sci Math, pp 1–13

Abed-alguni BH, Alkhateeb F (2018) Intelligent hybrid cuckoo search and β-hill climbing algorithm. J King Saud Univ Comput Inf Sci 32(2):159–173

Abualigah LM, Khader AT, Al-Betar MA (2017a) β-hill climbing technique for the text document clustering. New Trends in Information Technology NTIT2017 Conference, Amman, Jordan, IEEE, pp 1–6

Abualigah LM, Khadery AT, Al-Betar MA, Alyasseri ZAA, Alomari OA, Hanandehk ES (2017b) Feature selection with β-hill climbing search for text clustering application. Second Palestinian International Conference on Information and Communication Technology (PICICT 2017), Gaza, Palestine, IEEE, pp 22–27

Al-Abdallah RZ, Jaradat AS, Doush IA, Jaradat YA (2017) Abinary classifier based on firefly algorithm. Jordan J Comput Inf Technol (JJCIT) 3(3)

Al-Betar MA (2017) β-hill climbing: an exploratory local search. Neural Comput Appl 28(1):153–168. https://doi.org/10.1007/s00521-016-2328-2

Al-Betar MA, Awadallah MA, Bolaji AL, Alijla BO (2017) β-hill climbing algorithm for sudoku game. Second Palestinian International Conference on Information and Communication Technology (PICICT 2017), Gaza, Palestine, IEEE, pp 84–88

Al-Betar MA, Awadallah MA, Abu Doush I, Alsukhni E, ALkhraisat H (2018) A non-convex economic dispatch problem with valve loading effect using a new modified β-hill climbing local search algorithm. Arab J Sci Eng. https://doi.org/10.1007/s13369-018-3098-1

Aljarah I, Mafarja M, Heidari AA, Faris H, Zhang Y, Mirjalili S (2018) Asynchronous accelerating multi-leader salp chains for feature selection. Appl Soft Comput 71:964–979

Alomari OA, Khader AT, Al-Betar MA, Alyasseri ZAA (2018a) A hybrid filter-wrapper gene selection method for cancer classification. 2018 2nd International Conference on BioSignal Analysis. Processing and Systems (ICBAPS), IEEE, pp 113–118

Alomari OA, Khader AT, Al-Betar MA, Awadallah MA (2018b) A novel gene selection method using modified mrmr and hybrid bat-inspired algorithm with β-hill climbing. Appl Intell 48(11):4429–4447

Alsaafin A, Elnagar A (2017) A minimal subset of features using feature selection for handwritten digit recognition. J Intell Learn Syst Appl 9(4):55–68

Alsukni E, Arabeyyat OS, Awadallah MA, Alsamarraie L, Abu-Doush I, Al-Betar MA (2017) Multiple-reservoir scheduling using β-hill climbing algorithm. J Intell Syst 28(4):559–570

Alyasseri ZAA, Khader AT, Al-Betar MA (2017a) Optimal eeg signals denoising using hybrid β-hill climbing algorithm and wavelet transform. ICISPC '17. Penang, Malaysia, ACM, pp 5–11

Alyasseri ZAA, Khader AT, Al-Betar MA (2017b) Optimal electroencephalogram signals denoising using hybrid β-hill climbing algorithm and wavelet transform. In: Proceedings of the International Conference on Imaging, Signal Processing and Communication, ACM, pp 106–112

Alyasseri ZAA, Khader AT, Al-Betar MA, Awadallah MA (2018) Hybridizing β-hill climbing with wavelet transform for denoising ecg signals. Inf Sci 429:229–246

Alzaidi AA, Ahmad M, Doja MN, Al Solami E, Beg MS (2018) A new 1d chaotic map and *beta*-hill climbing for generating substitution-boxes. IEEE Access 6:55405–55418

Bermejo P, Gámez JA, Puerta JM (2011) A grasp algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. Pattern Recognit Lett 32(5):701–711

Bermejo P, Gámez JA, Puerta JM (2014) Speeding up incremental wrapper feature subset selection with naive bayes classifier. Knowl-Based Syst 55(Supplement C):14–147, https://doi.org/10.1016/j.knosys.2013.10.016, http://www.sciencedirect.com/science/article/pii/S0950705113003274

Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv (CSUR) 35(3):268–308

Bolón-Canedo V, Alonso-Betanzos A (2019) Ensembles for feature selection: a review and future trends. Inf Fusion 52:1–12

Boughaci D, Alkhawaldeh AAs (2018) Three local search-based methods for feature selection in credit scoring. Vietnam J Comput Sci 5(2):107–121

Chen Y, Garcia EK, Gupta MR, Rahimi A, Cazzanti L (2009) Similarity-based classification: concepts and algorithms. J Mach Learn Res 10(Mar):747–776

Corana A, Marchesi M, Martini C, Ridella S (1987) Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm corrigenda for this article is available here. ACM Trans Math Softw (TOMS) 13(3):262–280

Cover TM, Thomas JA (2012) Elements of information theory. Wiley, Hoboken

Delen D, Walker G, Kadam A (2005) Predicting breast cancer survivability: a comparison of three data mining methods. Artif Intell Med 34(2):113–127

Deng X, Li Y, Weng J, Zhang J (2019) Feature selection for text classification: a review. Multimed Tools Appl 78(3):3797–3816

Doush IA, Sahar AB (2017) Currency recognition using a smartphone: comparison between color sift and gray scale sift algorithms. J King Saud Univ Comput Inf Sci 29(4):484–492

Dubey SR, Singh SK, Singh RK (2015) Local wavelet pattern: a new feature descriptor for image retrieval in medical ct databases. IEEE Trans Image Process 24(12):5892–5903

Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley, Hoboken

Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. Neurocomputing 213:54–65

Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3(Mar):1289–1305

García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Inf Sci 180(10):2044–2064

Ghareb AS, Bakar AA, Hamdan AR (2016) Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Syst Appl 49:31–47

Gheyas IA, Smith LS (2010) Feature subset selection in large dimensionality domains. Pattern Recognit 43(1):5–13. https://doi.org/10.1016/j.patcog.2009.06.009, http://www.sciencedirect.com/science/article/pii/S0031320309002520

Goltsev A, Gritsenko V (2012) Investigation of efficient features for image recognition by neural networks. Neural Netw 28:15–23

Gu S, Cheng R, Jin Y (2018) Feature selection for high-dimensional classification using a competitive swarm optimizer. Soft Comput 22(3):811–822

Hall MA, Smith LA (1999) Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. FLAIRS Conf 1999:235–239

Hu Q, Yu D, Xie Z (2006) Information-preserving hybrid data reduction based on fuzzy-rough techniques. Pattern Recognit Lett 27(5):414–423

Jing LP, Huang HK, Shi HB (2002) Improved feature selection approach tfidf in text mining. In: Proceedings. International Conference on Machine Learning and Cybernetics, IEEE, vol 2, pp 944–946

Kabir MM, Shahjahan M, Murase K (2012) A new hybrid ant colony optimization algorithm for feature selection. Expert Syst Appl 39(3):3747–3763

Kashef S, Nezamabadi-pour H (2015) An advanced aco algorithm for feature subset selection. Neurocomputing 147:271–279

Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, IEEE, vol 5, pp 4104–4108

Lai C, Reinders MJ, Wessels L (2006) Random subspace method for multivariate feature selection. Pattern Recognit Lett 27(10):1067–1076

Lee C, Lee GG (2006) Information gain and divergence-based feature selection for machine learning-based text categorization. Inf Process Manag 42(1):155–165

Li P, Shrivastava A, Moore JL, König AC (2011) Hashing algorithms for large-scale learning. In: Advances in neural information processing systems, pp 2672–2680

Li Y, Li T, Liu H (2017) Recent advances in feature selection and its applications. Knowl Inf Syst 53(3):551–577

Liao Y, Vemuri VR (2002) Use of k-nearest neighbor classifier for intrusion detection. Comput Secur 21(5):439–448

Liu H, Setiono R (1995) Chi2: Feature selection and discretization of numeric attributes, pp 388–391

Ma B, Xia Y (2017) A tribe competition-based genetic algorithm for feature selection in pattern classification. Appl Soft Comput 58(Supplement C):328–338, https://doi.org/10.1016/j.asoc.2017.04.042, http://www.sciencedirect.com/science/article/pii/S1568494617302247

Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453

Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li X, Mirjalili S (2018a) Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowl-Based Syst 161:185–204

Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Ala'M AZ, Mirjalili S (2018b) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl-Based Syst 145:25–45

Mafarja M, Aljarah I, Faris H, Hammouri AI, Ala'M AZ, Mirjalili S (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Syst Appl 117:267–286

Marinaki M, Marinakis Y (2015) A hybridization of clonal selection algorithm with iterated local search and variable neighborhood search for the feature selection problem. Memetic Comput 7(3):181–201

Mashrgy MA, Bdiri T, Bouguila N (2014) Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted dirichlet mixture models. Knowl-Based Syst 59(Supplement C):182–195, https://doi.org/10.1016/j.knosys.2014.01.007, http://www.sciencedirect.com/science/article/pii/S0950705114000185

Mlakar U, Fister I, Brest J, Potočnik B (2017) Multi-objective differential evolution for feature selection in facial expression recognition systems. Expert Syst Appl 89:129–137

Moayedikia A, Ong KL, Boo YL, Yeoh WG, Jensen R (2017) Feature selection for high dimensional imbalanced class data using harmony search. Eng Appl Artif Intell 57:38–49

Park CH, Kim SB (2015) Sequential random k-nearest neighbor feature selection for high-dimensional data. Expert Syst Appl 42(5):2336–2342. https://doi.org/10.1016/j.eswa.2014.10.044, http://www.sciencedirect.com/science/article/pii/S095741741400668X

Rashedi E, Nezamabadi-Pour H, Saryazdi S (2013) A simultaneous feature adaptation and feature selection method for content-based image retrieval systems. Knowl-Based Syst 39:85–94

Ravisankar P, Ravi V, Rao GR, Bose I (2011) Detection of financial statement fraud and feature selection using data mining techniques. Decision Support Syst 50(2):491–500

Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis of relieff and rrelieff. Mach Learn 53(1):23–69

Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Sawalha R, Doush IA (2012) Face recognition using harmony search-based selected features. Int J Hybrid Inf Technol 5(2):1–16

Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. Neural Comput Appl 31(1):171–188

Shang C, Li M, Feng S, Jiang Q, Fan J (2013) Feature selection via maximizing global information gain for text classification. Knowl-Based Syst 54(Supplement C):298 – 309, https://doi.org/10.1016/j.knosys.2013.09.019, http://www.sciencedirect.com/science/article/pii/S0950705113003067

Shao C, Paynabar K, Kim TH, Jin JJ, Hu SJ, Spicer JP, Wang H, Abell JA (2013) Feature selection for manufacturing process monitoring using cross-validation. J Manuf Syst 32(4):550–555

Sindhu SSS, Geetha S, Kannan A (2012) Decision tree based light weight intrusion detection using a wrapper approach. Expert Syst Appl 39(1):129–141. https://doi.org/10.1016/j.eswa.2011.06.013, http://www.sciencedirect.com/science/article/pii/S0957417411009080

Talbi EG (2009) Metaheuristics: from design to implementation, vol 74. Wiley, Hoboken

Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H (2019) An evolutionary gravitational search-based feature selection. Inf Sci 497:219–239

Urbanowicz RJ, Olson RS, Schmitt P, Meeker M, Moore JH (2018) Benchmarking relief-based feature selection methods for bioinformatics data mining. J Biomed Inform 85:168–188

Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10(Feb):207–244

Wieland M, Pittore M (2014) Performance evaluation of machine learning algorithms for urban pattern recognition from multispectral satellite images. Remote Sens 6(4):2912–2939

Wolpert DH, Macready WG et al (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 856–863

Zhang H, Sun G (2002) Feature selection using tabu search method. Pattern Recognit 35(3):701–711

Zhang L, Mistry K, Lim CP, Neoh SC (2018) Feature selection using firefly optimization for classification and regression models. Decision Support Syst 106:64–85

Zhang Y, Wang S, Phillips P, Ji G (2014) Binary pso with mutation operator for feature selection using decision tree applied to spam detection. Knowl-Based Syst 64:22–31

Zhao Z, Liu H (2007) Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th international conference on Machine learning, ACM, pp 1151–1157

Zhong N, Dong J, Ohsuga S (2001) Using rough sets with heuristics for feature selection. J Intell Inf Syst 16(3):199–214