



An improved binary sparrow search algorithm for feature selection in data classification

Ahmed G. Gad¹ · Karam M. Sallam^{2,3} · Ripon K. Chakraborty⁴ · Michael J. Ryan⁵ · Amr A. Abohany¹

Received: 13 February 2021 / Accepted: 29 March 2022
© The Author(s) 2022

Abstract

Feature Selection (FS) is an important preprocessing step that is involved in machine learning and data mining tasks for preparing data (especially high-dimensional data) by eliminating irrelevant and redundant features, thus reducing the potential curse of dimensionality of a given large dataset. Consequently, FS is arguably a combinatorial NP-hard problem in which the computational time increases exponentially with an increase in problem complexity. To tackle such a problem type, meta-heuristic techniques have been opted by an increasing number of scholars. Herein, a novel meta-heuristic algorithm, called Sparrow Search Algorithm (SSA), is presented. The SSA still performs poorly on exploratory behavior and exploration-exploitation trade-off because it does not duly stimulate the search within feasible regions, and the exploitation process suffers noticeable stagnation. Therefore, we improve SSA by adopting: i) a strategy for Random Re-positioning of Roaming Agents (3RA); and ii) a novel Local Search Algorithm (LSA), which are algorithmically incorporated into the original SSA structure. To the FS problem, SSA is improved and cloned as a binary variant, namely, the improved Binary SSA (iBSSA), which would strive to select the optimal or near-optimal features from a given dataset while keeping the classification accuracy maximized. For binary conversion, the iBSSA was primarily validated against nine common S-shaped and V-shaped Transfer Functions (TFs), thus producing nine iBSSA variants. To verify the robustness of these variants, three well-known classification techniques, including *k*-Nearest Neighbor (*k*-NN), Support Vector Machine (SVM), and Random Forest (RF) were adopted as fitness evaluators with the proposed iBSSA approach and many other competing algorithms, on 18 multifaceted, multi-scale benchmark datasets from the University of California Irvine (UCI) data repository. Then, the overall best-performing iBSSA variant for each of the three classifiers was compared with binary variants of 12 different well-known meta-heuristic algorithms, including the original SSA (BSSA), Artificial Bee Colony (BABC), Particle Swarm Optimization (BPSO), Bat Algorithm (BBA), Grey Wolf Optimization (BGWO), Whale Optimization Algorithm (BWOA), Grasshopper Optimization Algorithm (BGOA) SailFish Optimizer (BSFO), Harris Hawks Optimization (BHHO), Bird Swarm Algorithm (BBSA), Atom Search Optimization (BASO), and Henry Gas Solubility Optimization (BHGSO). Based on a Wilcoxon's non-parametric statistical test ($\alpha = 0.05$), the superiority of iBSSA with the three classifiers was very evident against counterparts across the vast majority of the selected datasets, achieving a feature size reduction of up to 92% along with up to 100% classification accuracy on some of those datasets.

Keywords Machine learning · Classification · Feature selection · Meta-heuristics · Sparrow search algorithm · Combinatorial optimization · Swarm-based optimization · Pattern recognition · Data mining · Binary optimization

1 Introduction

Data mining is a rapidly growing paradigm in data science, due to the massive data gathered daily as well as the excessive demand to turn this data into useful knowledge

[46]. Data mining includes multiple preprocessing steps (i.e., integration, filtering, transformation, reduction, etc.), knowledge presentation, and pattern evaluation [69]. Amongst them, Feature Selection (FS), also known as variable (feature) subset selection, is one of such prerequisites, which is reliable to strive to reduce the high computational costs required by heavy mining tasks by

Extended author information available on the last page of the article

discarding potential noisy, redundant, and irrelevant features that may degrade the classification accuracy. Basically, the major objective of an FS procedure is to optimally carry out the classification task by first finding a subset of variables. A given set of features are usually initially given and then an optimal subset of features is thought, pursuing high classification performance.

The term FS resembles an algorithm using an input feature set to output a subset of it. In supervised learning, a set of instances of the same attributes exists along with another attribute representing the corresponding class. Via those instances, a rule or set of rules can be created and generalized, enabling the classification of the instance set highly possible precision. Research on FS has been since the early 1960s (see [27] or [103] for an overview and biographical notes).

Raw datasets by themselves are well-known for not providing much information. So, to extract useful knowledge or patterns, raw data must first be processed. To this end, deriving informative knowledge from these data calls for the development of new effective and efficient methods. This is becoming increasingly substantial [37]. Typically, many as deemed necessary as are features are involved to compose a real-world dataset. This is closely related to: i) knowledge and, in turn, essential features capturing this knowledge concerning the domain of interest; ii) the availability of these features; iii) the resources available for collecting these available features; and (iv) the resources available for storing, maintaining, and retrieving these features. Thus, it is obvious the unnecessary or insufficiency of the all features stored in the resulting dataset to learn the concerned concept. Assuming the presence of the relevant features in this data, FS is the art of electing a small feature subset ideally necessary and sufficient to describe the concept of interest [61].

FS is a recommended data-preprocessing phase in most machine learning tasks due to its potential to alleviate the curse of high dimensionality in large and complex datasets. This helps in reducing the data processing workload and decreasing the computational cost of classification, thus saving resources while enhancing the algorithm overall performance. There are a large number of applications that depend on high-dimensional datasets with a large number of features. Many of these features may be redundant or obsolete. Additionally, the database at hand may be inspected by a high noise level, which may result in degrading the overall classification performance and increasing computational cost [69]. Hence, proposing an efficient FS method is crucial to find the best subset of highly relevant and informative features while maintaining the representation of the original dataset with high classification accuracy as well. Ultimately, FS has three main advantages [46]: i) enhancing the predictor's performance;

ii) gaining a better understanding of the underlying process leading to faster data processing; and iii) giving more cost-effective prediction models.

1.1 Motivation

It is an NP-hard problem searching for a variable subset [19, 22, 36, 64]. For this reason, it cannot be guaranteed to acquire the optimal solution except when conducting an exhaustive search which is quite unfeasible for this situation. Meta-heuristic techniques, being not having to do a comprehensive search for the whole solution space, permits obtaining reasonably good solutions. The quality of each "heuristic" solution obtained is tightly based on the method adopted. In other optimization problems, meta-heuristic methodologies have been found so superior [23, 52, 56, 84, 110, 117]. In this field, we mention the Differential Evolution (DE) [11], Ant Colony Optimization (ACO) [58], Particle Swarm Optimization (PSO) [77], Genetic Algorithm (GA) [29], and most recently, Harris Hawks Optimization (HHO) [132], Whale Optimization Algorithm (WOA) [74], and Salp Swarm Algorithm (SSA) [35]. In real-world applications, people favor a timely obtaining of good solution rather than being obsessed with optimal solutions. Therefore, we have opted the meta-heuristic techniques due to their practicality in real-world applications.

Among various meta-heuristics, swarm-based optimization algorithms have been used to shatter the obstacle to choosing the best solution (optimal feature subset) to the FS problem [78]. The Sparrow Search Algorithm (SSA) is a novel swarm-based optimization algorithm recently introduced by Xue et al. [119] for solving continuous optimization problems. An improved binary version called iBSSA is implemented and introduced based on SSA for solving the problem of wrapper-based FS. In this study, the SSA algorithm was chosen for solving the FS problem, driven by a number of motivations. First, based on the rigorous discussion through Sect. 2, meta-heuristic algorithms have proven superior in solving the FS problem. Therefore, the performance of the new SSA algorithm, being a novel member of this toolbox, is tested. Second, SSA is a recently proposed meta-heuristic algorithm with the capabilities of fast convergence speed, stability, searching precision, and escaping from local optima [119], which is yet to be systematically applied to solving the problem of FS. Last, when the proposed iBSSA algorithm was compared with some high-performance optimization algorithms as well as other latest well-developed ones, it highly supported the above claim by providing higher efficiency (i.e., fewer number of iterations or less calculation time) and providing an optimal or near-optimal solution to most of the datasets involved. In order for the

FS problem to be dealt more effectively, this work proposes a novel improved binary SSA algorithm by mapping the continuous SSA into discrete values using a set of common S-shaped and V-shaped TFs, along with incorporating the two promising improvements, roaming agents' (sparrows') resetting strategy and the local search method.

On the other hand, k -Nearest Neighbor (k -NN), Support Vector Machine (SVM), and Random Forest (RF) are prevalent learning techniques in the machine learning paradigm. In this study, while iBSSA is adopted as a search optimization approach for finding out the most relevant features, the diverse classifiers adopted herein (i.e., k -NN, SVM, and RF) are used as fitness evaluation functions (or evaluators), in order to, at the last, compose a new wrapper FS method. This research will only address supervised FS methods. As mentioned, meta-heuristics have achieved great success in the FS domain. However, most of the proposed methods typically only considered the k -NN classifier and omitted SVM in many cases, while RF has been almost completely ignored, although SVM and RF typically provide better results than k -NN in diverse classification tasks [7, 25, 80]. Therefore, in this study, the presented iBSSA method is tested with the three referred classifiers which are used for estimating the performance of the proposed approach in terms of the fitness value inferred from the classification error rate.

A last motive, many researchers have turned a blind eye to further addressing the roaming behavior of individuals as well as the premature convergence problem in meta-heuristic optimization while solving the FS problem [1, 71, 82]. For this reason, the present work accounts for both re-positioning of roaming individuals to promote the search within feasible regions to amend the best solution randomly faster, combined with a local search algorithm for enhancing the exploitation capability.

1.2 Contribution

To improve the exploration of SSA as well as to promote its exploratory power within feasible regions, a Random Re-positioning of Roaming Agents (3RA) strategy was proposed. Further, a novel Local Search Algorithm (LSA) was also incorporated into the standard SSA algorithm to boost the exploitation process by enhancing the optimum solution obtained at the end of each main loop or iteration in the algorithm. First of all, iBSSA was tested over nine common S-shaped and V-shaped TFs, in order to determine the most effective one that allows iBSSA to exhibit the best convergence speed as well as the best averages of fitness,

accuracy, and number of selected features. Then, in pursuit of a fair comparison, the best-performing TF was introduced into the proposed iBSSA as well as into the binary versions of other competitor algorithms. To demonstrate the potential of the proposed approach to significantly select the most relevant features, such popular expert systems as k -NN, SVM, and RF were used to estimate the average fitness value based on the classification error rate averaged on 30 independent runs of the algorithm, over 18 multi-scale benchmark datasets carefully selected from the University of California Irvine (UCI) machine learning repository [38]. To sum up, the main contributions of this paper are outlined as follows:

1. An improved binary version of SSA (iBSSA), which is enhanced through the 3RA strategy and the LSA method, is proposed for the first time based on a feature transformation method for wrapper feature selection in classification tasks.
2. iBSSA was evaluated based on nine different TFs, including S-shaped and V-shaped functions, over 18 benchmark multifaceted, multi-scale UCI datasets.
3. For a fair comparison, the TF most effective with SSA was also integrated into other well-known meta-heuristics, including SSA, ABC, PSO, BA, GWO, WOA, GOA, SFO, HHO, BSA, ASO, and HGSO, for binary conversion, which revealed the supremacy of the proposed iBSSA method for feature selection in classification when compared to other competitor algorithms, over all three classifiers adopted in this study (i.e., k -NN, SVM, and RF), for the majority of the datasets used.
4. The final results were validated based on different performance metrics, including mean classification accuracy, mean fitness, mean number of selected features, along with the respective standard deviation values.
5. The supremacy of the proposed iBSSA method with the three classifiers was affirmed compared to competitors, based on a Wilcoxon's non-parametric statistical test at a significance level $\alpha = 0.05$.

1.3 Structure

The rest of the paper is organized as follows: Section 2 reviews the existing FS techniques; Section 3 presents a simplified formulation of the FS problem, a description of the original SSA, a brief discussion on the nine TFs, a discussion on the two improvements embedded into the

SSA, as well as a debate on the three classifiers adopted in this study: k -NN, SVM, and RF; Section 4, elaborates for and investigates the proposed iBSSA algorithm; Section 5 introduces the computational results and comparisons with a few prominent rival algorithms, and the conclusions as well as some suggested prospective works follows in Sect. 6.

2 An overview of feature selection methods

Dimensionality reduction is so crucial especially when experiencing a high-dimensional feature space, therefore it is commonly used in machine learning where a new space of reduced dimensions is produced by mapping the original feature space onto it. It is extremely important to identify relevant features in classification tasks (reducing computational costs while maintaining the highest possible accuracy), as well as to recognize the features' relative significance. Two different paths are usually taken for dimensionality reduction, constructing new dimensions or picking a subset of the original dimensions. Research literature has often distinguished feature selection from feature extraction. In feature selection, the aim is to individually—from among a large set of features—find the best features that can effectively boost the classification capability. More general methods, termed feature extraction algorithms, adopt transformation or combination techniques to create new features from the original feature set. Different weighting schemes have been traditionally used in feature extraction to generate a smaller number of ideally uncorrelated features either by linearly combining features (as in Linear Discriminant Analysis (LDA), and Principal Component Analysis (PCA)) or combining them non-linearly (as in neural networks).

To use feature selection or feature extraction is the first critical issue. Each has its own limitations, and no clear-cut evidence exists for the superiority of one to the other for diverse tasks. Feature extraction methods, including PCA [39]—which linearly combine all the available features to produce an, ideally uncorrelated, group of features—may struggle if some of the features are not heterogeneous at all, and there is no correlation between data assumptions. Similarly, it is unfeasible to use sequential search methods or do exhaustive search for high-dimensional datasets in FS. It is simple to examine features individually; however, poor feature subsets can be then yielded. On the other hand, it is too much time consumed, as well as running out of computer memory to try different feature combinations.

Finally, it is not necessarily that important features on training data exhibit the same importance level on test data.

As mentioned, the FS problem is the only focus in this work.

Much research on applications have concerned with feature selection (so-called variable selection) for which tens or hundreds of thousands of features are available in datasets. Machine learning tasks, such as time-series prediction, regression, classification, etc. may involve FS problems.

A resurgence of interest has been paid to applying FS methods to treat the large numbers of features encountered in different types of problems, for example:

- Information fusion of multiple sensors' data. For instance, both color and shape features have been merged in a trademark image database to provide a higher retrieval accuracy [54].
- Pooling the parameters from different mathematical models for classification purpose, as the integration of multiple models in [97].
- Discovering the hidden relationships correlating a plethora of features, as the data mining application in [89].

The effectiveness of an inference model can be enhanced using an appropriate feature selection. Feature selection can grant such significant merits as follows [69]: i) improving the overall performance (simplicity of rules, predictive accuracy, or speed of learning); ii) model selection via data visualization; and iii) dimensionality reduction and noise removal. Although many advantages are offered by feature selection, the risk of over-fitting or decreasing accuracy may be encountered. Thus, it should be well managed how to avoid these risks to get the desired classification performance.

2.1 Typical Feature Selection techniques

There are two typical procedures in FS techniques: exploring the feature space using an underlying search or ranking algorithm; and guiding the underlying algorithm by, for example, measuring the classification error rate in the form of a cost function. On how to evaluate this cost function, proposed approaches most important are described below. The universal algorithms of FS often belong to one of three modalities: filters, wrappers, and embedded [43, 63]. The filter-based approach [61, 65] does not refer to or learn from the target classifier while determining the fitness of an examined feature subset. The cost function is

evaluated independently of the target classifier in the following classification of independent datasets using the pre-selected subset of features. Rather, the cost function value can be computed based on a generic error estimation function guides the search for an optimal subset or the ranking of the individual features in the feature search space. This procedure is flawed by ignoring the potential effect of the feature subset on the learning algorithm. In contrast, in the wrapper-based approach [27, 120] the fitness of an examined feature subset is determined by underlying the subset to the design of the target classifier (used as a black box) to get an estimated classification error rate through ranking features in the subset based on their predictive power. Usually, wrappers approach can select more relevant subsets of features than filters approach and thus give better classification accuracy. However, each examined subset's fitness needs a more computationally intensiveness for fitness assessment than the filter-based methods in terms of the classification error estimation. Consequently, since the wrapper-based methods must evaluate many subsets, they are often criticized by “brute force” and an additional computational overhead may emerge as a result of the target classifier-based evaluations.

Obviously, in the case of the not-too large variables' number, exhaustive search can be adopted and performed under these two approaches. However, it is well-known that the problem is NP-hard [19, 22, 36, 64] and the computational intractability quickly results from the search. When it is too large the number of variables, high computational costs may be incurred by the approaches. In contrast to wrapper approaches, embedded methods determine the feature subset given the classifier design.

To solve the FS problem more pragmatically, other techniques have been adopted to find a good approximate solution, hopefully closest possible to the optimal subset. Some of the potential subsets of features are intelligently sequentially examined based on “forward” selection or “backward” elimination criteria to select the optimal subset amongst all the examined subsets based on the best cost function evaluation. In the following, the most commonly used methods—under these techniques—for performing FS are discussed. Sequential Forward/Backward Selection (SFS/SBS) [5] typically performs according to simple greedy deterministic heuristics. SFS [115] starts by an empty subset and new features are sequentially incorporated into this subset, whereas in SBS [75], the selection process is reversed, SBS begins with the original subset of

all features and sequentially removes the least promising ones until satisfying a certain termination criterion. Both methods suffer from what is called the “nesting effect”; that is, features once selected in the SFS method cannot be discarded later, whereas the features discarded in the SBS method cannot be re-selected. The result is the permanent suboptimality of these methods. In addition, high computation time is required by forward selection and backward elimination when the dimensionality is very high [63].

The nesting effect can be definitely avoided by fusing SFS and SBS. To prevent the “nesting” impact, a method called “plus- l -take-away- r ” has been suggested. This method [104] does by applying SFS followed by SBS l and r times, respectively. This forward and backward selection fixed cycle is iterated until reaching a predefined number of features. Thus, the nesting effect can be avoided by removing (in posterior steps) some of the features that have been priorly added. This method enables “fixed backtracking” based on top-down or bottom-up search and this is defined by l 's or r 's values. Although this procedure can partially overcome the problem of nested features, another problem arises: it is not theoretically easy to appropriately determine the land r 's value so that good enough solutions can—with a moderate amount of computational resources—be obtained. The plus- l -take-away- r method basically aims to counteract the nesting effect, and its implementation can be improved more efficiently by conditionally including and excluding features, guided by the value of the fitness itself.

In 1994, the concept of “floating feature search” was introduced by Pudil et al. [87]. In addition, Sequential Forward Floating Selection (SFFS) and Sequential Backward Floating Selection (SBFS) were introduced as two “floating” search methods. Floating selection methods are related to the plus- l -takeaway- r . Unlike the latter, the number of forward and backward steps is not beforehand fixed but controlled dynamically. In the methods of SFFS and SBFS, number of features included or excluded is changing at the different stages of the procedure. These two methods are probably the most effective FS techniques [53]. Mainly, in a forward search, a null feature set is initialized and, for each step, the current feature set contains the best feature that satisfies some fitness function (i.e., one SFS step is performed). At the same time, it is verified by the algorithm if it is possible to improve the fitness by excluding some feature. if so, the set is then refined by eliminating the worst feature, controlled by the

fitness value (i.e., one SBS step is performed). Therefore, the SFFS proceeds by dramatically tuning the number of features in subset (increasingly and decreasingly) until reaching the desired number. Similarly, SFBS initiates with the full feature set and uses SBS and SFS steps to perform the search until reaching the desired dimension. Even though these two methods cannot always get the best feature subset, they perform very well in comparison with other search methods.

Compared to the plus- l -take-away- r methods, floating search methods are particularly characterized by the potential to achieve good performance by making more than one sweep while pursuing the best subset of features. In practice, dynamic backtracking search performs very robustly and, among different feature set search methods, the first choice for many scholars would be the floating search procedures [86]. Plus, floating methods behave just as efficiently as the best sequential methods.

Despite the significant intelligence exhibited by the floating methods, they still suffer suboptimality. These sequential floating methods are criticized by being highly elected to fall into the local optimal solutions' trap, even if the problem scale is quite small and the fitness function is monotonic.

Other literature methods or algorithms adopted for the FS problem include the following:

- The Branch-and-Bound (BB) FS algorithm has been invented by Narendra and Fukunaga [81] to find the optimal feature subset. This method has one drawback: it requires the monotonousness of the FS criterion function. Thus, adding new features to a feature subset cannot enable to reduce the fitness function value. Unfortunately, it is seldom to satisfy the monotonic condition. Nevertheless, in large feature spaces, the computational cost is prohibitive: in the worst scenario, an exhaustive search is performed and the time exponentially becomes more complex in accordance with the expansion in feature space dimensionality. As pointed out in [53], it is still impractical to apply the BB method to very-large-scale problems.
- The Max–Min (MM) FS method has been invented by Backer and Shipper [10]. It is a computationally efficient method in which only individual and pairwise merits of features are evaluated. This method invariably achieves unsatisfactory results [53, 62].
- The use of GA for FS has been firstly introduced by Siedlecki and Sklansky in 1989 [103]. In a GA

approach, a binary string (a “chromosome”) is used to represent a given feature subset of total length D , with a value of one or zero in the j -th position to denote the presence or absence of feature j in the set. The algorithm maintains a population of chromosomes and the “fitness” of each chromosome is evaluated in terms of how likely is the survival and breed of the chromosome into the next generation. Old chromosomes are used to create new ones by the following processes: i) crossover, where offspring is created by mixing parts (genes) of two different parent chromosomes; and ii) mutation, where a child is created by randomly disturbing the bits of a single parent. Other works that apply GA to the FS problem include those of Emary et al. [32], Jiang et al. [55], Raman et al. [90], Dong et al. [29], Bouktif et al. [14], and Das et al. [21].

- The Tabu Search (TS) method has been developed for feature subset selection by Zhang and Sun [128]. An adaptive memory (the tabu list) is used by TS to keep track of solutions that have been visited and should—for a number of iterations—be avoided. The “tabu tenure” determines how long the tabu list should reserve a solution. In [128], a comparative analysis of the TS-based technique and other FS techniques (SFS, SBS, plus- l -take-away- r , SFFS, SFBS, and GA) has been conducted and the performance analysis shows TS as a promising search “heuristic” for the FS problem.

Despite some progress, it is not yet completely satisfactory the FS techniques available now for large feature sets. They are either enriched with computational feasibility but far from optimality, or they are enriched with optimality (or near-optimality) but cannot cope with the computational complexity of real-world FS problems. More powerful methods for FS are required to be developed, to provide very good results with more efficient computations. Therefore, in the present work, a novel improved meta-heuristic strategy, called iBSSA, is proposed to solve the FS problem.

Several challenges usually arise when getting up to find an optimal subset of features using the aforementioned traditional techniques. Multiple search methods, such as breadth search, random search, depth search, or hybridization of them have been additionally tried to determine the optimal subset of features. Further techniques have been proposed in [2, 3]. However, exhaustive search technique typically adopted there has been found time-consuming and unfavorable, especially with high-di-

mensional datasets [106]. Feature selection is arguably formulated as a combinatorial NP-hard optimization problem [19, 22, 36, 64]. For example, if we suppose that the number of features in a given dataset is D , it will be computationally expensive to find the optimal subset of features among 2^D different candidate combinations (solutions), given the fact that 2^D different evaluations are required for conducting a full search, methods of forward selection or backward elimination are recursively used in SFS and SBS. In addition, local optima can be more efficiently avoided by using stochastic methods. Therefore, those methods, especially meta-heuristics (discussed below), have been increasingly attempted by researchers as wrapper-based approaches to find out the best feature subset based on their global search process that is significantly effective with large feature spaces [4, 101, 124].

2.2 Meta-heuristics-based feature selection techniques

In recent years, meta-heuristic optimization algorithms, which are very close to stochastic methods in generating and using random variables, have been incorporated into different application areas, including building construction, health-care, agriculture, computer engineering, and many more [117]. This may be due to the obvious merits of these algorithms, including gradient-free nature, flexibility, simplicity, and independency of the problem of interest [51, 79]. Moreover, these algorithms can potentially approximately find the best (optimal) solution in small, comparable amounts of time, even for very large-scale problems. There is therefore still an appeal of solving complex optimization problems, such as the FS problem, using different advanced meta-heuristic algorithms [28]. Among many others, swarm intelligence and physics-based methods are two major categories of meta-heuristic algorithms [50, 130, 131]. Swarm intelligence mimics the collective social behavior of various species (i.e., animals or insects) in swarms for hunting or foraging, in which all individuals cooperate and share information during the optimization process. Representative algorithms include Artificial Bee Colony (ABC) [57], PSO [30], Bat Algorithm (BA) [122], Grey Wolf Optimizer (GWO) [79], Firefly Algorithm (FA) [121], and Cuckoo search (CS) [123]. This latest algorithms in this category include the WOA [78], Bird Swarm Algorithm (BSA) [76], HHO [49], Sailfish Optimizer (SFO) [98], Grasshopper Optimization Algorithm (GOA) [93], and Butterfly Optimization Algorithm (BOA) [9]. Moreover, the laws of physics in nature were utilized to come up with physics-based optimization methods, including Simulated Annealing (SA) [112], Gravitational Search Algorithm (GSA) [91], Atom Search

Optimization (ASO) [133], and Henry Gas Solubility Optimization (HGSO) [47]. To learn more about meta-heuristic algorithms, the reader is referred to [28].

In an attempt to tackle the FS problem, meta-heuristics methods have been, due to their simplicity and flexibility, tried by many researchers, and some of them are investigated here. Sharawi et al. [99] introduced a new variant of the WOA algorithm that employed the wrapper-based technique to select the optimal feature number to yield the best mean classification accuracy. Subsequently, Eid [31] suggested the S-shaped function with WOA for tackling FS problems. Two versions of WOA were introduced by Mafarja and Mirjalili [70]. In the first, instead of a random operator, the roulette wheel and tournament selection were applied. The mutation and crossover operators were used in the second version to boost the performance of the proposed algorithm. Furthermore, Sayed et al. [94] hybridized WOA with the chaotic search to solve the slow convergence speed and stagnation to local optima issues potentially encountered while solving FS problems. Mafarja et al. [73] introduced two binary versions of the GOA algorithm. The first method used the S-shaped and V-shaped functions as Transfer Functions (TFs), while the second version employed the mutation operator to improve the exploratory ability of the proposed algorithm. Mafarja et al. [72] employed evolutionary population dynamics and selection operators to enhance the performance of the conventional GOA. Zakeri and Hokmabadi [125] hybridized the GOA with some statistical metrics for replacing the duplicated features with the most favorable one. In a related context, Emary et al. [32, 33] introduced the GWO algorithm and two binary versions of the Lion Optimization Algorithm (LOA), based on S-shaped and V-shaped TFs. Additionally, Arora and Anand [8] proposed a novel binary variant of the BOA based on some of the above-mentioned TFs. Chen et al. [17] introduced a discrete PSO algorithm based on a logistic map sequence to improve the swarm diversity. De Souza et al. [24] suggested a new variant of Crow Search Optimization (CSO) algorithm using a V-shaped TF. Sayed et al. [96] hybridized the Flower Pollination Optimization Algorithm (FPOA) with the clonal selection search, in which the optimum path forest accuracy was applied to evaluate the solution performance. Zawbaa et al. [127] managed to integrate the rough theory into FPOA to solve FS. Sayed et al. [95] validated the performance of 10 chaos maps incorporated into both the CSO and SSA algorithms. Hegazy et al. [48] introduced a new control parameter which was involved to refine the best solution obtained so far, using the k -Nearest Neighbor as an expert system for the evaluation of the relevance of the selected features. Moreover, Zhang et al. [129] hybridized the FA and SA algorithms with the aim to escape from the local optima while increasing the accuracy

of the final solutions. Several enhancements, such as return-cost attractiveness, Pareto dominance-based selection, and binary movement with adaptive jump, were employed in [129] in order to solve the underlying FS problems more effectively. Faris et al. [34] introduced a multi-verse optimization algorithm, adopting the Support Vector Machine (SVM) classifier as a fitness evaluator. Gu et al. [41] suggested a new version of the PSO algorithm to solve a high-dimensional FS problem. Many other meta-heuristic based approaches were proposed for FS by, for example, Hafez et al. [44], Kashef and Nezamabadi-pour [58], Li et al. [67], Tabakhi et al. [105], Wang et al. [114], and many others. In fact, the introduction of meta-heuristic algorithms to solve FS problems is a trend for some years now. To learn more about meta-heuristic algorithms for FS, please refer to the review paper [100]. As discussed earlier, there are various advantages of meta-heuristics, so will the FS problem be solved using existing methods sufficiently? The No Free Lunch theorem (NFL) [116] is the appropriate answer to this question. This theorem indicates that a single algorithm cannot optimally solve all optimization problems. For FS on a dataset, while an algorithm performs very well for one kind of dataset, another one may perform poorly. Thence, the search for an advanced meta-heuristic approach for solving almost all possible FS dataset types is still an open research question.

3 Preliminaries

3.1 The Feature Selection (FS) problem

The problem of selecting a feature subset can be formulated as a combinatorial optimization problem with the major aim of obtaining superior classificatory performance as follows: Let an initial feature set, \mathbf{f} , with cardinality D . Let the number of feature to be pursued be represented by d in the selected subset \mathbf{s} . Let the fitness function of the selected subset \mathbf{s} be represented by $f(\mathbf{s})$. Let the misclassification error rate, $c(\mathbf{s})$, of a given classifier to be minimized when presented with the feature subset \mathbf{s} . Formally, the FS problem is to find a subset $\mathbf{s} \subseteq \mathbf{f}$ such that $|\mathbf{s}| = d$ and

$$f(\mathbf{s}) = \min_{\mathbf{s} \subseteq \mathbf{f}, |\mathbf{s}|=d, d < D} c(\mathbf{s}).$$

So, a lower value of $c(\mathbf{s})$ indicate a highly feasible feature subset. In this case, the “goodness” of a selected subset of features is assessed and computed in the form of a fitness function as follows: Let C be a set of cases (i.e., the whole

instances or samples in a given dataset). We know the class to which each case relates (here binary classifications, with only two classes, is considered). A certain partition is made in C , $C = C_1 \cup C_2$, where C_1 (training data) and C_2 (test data) are in the vicinity of the same cases’ number and the same representation ratio for each class. The Euclidean distance is calculated from each case in C_2 to every case in C_1 and the class corresponding to nearest case is assigned. $f(\mathbf{s})$ ’s value is the proportion of mishits in the assigned classes. That is, how much times the assigned class was NOT the real (true) class.

If we adopted an exhaustive approach that is applied to this problem, this would require investigating all $\binom{D}{d}$ possible d subsets (which can be derived from the feature set \mathbf{f}). But the exponential growth of possibilities makes exhaustive search impractical even with reasonable D ’s values. As mentioned, FS is an NP-hard problem [19, 22, 36, 64] and heuristic and meta-heuristic techniques are therefore the most appropriate choice.

3.2 Sparrow Search Algorithm (SSA)

Sparrow Search Algorithm (SSA) [119] is a meta-heuristic optimization algorithm under the umbrella of swarm intelligence, meta-heuristics and computational intelligence. SSA is a natural extension of the BSA algorithm [76] extracted from the social behavior and interactions of bird swarms. Sparrows are gregarious birds of several species distributed across the globe and prefer to live in areas where people live. Besides, it belongs to the family of omnivorous birds that usually feed on weed or grain seeds. The sparrow is intelligent in comparison with other small birds and has a strong memory inspired by behaviors of anti-predation and foraging. In fact, captive house sparrows are divided into two main types based on their foraging behaviors: scrounger and producer [12]. The producers energetically pursue potential sources of food, while the scroungers grab food via those producers. In addition, it has been evidenced that birds usually switch between producing and scrounging by utilizing behavioral strategies flexibly [13]. With that being said, sparrows usually use the same strategies which are adopted by producers and scroungers to find their food [68].

Literature studies have shown that each individual in SSA keeps monitoring the behavior of its neighbors. Meanwhile, attackers against the bird flock are competing for foraging high intakes of food for the companions, thereby increasing their own predation probability [66].

Though, when different foraging strategies are adopted by the sparrows, the energy retained by the individuals may be then well utilized in the pursuit of food, so that the scrawny sparrows can scrounge even more [66]. It should be noted that birds within the search space are more vulnerable to attack by predators and therefore are required to seek a better (safer) location [15]. While the birds at the centre can minimize their domain of danger by moving closer to the neighbors [45]. It is well known that the bird family, especially sparrows, shows a natural instinct of curiosity along with constant vigilance. For example, when a predator is detected by a bird, a chirp is given by one or more individuals so that the entire group flies away from the source of danger [88]. According to the previous description of sparrows, a mathematical model can be formulated to construct the SSA algorithm. In simulation experiments, the food source is pursued by using virtual sparrows with a position vector expressed as:

$$\mathbf{x} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,D} \\ x_{2,1} & x_{2,2} & \dots & x_{2,D} \\ \vdots & \vdots & \dots & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,D} \end{bmatrix}, \tag{1}$$

where the number of sparrows are represented by N , and the number of dimensions to be optimized are represented by D . Then, the following vector can represent the fitness value of all sparrows:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \dots & x_{1,D}]) \\ f([x_{2,1} & x_{2,2} & \dots & x_{2,D}]) \\ \vdots & \vdots & \dots & \vdots \\ f([x_{N,1} & x_{N,2} & \dots & x_{N,D}]) \end{bmatrix}, \tag{2}$$

where the value of each row in $\mathbf{f}(\mathbf{x})$ represents each individual's objective function. In SSA, the producers with the best fitness for getting food during the search process are given a higher priority. Besides, because producers are primarily responsible for foraging, in addition to managing the movement of the entire swarm, they have the ability to forage for food in a wider range than the scroungers do. At each iteration t , the position of each producer i is adjusted per every dimension j as:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right) & \text{if } R_2 < ST, \\ x_{ij}^t + Q \cdot L_{ij} & \text{if } R_2 \geq ST, \end{cases} \tag{3}$$

where $j \in \{1, 2, \dots, D\}$, and T represents the maximum number of iterations. $R_2 \in [0, 1]$ represents a warning

(alarm) value, while $ST \in [0.5, 1.0]$ denotes the safety threshold. $\alpha \in (0, 1]$ and the normally distributed Q are two independent random numbers. L is a matrix of $1 \times D$, inside which each element is 1. $R_2 < ST$ means no predators around, allowing producers to search wider. On the contrary, $R_2 \geq ST$ implies the existence of predators discovered by some sparrows, and because of this, all sparrows are needed to quickly move to more secure regions.

As mentioned above, some scroungers continue to track down producers until those producers find a good food source, so that they can then leave their current position to pursue the target source of food. If they could win, they would automatically have a chance to get food from the producers; otherwise, they continue to execute their main task of vigilance. The scroungers update their position according to Eq. (4):

$$x_{ij}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_{worst,j}^t - x_{ij}^t}{i^2}\right) & \text{if } i > N/2, \\ x_{pj}^{t+1} + |x_{ij}^t - x_{pj}^{t+1}| \cdot A_{0j}^+ \cdot L & \text{if } i \leq N/2, \end{cases} \tag{4}$$

where $x_{p,j}$ is the personal best position obtained so far by the producers at the j -th dimension, and $x_{worst,j}$ denotes the current global worst location at the j -th dimension. A shows a matrix of $1 \times D$ with the randomly assigned value 1 or -1 for each of its elements, while $A^+ = A^T(AA^T)^{-1}$. L represents a matrix of $1 \times D$, all of whose entries are 1. When $i > N/2$, this implies the starvation of the i -th scrounger whose fitness value is among the worst.

In the simulation experiments, those sparrows who are aware of the danger (i.e., expectant sparrows) typically account for 10% to 20% of the whole swarm. These sparrows' positions are initially randomly generated through the swarm as:

$$x_{ij}^{t+1} = \begin{cases} x_{worst,j}^t + \beta \cdot |x_{ij}^t + x_{best,j}^t| & \text{if } f_i > f_g, \\ x_{ij}^t + K \cdot \left(\frac{|x_{ij}^t + x_{worst,j}^t|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g, \\ x_{ij}^t & \text{if } f_i < f_g. \end{cases} \tag{5}$$

The algorithmic structure of the standard SSA is exhibited in Algorithm 1.

Algorithm 1 Standard SSA algorithmic structure

Input:

- T – maximum number of iterations
- N – total number of sparrows
- PD – number of producers
- SD – number of scroungers
- ST – safety threshold

Output:

- \mathbf{x}_{best} – the overall best location found so far
- f_g – the global best fitness so far

```

1: Start
2: Initialize a swarm of  $N$  sparrows and its relevant parameters;
3:  $t \leftarrow 0$ ;
4: while  $t < T$  do
5: Rank the fitness values  $\mathbf{f}(\mathbf{x})$  (see Eq. (2));  $\triangleright$  Rank the producers in ascending order based on their fitness values
6: Find the current personal best and worst producers  $\mathbf{x}_p$  and  $\mathbf{x}_{worst}$ ;  $\triangleright$  First and last individuals in the ranking,
   respectively
7:  $R_2 \leftarrow$  a random value in  $[0, 1]$ ;
8: for producer  $i = 1, 2, \dots, PD$  do
9: Update the producer's location using Eq. (3);
10: end for
11: for sparrow  $i = PD + 1, PD + 2, \dots, N$  do
12: Update the sparrow's location using Eq. (4);
13: end for
14: for scrounger  $i = 1, 2, \dots, SD$  do
15: Update the scrounger's location using Eq. (5);
16: end for
17: Find the current new location  $\mathbf{x}_i^{t+1}$ ;
18: if  $f(\mathbf{x}_i^{t+1}) < f(\mathbf{x}_i^t)$  then  $\triangleright$  If the current new location is better than before, update it
19:  $\mathbf{x}_i \leftarrow \mathbf{x}_i^{t+1}$ ;
20:  $f_i \leftarrow f(\mathbf{x}_i^{t+1})$ ;
21: end if
22: Re-rank the whole swarm in ascending order based on the fitness values  $\mathbf{f}(\mathbf{x})$  (see Eq. (2));
23: Find the current global optimum position  $\mathbf{x}_{best}^{t+1}$ ;  $\triangleright$  First individual in the ranking
24:  $\mathbf{x}_{best} \leftarrow \mathbf{x}_{best}^{t+1}$ ;
25:  $f_g \leftarrow f(\mathbf{x}_{best})$ ;
26:  $t \leftarrow t + 1$ ;
27: end while
28: End

```

3.3 Transfer Functions (TFs)

Since the final solution obtained by SSA is composed of continuous values, SSA cannot directly be used to solve a FS problem. Therefore, it is necessary to use a mapping (transfer) function to transform the continuous values into binary 0 or 1. Transfer Functions (TFs) [77] determine the rate of changing in the values of the decision variables from 0 to 1 and back. On selecting a TF to transform the values from continuous to binary, some concepts should be taken into consideration—from an SSA perspective—as follows:

- The range of values obtained from a TF should be within the interval $[0, 1]$, representing the probability whether a sparrow will change its current location.
- If the alarm value R_2 is lower than the safety threshold ST , then the TF should present a higher probability of changing the current location at the next iteration as sparrows having R_2 larger than ST are probably going so far from the best solution.

- When R_2 is small, the TF should provide a small probability of changing the current location.
- In summary, the probability returned by the TF should increase as R_2 approaches ST , so that sparrows that are moving away from the best solution can have a higher probability of changing their location vector, which enables them to return to their previous best personal location as soon as possible throughout the next iterations.
- Also, the probability obtained from a TF should decrease as R_2 value keeps away from ST value.

These concepts judge the high capability of TFs to map the process of continuous search into binary for each individual \mathbf{x} , using Eq. (6):

$$\left(x_{ij}^{t+1} \right)_{bin} = \begin{cases} \begin{cases} 0 & \text{if } rand < TF(x_{ij}^{t+1}), \\ 1 & \text{if } rand \geq TF(x_{ij}^{t+1}), \end{cases} & \text{if TF is S-shaped,} \\ \begin{cases} \neg(x_{ij}^t)_{bin} & \text{if } rand < TF(x_{ij}^{t+1}), \\ (x_{ij}^t)_{bin} & \text{if } rand \geq TF(x_{ij}^{t+1}), \end{cases} & \text{if TF is V-shaped,} \end{cases} \tag{6}$$

where $(x_{ij}^{t+1})_{bin}$ represents the j -th dimension of the i -th individual at the current iteration $t + 1$, $rand$ is a number selected randomly from within the range $[0, 1]$, and $TF(x_{ij}^{t+1})$ is the probability value obtained when applying a given TF to every j -th component's continuous value of agent i . It is clear from Eq. (6) that we have two cases: i) if the TF is S-shaped, then if $rand$ is less than the probability returned by the involved TF, the j -th dimension of the original individual is set to 0; otherwise, it is set 1; and ii) if the TF is V-shaped, then if $rand$ is less than the probability returned by the involved TF, the j -th dimension is negated; otherwise, it remains unchanged. Thus, by using the S-shaped and V-shaped TFs and Eq. (6), continuous variables are successfully mapped into binary.

Table 1 reports two families of TFs while Fig. 1 exhibits visually their behavior, divided into S-shaped and V-shaped transfer functions. Here, it should be pointed out that the proposed iBSSA method was evaluated based on those nine TFs whose mathematical expressions are shown in Table 1.

3.4 Learning algorithms adopted in this study

k -NN, SVM, and RF are widespread learning algorithms from different families in the machine learning paradigm. In this article, while iBSSA is used as a search optimization strategy, those classifiers are used to design a new wrapper FS model to assess the goodness of each feature subset. Thence, these classifiers (or expert systems) are described as follows:

3.4.1 k -Nearest Neighbor (k -NN)

k -Nearest Neighbor (k -NN) [18] is a largely popular pattern recognition and machine learning algorithm. It is commonly used due to its advantage of simplicity in implementation over other complicated supervised machine

learning algorithms [118]. In pattern classification techniques, k -NN is applied in diverse fields, such as health-care, forestry, image and video recognition, finance, and many more. The k -NN algorithm can be considered as a wrapper method in which the rules of classification are produced by training instances. After k -NN learns from the training process, the unknown instances in the test set are approximated based on their vicinity to the instances of the training set, so that an unlabeled instance can be subsequently classified according to the largest probability of category. Though, the selection of k in k -NN is crucial and it is therefore chosen after various trial and error experiments. In this study's empirical experiments, the selected feature subsets are validated by using the k -NN classifier ($k = 5$ [70, 71, 108]) with the Euclidean distance metric.

3.4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) [113] is a widely used wrapper-based classifier, in which hyper-planes are applied for separating multiple classes. SVM is widespread in the data science community as it has the potential to classify with reliable accuracy and less computational resources as well. This is achieved by mapping the primary data from the original input space – using the non-linear function ϕ — into a higher dimensional space wherein linear separation of the data can take place by finding a hyper plane with the maximal margin in this higher dimensional space > 0 for discovering the boundaries between the input classes. However, this particular approach faces two main critical challenges: acceptable basic function selection as well as the adjustment of its parameters [109]. Logically, selecting the best decision plane is mainly treated in the form of an optimization task that helps a kernel function find out the most optimal space wherein categories are often divided linearly through one non-linear transformation.

In SVM models, given that $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ denotes the kernel functions, there are a number of common kernel functions, including:

- Linear kernel, where $\phi = x_i * x_j$,

Table 1 S-shaped and V-shaped families of TFs

S-shaped family		V-shaped family	
Name	Transfer function	Name	Transfer function
S_{v1}	$TF(x) = \frac{1}{1+exp^{-x}}$ [60]	V_{v1}	$TF(x) = \tanh(x) $ [92]
S_{v1c}	$TF(x) = \frac{1}{1+exp^x}$	V_{v2}	$TF(x) = \left \operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right) \right $
S_{v2}	$TF(x) = \frac{1}{1+exp^{(-x/2)}}$	V_{v3}	$TF(x) = \left (x)/\sqrt{1+x^2} \right $
S_{v3}	$TF(x) = \frac{1}{1+exp^{(-x/3)}}$	V_{v4}	$TF(x) = \left \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right $
S_{v4}	$TF(x) = \frac{1}{1+exp^{-2x}}$		

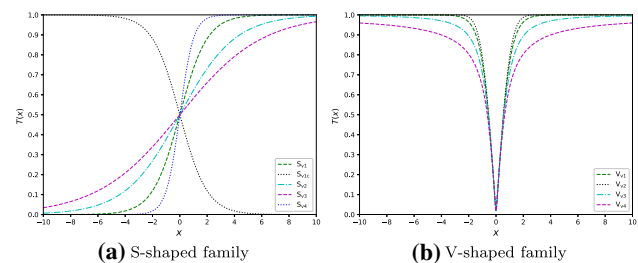


Fig. 1 Families of transfer functions

- Polynomial kernel of degree d , where $\phi = (x_i \cdot x_j + 1)^d$,
- RBF kernel, where $\phi = \exp(-\|x_i - x_j\|^2/2\sigma^2)$, and
- Sigmoid kernel, where $\phi = \tanh(x_i \cdot x_j + 1)$.

Typically, the polynomial and RBF (Radial Basis Function) kernels are more applicable and suitable for non-linear problems in diverse areas, including intrusion detection, classification, and image processing as they have proven efficient in delivering a better performance. Polynomial kernel provides a high performance—especially on high dimensional databases—by consuming less computational time. According to the works [83] and [102] penetrating the fields of satellite data analysis and intrusion detection, the results exhibit a slight advantage for the polynomial kernel in comparison with the RBF kernel. Additionally, multiple works judge that the optimum value to be assigned to the polynomial degree d should be 2, which would exclude an extensive seeking process regarding this parameter, generating however higher accuracy. On the other hand, the sigma value σ in the RBF kernel makes a greater impact both on the mapping transformation of data space and the ultimately obtained average classification accuracy. Also, the mandatory tuning process of RBF's controlled parameters complicates the situation because of the much time required. Based on the explained reasons, the polynomial kernel with $d = 2$ was chosen as the most adequate kernel type for SVM. However, to optimally solve the above-mentioned dilemma, a hyper-heuristic mechanism can be incorporated for the automatic selection of kernel types as well as the tuning of controlled parameters.

3.4.3 Random Forest (RF)

Random Forest (RF) [126] is a well-known machine learning algorithm which typically gets involved in heavy-duty tasks, including image classification, action recognition and detection, visual tracking, facial expression recognition, label distribution learning, time-series forecasting, and so on. RF is formally defined as an ensemble of decision trees featuring characteristics, such as robustness to label noise, inherent multi-class handling capacity, enabled FS, parallel processing, fewer tunable parameters, and efficiency in handling numerical and categorical data. Although RF has been proposed two decades ago, it is still employed in various applications [16], thanks to its simplicity in interpretability and implementation as well as the significant computational performance [59]. Decision trees in RF employ recursive partitioning of the training data into small sets, which greatly assists in the classification task by optimizing an impurity criterion, such as information gain or “*gini*” index [20].

Frankly, the large number of trees (estimators) in the forest increases the execution time of this algorithm, making it ineffective for real-time applications where the run time is an important factor. Moreover, the maximum depth of the tree (height of the tree) is a critical parameter. Once again, we can overcome this dilemma by adopting a hyper-heuristic approach for automatic selection of the split strategy and tuning of the algorithm's parameters. In our experiments, in order to achieve a relatively high performance in terms of higher classification accuracy, the RF algorithm was implemented and executed extensively with a number of estimators ($n = 10$) and maximum depth ($d = 5$).

4 Proposed improved Binary SSA (iBSSA) for feature selection

Due to the high performance derived from the extensive design of the canonical SSA algorithm in balancing exploration and exploitation capabilities and the absence of this algorithm to solve FS problems, this article adopts SSA as a search strategy for a wrapper-based FS. In the successive versions of SSA, any point in the search space can be the optimum position within the swarm. While FS is a discrete optimization problem, and only binary values (0 or 1) can be used to represent the components of the sparrow position in the binary SSA variant based on TFs. The underlying motivation is the appeal of the simplicity of those binary operators over continuous ones. An improved binary SSA variant is presented in this study to solve the feature selection problem. The method for binary conversion of continuous SSA mainly depends on the TFs. It is worth mentioning that two improvements were also introduced to straighten the roaming behavior of the algorithm as well as to enhance its exploitation capability.

The two improvements embedded into the standard SSA algorithm are described through in this section. The first is the 3RA method that is proposed in order to amend the optimum solution randomly faster to become within the feature space. The use of LSA algorithm is the second improvement which enhances the exploitation phase of SSA to avoid becoming stuck in local optima. As shown in Algorithm 4, the proposed iBSSA algorithm works by initially generating a swarm of N sparrows. Then, the main loop is iterating through these N sparrows to adjust their own locations according to Eqs. (3), (4), or (5). Eventually, LSA is applied to the current \mathbf{x}_{best} at the end of the iBSSA main loop, hoping to obtain a better solution than the current best one obtained so far. At the end, iBSSA will return the best optimum solution composed of \mathbf{x}_{best} and f_g .

4.1 Improvements embedded into the standard SSA

To improve the exploration of iBSSA as well as to rectify the roaming behavior usually exhibited by individuals in the swarm, the 3RA strategy was proposed. Further, a novel LSA method was incorporated into the standard SSA algorithm to boost the exploitation process by enhancing the optimum solution obtained so far at the end of each main loop or iteration of the algorithm. These two arguably promising improvements are discussed in detail in the subsection to come.

4.1.1 Improving the exploration by Random Re-positioning of Roaming Agents (3RA)

While refreshing the area of sparrows, they may abuse the feature space limit, especially with the high dimensionality of the FS problem at hand. In this way, an appropriate clipping technique is expected to be significant to inhibit roaming individuals (sparrows) which go beyond the search space. In general, numerous past investigations have added this upgrade step in their works [111]. Notwithstanding, which technique has been utilized to re-position roaming agents to the search space (i.e., re-positioning to the hunt space by cutting the detachable situation to stretch edges, irregular re-positioning to the pursuit area, or any others) is not clearly stated. Irrespective of that, we think that random re-positioning of roaming agents to an arbitrary area of the hunt space would provide improved out-

original SSA. The pseudo-code for the 3RA strategy is presented in Algorithm 2.

Algorithm 2 The 3RA strategy

```

 $[x_{\min}, x_{\max}]$  – continuous search domain, in our state,
 $[-1, 1]$ 
 $\mathbf{x}_i^t$  – the current position of  $\mathbf{x}$ 
for dimension  $j \in \{1, 2, \dots, D\}$  do
    if  $x_{i,j}^t < x_{\min} \vee x_{i,j}^t > x_{\max}$  then
         $x_{i,j}^t \leftarrow U(x_{\min}, x_{\max});$ 
    end if
end for
return  $\mathbf{x}_i^t$ ;

```

4.1.2 Improving the exploitation by Local Search Algorithm (LSA)

A new LSA algorithm is developed and presented as shown in Algorithm 3. In the original SSA, in the tail of each current iteration $t + 1$, LSA is called to enhance the currently obtained best solution \mathbf{x}_{best}^{t+1} by further omitting potentially irrelevant features still there. At first, LSA stores, in a variable *Temp*, the value of \mathbf{x}_{best}^{t+1} produced at the end of each iBSSA iteration. To improve *Temp*, LSA runs iteratively *LT* times. At each iteration *Lt* of LSA, four features *rand_feat* are randomly selected from *Temp*. LSA flips the value of each variable in *rand_feat*. Then, the fitness value $f(Temp)$ of the new solution (the new *Temp*) is evaluated; if it is better than $f(\mathbf{x}_{best}^{t+1})$, then \mathbf{x}_{best}^{t+1} is set to *Temp*; otherwise, \mathbf{x}_{best}^{t+1} and f_g are kept unchanged.

Algorithm 3 The proposed LSA algorithm

```

LT – maximum number of local iterations
 $\mathbf{x}_{best}^{t+1}$  – the best position so far at the end of iBSSA current iteration  $t + 1$ 
Temp  $\leftarrow \mathbf{x}_{best}^{t+1}$ ;
Lt  $\leftarrow 0$ ;
while Lt < LT do
    rand_feat  $\leftarrow$  four features randomly selected from Temp;
    for feature  $a \in rand\_feat$  do
         $a \leftarrow \neg a$ ;
    end for
     $f(Temp) \leftarrow$  Calculate the fitness value of Temp;
    if  $f(Temp) < f(\mathbf{x}_{best}^{t+1})$  then
         $\mathbf{x}_{best}^{t+1} \leftarrow Temp$ ;
         $\mathbf{x}_{best} \leftarrow \mathbf{x}_{best}^{t+1}$ ;
         $f_g \leftarrow f(\mathbf{x}_{best})$ ;
    end if
    Lt  $\leftarrow Lt + 1$ ;
end while
return  $\mathbf{x}_{best}, f_g$ ;

```

comes, inspired by the stochastic nature of meta-heuristics. Therefore, we have adopted it as an upgrade step that is supposed to combat the misuse cycling of sparrows and correct the haphazardness potentially caused by the

Furthermore, the classification techniques *k*-NN, SVM, and RF, were adopted to work with the proposed iBSSA algorithm in wrapper mode for solving FS problems. At each iteration, each classifier is individually applied to train

and test the data with the best subset of features selected by iBSSA. To represent the selected and deselected features in FS problem, binary values were used, such that each “1” value in the solution means this index is a selected feature, while “0” value means this index is an deselected feature. The proposed iBSSA algorithm operates as follows:

4.2 Initialization

The initialization phase is the first step in a population-based algorithm or approach, wherein a swarm of N sparrows (search agents) is generated randomly. In this study, each candidate solution i is limited by lower and upper bounds in the range $[-1, 1]$, in order to allow individuals to relatively search more broadly, yet restrictively within the continuous search space domain. Each search agent represents a potential solution with a dimension D which, in the paradigm of feature selection, equalizes the original number of features in any of the datasets used. The problem of FS for classification purposes can, in brief, be described as selecting a minimal relevant feature subset that can help maximize (or even maintain) the mean classification accuracy. Therefore, in this study, we manage to identify the salient features (“one” values) and reject the other ones (“zero” values). Initial position of each sparrow in the swarm is firstly converted to random binary values (among 0s and 1s) by discretizing the position at each dimension which takes either 0 (not selected) or 1 (selected) according to Eq. (6) and as shown in Fig. 2, prior to initiating the fitness evaluation process.

4.3 Fitness evaluation

When it is required to optimize more than one objective simultaneously, the underlying optimization problem is then described as many- or multi-objective, and all objectives must be satisfied to obtain the target optimum solution. Accordingly, the FS problem can be considered multi-objective as it must achieve two conflicting targets: minimizing the number of selected features and maximizing (or even maintaining) the classification accuracy, given a wrapper classifier [1, 85]. The FS method mainly aims at achieving high classification accuracy with as minimal feature subset as possible. The aforementioned two conflicting objectives are formulated in this work by adopting an objective function to evaluate the optimality of the solution obtained at the end of each iteration:

F_1	F_2	F_3	F_4	...	F_{D-2}	F_{D-1}	F_D
1	1	0	0	...	1	0	1

Fig. 2 Binary solution representation

$$Fit_i = \alpha \times Err_i + \beta \times \frac{|d^*|}{|D|}, \quad (7)$$

where Err_i is the classification error rate produced by wrongly predicting instances in a test set, and is computed based on the classifiers adopted herein; i.e., k -NN, SVM, and RF. While $|D|$ represents the number of all features in the original dataset, $|d^*|$ denotes the length of the selected feature subset. As for α and β , they respectively weigh the importance of both the classification accuracy and the selected feature subset’s size, where $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$. Based on extensive experiments in previous studies [40, 71], it has been prescribed that $\alpha = 0.99$ and $\beta = 0.01$ is an adequate adjustment. The large impact and weight should highly be given to the classification accuracy rather than the number of selected features. Intuitively, if only the classification accuracy is considered in the evaluation function, the result may then disregard the solutions having the same accuracy but with less number of selected features that contribute greatly to alleviating the dimensionality curse problem. In this study, we utilize the classification methods k -NN, SVM, and RF as expert systems (or evaluators) to assess the process of feature selection based on the classification error rate.

4.4 Position updating

Inclusively, the positions of sparrows in iBSSA are updated according to Eqs. (3), (4), or (5). If the current sparrow is a producer, then Eq. (3) is applied to update its position; otherwise, Eq. (5) is applied to update the scrounger’s position. Meanwhile, Eq. (4) will be used to update the rest of the swarm (i.e., the whole swarm apart from producers). Note that, after the position is updated at each iteration, the continuous values of the position vector are preserved for future use in the continuous position update throughout the successive iterations. These values are also discretized using Eq. (6), so as to evaluate the fitness value of the produced binary solution based on the classification error rate obtained by the classifier involved using the features selected by iBSSA. Next, this process iterates until meeting a stopping criterion which is, in this study, the maximum number of iterations (which has proved sufficient to quantify the quality of the iBSSA algorithm). Finally, the overall pseudo-code of iBSSA can be found in Algorithm 4.

Algorithm 4 The proposed iBSSA based on 3RA and LSA

```

1: Initialize a swarm of  $N$  sparrows and its relevant parameters;
2: Apply binary conversion using Eq. (6), producing  $(\mathbf{x}_i)_{\text{bin}}$  while preserving the continuous  $\mathbf{x}_i$  for future fitness calculation;
3: Calculate the initial fitness of each sparrow using Eq. (7);
4:  $t \leftarrow 0$ ;
5: while  $t < T$  do
6:   Rank the fitness values  $\mathbf{f}(\mathbf{x})$  (see Eq. (2));  $\triangleright$  Rank the producers in ascending order based on their fitness values
7:   Find the current best and worst producers  $\mathbf{x}_p$  and  $\mathbf{x}_{\text{worst}}$ ;  $\triangleright$  First and last individuals in the ranking, respectively
8:    $R_2 \leftarrow$  a random value in  $[0, 1]$ ;
9:   for producer  $i = 1, 2, \dots, PD$  do
10:     Update the producer's location using Eq. (3);
11:     Apply 3RA to producer  $i$  using Algorithm 2;
12:   end for
13:   for sparrow  $i = PD + 1, PD + 2, \dots, N$  do
14:     Update the sparrow's location using Eq. (4);
15:     Apply 3RA to sparrow  $i$ , using Algorithm 2;
16:   end for
17:   for scrounger  $i = 1, 2, \dots, SD$  do
18:     Update the scrounger's location using Eq. (5);
19:     Apply 3RA to scrounger  $i$  using Algorithm 2;
20:   end for
21:   Apply binary conversion, using Eq. (6), to all sparrows in the swarm, producing  $(\mathbf{x}_i)_{\text{bin}}$  while preserving the continuous
 $\mathbf{x}_i$  for future position update;
22:   Calculate the fitness of all sparrows using Eq. (7)
23:   Re-rank the whole swarm in ascending order based on the fitness values  $\mathbf{f}(\mathbf{x})$  (see Eq. (2));
24:   Find the current global optimum position  $\mathbf{x}_{\text{best}}^{t+1}$ ;  $\triangleright$  First individual in the ranking
25:    $\mathbf{x}_{\text{best}} \leftarrow \mathbf{x}_{\text{best}}^{t+1}$ ;
26:    $f_g \leftarrow f(\mathbf{x}_{\text{best}})$ ;
27:   Apply LSA to  $\mathbf{x}_{\text{best}}$  using Algorithm 3;
28:    $t \leftarrow t + 1$ ;
29: end while
30: return  $\mathbf{x}_{\text{best}}, f_g$ ;

```

5 Experiments and discussion

This section highlights the computational experiments for the iBSSA algorithm against different competing meta-heuristic algorithms. Parameter settings, benchmark datasets, and performance measures adopted to validate the proposed approaches are articulated in this section as well.

5.1 Description of datasets

In order to extensively evaluate and validate the performance of the techniques proposed in this manuscript, 18 multifaceted, multi-scale benchmark datasets from the UCI data repository [38] in different fields (e.g., biology, politics, electromagnetic, game, physics, chemistry, and artificial) are used in all experiments. These datasets are very beneficial to better verify the methods proposed herein, based on the different number of instances and features involved. Table 2 shows the details of these datasets.

5.2 Environment and parameter settings

The proposed iBSSA algorithm was compared against a binary variant of the standard SSA algorithm (BSSA) as well as binary variants of other 11 state-of-the-art meta-heuristic algorithms, including BABC, BPSO, BBA, BGWO, BWOA, BGOA, BSFO, BHHO, BBSA, BASO, and BHGSO, which are all implemented in this study.

Concerning the classifiers adopted in this study, for k -NN, the Euclidean distance metric $k = 5$, while SVM builds on a polynomial kernel with degree $d = 2$. In RF, the number of estimators $n = 10$, and the maximum depth $d = 5$. Due to the stochastic nature of meta-heuristics, for each method, 30 independent runs were performed. Then, the average values of the performance metrics were recorded over the 30 experiments. In order for the comparison to be fair, the maximum number of iterations and the swarm size were respectively set to 100 and 10 for all algorithms. Besides, the number of features in the datasets adopted herein represents the problem size, while the continuous search domain is set to $[-1, 1]$ in order to allow individuals to relatively search more broadly, yet restrictively within the continuous search space. Lastly, the number of local iterations in the proposed LSA algorithm was set to 20.

In the proposed approach, the optimality of the results was validated by using the hold-out strategy, where the training and test sets are realized by randomly dividing each dataset into two parts, where the training phase is done on 80% of the dataset while the remaining 20% is for testing purposes [70]. The results in the following tables represent the average values over 30 runs in terms of the mean fitness value, the mean classification accuracy, and the mean number of selected features. The remaining parameters of each algorithm are set according to the standard versions and the information provided in their first publications. The common settings of all algorithms, along

Table 2 Description of the datasets used in this study

#	Dataset	No. of features	No. of instances	Domain
1	Breastcancer	9	699	Biology
2	BreastEW	30	569	Biology
3	CongressEW	16	435	Politics
4	Exactly	13	1000	Biology
5	Exactly2	13	1000	Biology
6	HeartEW	13	270	Biology
7	IonosphereEW	34	351	Electromagnetic
8	KrvskpEW	36	3196	Game
9	Lymphography	18	148	Biology
10	M-of-n	13	1000	Biology
11	PenglungEW	325	73	Biology
12	SonarEW	60	208	Biology
13	SpectEW	22	267	Biology
14	Tic-tac-toe	9	958	Game
15	Vote	16	300	Politics
16	WaveformEW	40	5000	Physics
17	WineEW	13	178	Chemistry
18	Zoo	16	101	Artificial

with parameter settings for each algorithm, are described in Table 3. To run all experiments in this study, Python was used on a computing environment with a Dual Intel® Xeon® Gold 5115 2.4 GHz CPU and 128 GB of RAM on the operating system Microsoft Windows Server 2019.

5.3 Performance metrics

In this study, to validate the performance of the proposed iBSSA against competitors, each optimization algorithm is evaluated independently 30 times for each dataset, in order to make the empirical results statistically more significant. To this end, some key performance measures in the FS problem were adopted as follows:

- **Mean accuracy** (μ_{Acc}): This metric estimates the rate at which the data is classified correctly. The mean classification accuracy (μ_{Acc}), which is obtained by executing the algorithm 30 independent times, is expressed as Eq. (8):

$$\mu_{Acc} = \frac{1}{30} \frac{1}{m} \sum_{k=1}^{30} \sum_{r=1}^m match(PL_r, AL_r), \quad (8)$$

where m represents the size of samples in the test dataset, PL_r and AL_r respectively denote the output label of the predicted class and the reference class label for sample r , while $match(PL_r, AL_r)$ denotes a discriminant comparison function. If $PL_r == AL_r$, then $match(PL_r, AL_r) = 1$; otherwise, $match(PL_r, AL_r) = 0$.

- **Mean fitness value** (μ_{Fit}): This metric measures the average fitness value obtained by executing the

algorithm 30 independent times, which defines the relationship between reducing the classification error rate and minimizing the selected features' number as per Eq. (7). The lower value implies the more optimal solution and is determined using Eq. (9):

$$\mu_{Fit} = \frac{1}{30} \sum_{k=1}^{30} f_*^k, \quad (9)$$

where f_*^k represents the optimal fitness value obtained from k -th run.

- **Mean size of selected features** (μ_{Feat}): This represents the average number of selected features (or the feature selection ratio) by executing the algorithm 30 independent times, and is estimated as:

$$\mu_{Feat} = \frac{1}{30} \sum_{k=1}^{30} \frac{d_*^k}{D}, \quad (10)$$

where d_*^k is the number of selected features for k -th run, and D is the dimensionality or the number of all features in the original dataset.

- **Standard deviation** (σ_Y): Corresponding to the aforementioned measures, the final results achieved over the 30 independent runs for each algorithm on every dataset are evaluated and analyzed in terms of stability as:

$$\sigma_Y = \sqrt{\frac{1}{29} \sum_{k=1}^{30} (Y_*^k - \mu_Y)^2}, \quad (11)$$

where Y denotes the metric to be measured, Y_*^k is the value of the metric Y in the k -th run, and μ_Y is the average of the metric over the 30 independent runs.

Table 3 Parameter setup for all algorithms

Algorithm	Parameter
All algorithms	Population size $N = 10$ Number of runs for each method = 30 Maximum number of iterations $T = 100$ Dimensionality $D =$ number of features in the datasets In Eq. (7), $\alpha = 0.99$ and $\beta = 0.01$
BSSA & iBSSA	Number of producers $PD = 0.2 * N$ Number of scroungers $SD = 0.1 * N$ Safety threshold $ST = 0.8$
iBSSA	Maximum number of iterations in LSA = 20
BABC	Number of employed bees = 16 Number of onlooker bees = 4 Number of scout bees = 3
BPSO	Acceleration coefficients ($c_1 = c_2 = 1.2$) Inertia weight ($\omega_{\min} = 0.4, \omega_{\max} = 0.9$)
BBA	Loudness $A = 0.8$ Pulse emission rate $r = 0.95$ Lower and upper pulse frequencies = $\{0, 10\}$
BGWO	a is linearly decreased from 2 to 0
BWOA	a is linearly decreased from 2 to 0 $p = 0.5$ $b = 1.0$
BGOA	$C_{\min} = 0.00004$ and $C_{\max} = 1$
BSFO	Ratio between sailfish and sardines $pp = 0.1$ $A = 1$ $\varepsilon = 0.0001$
BHHO	Energy of rabbit $E \in [-1, 1]$
BBSA	Flight frequency $ff = 10$ Probability of foraging for food $p = 0.8$ Acceleration coefficients ($c_1 = c_2 = 1.5$) Effect on birds' vigilance behaviors ($a_1 = a_2 = 1.0$) Followed coefficient $fl = 0.5$
BASO	Depth weight $\alpha = 50$ Multiplier weight $\beta = 0.2$
BHGSO	Number of clusters = 2 $\alpha = \beta = 0.1$ and $K = 1.0$ $l_1 = 5E - 03, l_2 = 1E + 02, \text{ and } l_3 = 1E - 02$

- Wilcoxon's rank-sum test:** For a better understanding of the significance of the underlying technique, its impact should be statistically proved. Thus, the quality of the final results obtained from the involved approaches is often verified using Wilcoxon's rank-sum non-parametric test. This test, among many other statistical tests, is widely used to statistically differentiate the significance and reliability of many approaches in competition [26]. Using this test, the present study signifies the proposed iBSSA approach against the competitor algorithms, where a null hypothesis states

that the iBSSA algorithm does not significantly differ – in terms of performance – from other methods in pairwise comparison; otherwise, the iBSSA performs significantly better than others. The Wilcoxon's rank-sum test calculates a so-called p -value, based on which the difference between paired groups is analyzed in terms of the results obtained from 30 independent runs of the proposed iBSSA and the other algorithms in the comparison.

Table 4 Assessment of the impact of the nine TFs on iBSSA based on k -NN in terms of the mean fitness value (μ_{Fit})

Benchmark	Metric	S _{v1}	S _{v1c}	S _{v2}	S _{v3}	S _{v4}	V _{v1}	V _{v2}	V _{v3}	V _{v4}
BreastCancer	μ_{Fit}	0.0206	0.0202	0.0203	0.0203	0.0202	0.0201	0.0201	0.0201	0.0201
	σ_{Fit}	0.0010	0.0004	0.0006	0.0006	0.0005	0.0000	0.0000	0.0000	0.0000
BreastEW	μ_{Fit}	0.0367	0.0365	0.0369	0.0369	0.0363	0.0366	0.0364	0.0364	0.0363
	σ_{Fit}	0.0004	0.0016	0.0005	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004
CongressEW	μ_{Fit}	0.0268	0.0262	0.0266	0.0265	0.0262	0.0261	0.0265	0.0261	0.0261
	σ_{Fit}	0.0019	0.0006	0.0007	0.0007	0.0007	0.0005	0.0016	0.0007	0.0005
Exactly	μ_{Fit}	0.0357	0.0289	0.0307	0.0290	0.0340	0.0074	0.0077	0.0083	0.0074
	σ_{Fit}	0.0332	0.0278	0.0276	0.0277	0.0332	0.0072	0.0073	0.0066	0.0072
Exactly2	μ_{Fit}	0.2308	0.2292	0.2302	0.2310	0.2294	0.2298	0.2290	0.2289	0.2290
	σ_{Fit}	0.0032	0.0023	0.0028	0.0028	0.0018	0.0023	0.0020	0.0017	0.0019
HeartEW	μ_{Fit}	0.0958	0.0959	0.0965	0.0958	0.0957	0.0954	0.0955	0.0955	0.0953
	σ_{Fit}	0.0009	0.0010	0.0034	0.0009	0.0011	0.0005	0.0007	0.0007	0.0006
IonosphereEW	μ_{Fit}	0.0665	0.0646	0.0734	0.0729	0.0512	0.0652	0.0675	0.0647	0.0640
	σ_{Fit}	0.0085	0.0093	0.0078	0.0103	0.0099	0.0102	0.0098	0.0080	0.0085
KrVsKpEW	μ_{Fit}	0.0226	0.0219	0.0226	0.0228	0.0230	0.0211	0.0199	0.0205	0.0208
	σ_{Fit}	0.0026	0.0023	0.0026	0.0031	0.0026	0.0030	0.0030	0.0030	0.0027
Lymphography	μ_{Fit}	0.1703	0.1681	0.1692	0.1694	0.1694	0.1602	0.1601	0.1622	0.1644
	σ_{Fit}	0.0058	0.0058	0.0083	0.0008	0.0008	0.0139	0.0141	0.0152	0.0108
M-of-n	μ_{Fit}	0.0098	0.0097	0.0100	0.0103	0.0127	0.0051	0.0054	0.0047	0.0047
	σ_{Fit}	0.0073	0.0069	0.0089	0.0084	0.0097	0.0021	0.0038	0.0002	0.0002
PenglungEW	μ_{Fit}	0.3394	0.3371	0.3508	0.3553	0.3097	0.3553	0.3421	0.3465	0.3486
	σ_{Fit}	0.0260	0.0290	0.0336	0.0308	0.0310	0.0352	0.0327	0.0312	0.0325
SonarEW	μ_{Fit}	0.0185	0.0208	0.0162	0.0193	0.0187	0.0140	0.0132	0.0155	0.0124
	σ_{Fit}	0.0109	0.0101	0.0115	0.0109	0.0108	0.0113	0.0113	0.0116	0.0112
SpectEW	μ_{Fit}	0.1129	0.1147	0.1176	0.1162	0.1148	0.1104	0.1121	0.1114	0.1131
	σ_{Fit}	0.0056	0.0089	0.0078	0.0078	0.0076	0.0061	0.0063	0.0051	0.0047
Tic-tac-toe	μ_{Fit}	0.1544	0.1544	0.1547	0.1547	0.1544	0.1547	0.1544	0.1544	0.1544
	σ_{Fit}	0.0000	0.0000	0.0018	0.0018	0.0000	0.0018	0.0000	0.0000	0.0000
Vote	μ_{Fit}	0.0024	0.0022	0.0024	0.0026	0.0023	0.0021	0.0021	0.0021	0.0021
	σ_{Fit}	0.0005	0.0004	0.0006	0.0005	0.0005	0.0003	0.0003	0.0003	0.0004
WaveformEW	μ_{Fit}	0.1607	0.1597	0.1608	0.1590	0.1598	0.1567	0.1564	0.1544	0.1557
	σ_{Fit}	0.0049	0.0042	0.0044	0.0044	0.0042	0.0036	0.0040	0.0057	0.0051
WineEW	μ_{Fit}	0.0033	0.0033	0.0035	0.0034	0.0034	0.0031	0.0031	0.0031	0.0031
	σ_{Fit}	0.0003	0.0004	0.0005	0.0005	0.0004	0.0002	0.0001	0.0002	0.0002
Zoo	μ_{Fit}	0.0035	0.0035	0.0036	0.0036	0.0035	0.0033	0.0034	0.0033	0.0033
	σ_{Fit}	0.0004	0.0003	0.0004	0.0003	0.0004	0.0003	0.0004	0.0003	0.0003
Ranking	W T L	0 1 1 7	0 1 1 7	0 0 1 8	0 0 1 8	2 2 1 4	1 6 1 1	2 4 1 2	2 7 9	2 9 7

The experimental results are closely analyzed in the subsequent subsections, in which **boldface** numbers indicate the best results.

5.4 Assessment of the impact of the nine TFs on the fitness values

SSA was originally proposed for continuous optimization problems; therefore, a proper transformation function

should be identified to turn the algorithm mechanism to efficiently search in discrete (or binary) space. To this end, iBSSA performance was evaluated based on nine different TFs, including S-shaped and V-shaped functions across the selected 18 UCI benchmark datasets to solve the wrapper FS problem, pursuing the best-performing approach. The different TFs were tested with iBSSA based on the k -NN classifier in terms of the mean fitness value (μ_{Fit}) and the results are reported in Table 4, along with the respective

Table 5 Assessment of the impact of the nine TFs on iBSSA based on SVM in terms of the mean fitness value (μ_{Fit})

Benchmark	Metric	S_{v1}	S_{v1c}	S_{v2}	S_{v3}	S_{v4}	V_{v1}	V_{v2}	V_{v3}	V_{v4}
BreastCancer	μ_{Fit}	0.0265	0.0264	0.0265	0.0264	0.0265	0.0263	0.0262	0.0262	0.0263
	σ_{Fit}	0.0004	0.0004	0.0004	0.0004	0.0005	0.0003	0.0000	0.0002	0.0002
BreastEW	μ_{Fit}	0.0546	0.0546	0.0552	0.0555	0.0530	0.0544	0.0545	0.0542	0.0544
	σ_{Fit}	0.0008	0.0007	0.0016	0.0022	0.0034	0.0007	0.0007	0.0006	0.0007
CongressEW	μ_{Fit}	0.0254	0.0255	0.0256	0.0255	0.0254	0.0249	0.0249	0.0250	0.0249
	σ_{Fit}	0.0005	0.0004	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004
Exactly	μ_{Fit}	0.2612	0.2617	0.2617	0.2637	0.2693	0.2609	0.2572	0.2593	0.2592
	σ_{Fit}	0.0134	0.0150	0.0153	0.0183	0.0233	0.0137	0.0003	0.0109	0.0102
Exactly2	μ_{Fit}	0.2483	0.2483	0.2483	0.2483	0.2483	0.2483	0.2483	0.2483	0.2483
	σ_{Fit}	0.0000	0.0000	0.0001	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
HeartEW	μ_{Fit}	0.0970	0.0976	0.0954	0.0971	0.0978	0.0951	0.0950	0.0951	0.0950
	σ_{Fit}	0.0055	0.0061	0.0005	0.0053	0.0060	0.0005	0.0004	0.0005	0.0005
IonosphereEW	μ_{Fit}	0.0381	0.0398	0.0380	0.0390	0.0392	0.0357	0.0352	0.0329	0.0354
	σ_{Fit}	0.0085	0.0068	0.0077	0.0078	0.0082	0.0061	0.0080	0.0074	0.0068
KrVsKpEW	μ_{Fit}	0.0233	0.0224	0.0251	0.0259	0.0211	0.0227	0.0207	0.0212	0.0209
	σ_{Fit}	0.0033	0.0033	0.0035	0.0034	0.0021	0.0037	0.0028	0.0027	0.0028
Lymphography	μ_{Fit}	0.1563	0.1584	0.1562	0.1586	0.1583	0.1549	0.1527	0.1476	0.1505
	σ_{Fit}	0.0158	0.0175	0.0162	0.0155	0.0155	0.0160	0.0163	0.0155	0.0163
M-of-n	μ_{Fit}	0.0054	0.0054	0.0052	0.0053	0.0054	0.0047	0.0048	0.0047	0.0048
	σ_{Fit}	0.0005	0.0005	0.0004	0.0005	0.0005	0.0003	0.0003	0.0003	0.0004
PenglungEW	μ_{Fit}	0.1369	0.1369	0.1658	0.1659	0.1008	0.1615	0.1746	0.1855	0.1768
	σ_{Fit}	0.0118	0.0207	0.0328	0.0328	0.0329	0.0321	0.0321	0.0278	0.0316
SonarEW	μ_{Fit}	0.0620	0.0573	0.0613	0.0612	0.0579	0.0563	0.0556	0.0563	0.0573
	σ_{Fit}	0.0132	0.0134	0.0144	0.0156	0.0164	0.0167	0.0142	0.0156	0.0118
SpectEW	μ_{Fit}	0.1680	0.1703	0.1705	0.1720	0.1731	0.1669	0.1658	0.1650	0.1671
	σ_{Fit}	0.0066	0.0062	0.0092	0.0103	0.0100	0.0088	0.0104	0.0128	0.0072
Tic-tac-toe	μ_{Fit}	0.1017	0.1017	0.1018	0.1018	0.1017	0.1017	0.1017	0.1017	0.1017
	σ_{Fit}	0.0000	0.0000	0.0004	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000
Vote	μ_{Fit}	0.0024	0.0024	0.0026	0.0027	0.0023	0.0022	0.0023	0.0021	0.0021
	σ_{Fit}	0.0006	0.0006	0.0007	0.0005	0.0006	0.0005	0.0005	0.0004	0.0004
WaveformEW	μ_{Fit}	0.1273	0.1282	0.1273	0.1267	0.1276	0.1253	0.1256	0.1254	0.1255
	σ_{Fit}	0.0032	0.0028	0.0028	0.0028	0.0031	0.0029	0.0027	0.0023	0.0026
WineEW	μ_{Fit}	0.0020	0.0022	0.0020	0.0022	0.0017	0.0017	0.0017	0.0017	0.0018
	σ_{Fit}	0.0005	0.0005	0.0005	0.0005	0.0003	0.0003	0.0004	0.0003	0.0004
Zoo	μ_{Fit}	0.0036	0.0036	0.0036	0.0036	0.0035	0.0033	0.0032	0.0034	0.0033
	σ_{Fit}	0.0004	0.0005	0.0004	0.0004	0.0004	0.0003	0.0002	0.0004	0.0003
Ranking	W T L	0 2 16	0 2 16	0 1 17	0 1 17	2 3 13	1 5 12	4 5 9	3 6 9	0 5 13

standard deviations. The same observation is highlighted for the SVM and the RF classifier as per Tables 5 and 6, respectively. In the FS optimization problem, the objective function (see Eq. (7)) is formulated based on both the mean classification error rate and the mean feature selection ratio; therefore, the mean fitness value was mainly considered in solving such a problem. The approaches were built based upon nine TFs called S_{v1} , S_{v1c} , S_{v2} , S_{v3} , and S_{v4} for the S-shaped TFs, and V_{v1} , V_{v2} , V_{v3} , and V_{v4} for the V-

shaped TFs. Consequently, throughout the following discussion, the proposed methods are termed as “iBSSA-TF”, where TF is any of the nine TFs. W|T|L at the bottom of the tables represent how many times each competitive method wins/ties/loses compared to others. By analyzing and comparing the above experimental results, the optimal iBSSA variant corresponding to each classifier is to be obtained based on the best-performing TF for that classifier.

Table 6 Assessment of the impact of the nine TFs on iBSSA based on the RF classifier in terms of the mean fitness value (μ_{Fit})

Benchmark	Metric	S _{v1}	S _{v1c}	S _{v2}	S _{v3}	S _{v4}	V _{v1}	V _{v2}	V _{v3}	V _{v4}
BreastCancer	μ_{Fit}	0.0195	0.0194	0.0195	0.0195	0.0197	0.0193	0.0193	0.0196	0.0195
	σ_{Fit}	0.0005	0.0004	0.0005	0.0005	0.0005	0.0004	0.0004	0.0005	0.0005
BreastEW	μ_{Fit}	0.0106	0.0098	0.0114	0.0106	0.0122	0.0097	0.0097	0.0125	0.0100
	σ_{Fit}	0.0038	0.0043	0.0038	0.0048	0.0044	0.0046	0.0048	0.0027	0.0043
CongressEW	μ_{Fit}	0.0238	0.0237	0.0232	0.0236	0.0241	0.0235	0.0220	0.0232	0.0226
	σ_{Fit}	0.0036	0.0036	0.0041	0.0039	0.0033	0.0035	0.0045	0.0037	0.0041
Exactly	μ_{Fit}	0.2718	0.2676	0.2765	0.2716	0.2634	0.2240	0.2384	0.2310	0.2039
	σ_{Fit}	0.0365	0.0412	0.0307	0.0364	0.0450	0.0611	0.0574	0.0595	0.0579
Exactly2	μ_{Fit}	0.2378	0.2372	0.2380	0.2383	0.2375	0.2372	0.2384	0.2377	0.2387
	σ_{Fit}	0.0025	0.0017	0.0030	0.0033	0.0024	0.0017	0.0036	0.0028	0.0039
HeartEW	μ_{Fit}	0.0772	0.0771	0.0754	0.0776	0.0739	0.0732	0.0746	0.0744	0.0771
	σ_{Fit}	0.0123	0.0123	0.0105	0.0117	0.0123	0.0124	0.0147	0.0136	0.0129
IonosphereEW	μ_{Fit}	0.0346	0.0357	0.0360	0.0370	0.0374	0.0350	0.0345	0.0339	0.0336
	σ_{Fit}	0.0052	0.0057	0.0056	0.0062	0.0063	0.0056	0.0064	0.0056	0.0040
KrVsKpEW	μ_{Fit}	0.0582	0.0569	0.0588	0.0587	0.0578	0.0558	0.0556	0.0543	0.0550
	σ_{Fit}	0.0053	0.0051	0.0054	0.0051	0.0067	0.0067	0.0048	0.0050	0.0051
Lymphography	μ_{Fit}	0.1291	0.1339	0.1280	0.1358	0.1342	0.1246	0.1239	0.1267	0.1278
	σ_{Fit}	0.0180	0.0239	0.0203	0.0177	0.0154	0.0176	0.0213	0.0144	0.0233
M-of-n	μ_{Fit}	0.0280	0.0305	0.0254	0.0251	0.0256	0.0072	0.0053	0.0057	0.0064
	σ_{Fit}	0.0207	0.0215	0.0195	0.0211	0.0217	0.0075	0.0027	0.0032	0.0047
PenglungEW	μ_{Fit}	0.2529	0.2548	0.2488	0.2553	0.2430	0.2533	0.2532	0.2578	0.2556
	σ_{Fit}	0.0280	0.0264	0.0454	0.0357	0.0356	0.0406	0.0405	0.0299	0.0314
SonarEW	μ_{Fit}	0.0894	0.0885	0.0918	0.0894	0.0815	0.0839	0.0831	0.0842	0.0853
	σ_{Fit}	0.0214	0.0155	0.0194	0.0143	0.0180	0.0166	0.0151	0.0167	0.0156
SpectEW	μ_{Fit}	0.1248	0.1285	0.1261	0.1236	0.1266	0.1204	0.1203	0.1197	0.1196
	σ_{Fit}	0.0100	0.0094	0.0117	0.0102	0.0082	0.0103	0.0092	0.0089	0.0089
Tic-tac-toe	μ_{Fit}	0.1367	0.1367	0.1370	0.1367	0.1367	0.1367	0.1367	0.1367	0.1367
	σ_{Fit}	0.0000	0.0000	0.0015	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Vote	μ_{Fit}	0.0019	0.0019	0.0021	0.0022	0.0016	0.0017	0.0017	0.0018	0.0017
	σ_{Fit}	0.0005	0.0004	0.0005	0.0004	0.0005	0.0004	0.0005	0.0005	0.0004
WaveformEW	μ_{Fit}	0.1872	0.1856	0.1879	0.1866	0.1868	0.1836	0.1829	0.1838	0.1845
	σ_{Fit}	0.0042	0.0072	0.0060	0.0056	0.0047	0.0050	0.0053	0.0056	0.0053
WineEW	μ_{Fit}	0.0024	0.0026	0.0026	0.0027	0.0024	0.0025	0.0024	0.0024	0.0025
	σ_{Fit}	0.0003	0.0004	0.0004	0.0004	0.0003	0.0003	0.0003	0.0002	0.0003
Zoo	μ_{Fit}	0.0030	0.0031	0.0031	0.0031	0.0030	0.0028	0.0027	0.0028	0.0028
	σ_{Fit}	0.0004	0.0004	0.0003	0.0005	0.0004	0.0003	0.0003	0.0004	0.0004
Ranking	W T L	0 2 16	0 2 16	0 0 16	0 1 17	3 2 13	1 5 12	5 4 9	1 2 15	3 1 14

Table 7 The three classifiers along with corresponding best-performing binary variants in the proposed iBSSA algorithm

Classifier	Best-performing variant
k-NN	iBSSA-V _{v4}
SVM	iBSSA-V _{v2}
RF	iBSSA-V _{v2}

5.4.1 Assessment using the k-NN classifier

Table 4 shows the mean fitness values (μ_{Fit}) and standard deviations (σ_{Fit}) of iBSSA based on the nine TFs, using the k-NN classifier. It is observed that iBSSA-V_{v4} shows a significant performance in 11 out of the 18 datasets, followed by iBSSA-V_{v3}, iBSSA-V_{v1}, iBSSA-V_{v2}, and iBSSA-S_{v4} with 9, 7, 6, and 4 datasets, respectively.

Table 8 Performance comparisons of iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF in terms of the mean fitness value (μ_{Fit}), mean classification accuracy (μ_{Acc}), and mean number of selected features (μ_{Feat})

Benchmark	Fitness (μ_{Fit})			Accuracy (μ_{Acc})			Feature (μ_{Feat})		
	iBSSA- <i>k</i> -NN	iBSSA-SVM	iBSSA-RF	iBSSA- <i>k</i> -NN	iBSSA-SVM	iBSSA-RF	iBSSA- <i>k</i> -NN	iBSSA-SVM	iBSSA-RF
BreastCancer	0.0201	0.0262	0.0193	0.9857	0.9786	0.9857	006.00	005.00	005.17
BreastEW	0.0363	0.0545	0.0097	0.9649	0.9474	0.9944	004.60	007.13	012.67
CongressEW	0.0261	0.0249	0.0220	0.9770	0.9770	0.9808	005.40	003.50	004.93
Exactly	0.0074	0.2572	0.2384	0.9973	0.7450	0.7633	006.13	006.20	005.37
Exactly2	0.2290	0.2483	0.2384	0.7737	0.7500	0.7632	006.47	001.00	005.07
HeartEW	0.0953	0.0950	0.0746	0.9074	0.9074	0.9302	004.77	004.33	007.23
IonosphereEW	0.0640	0.0352	0.0345	0.9385	0.9681	0.9690	010.63	012.23	013.10
KrVsKpEW	0.0208	0.0207	0.0556	0.9852	0.9854	0.9487	021.97	022.60	017.17
Lymphography	0.1644	0.1527	0.1239	0.8378	0.8500	0.8800	006.90	007.53	009.20
M-of-n	0.0047	0.0048	0.0053	1.0000	1.0000	0.9993	006.07	006.27	006.07
PenglungEW	0.3486	0.1746	0.2532	0.6511	0.8267	0.7489	103.57	096.90	150.33
SonarEW	0.0124	0.0556	0.0831	0.9913	0.9476	0.9206	022.43	022.27	027.33
SpectEW	0.1131	0.1658	0.1203	0.8889	0.8364	0.8815	006.77	008.43	006.57
Tic-tac-toe	0.1544	0.1017	0.1367	0.8542	0.9062	0.8698	009.00	008.00	007.00
Vote	0.0021	0.0023	0.0017	1.0000	1.0000	1.0000	003.37	003.60	002.73
WaveformEW	0.1557	0.1256	0.1829	0.8484	0.8791	0.8201	022.43	023.47	019.33
WineEW	0.0031	0.0017	0.0024	1.0000	1.0000	1.0000	004.07	002.27	003.13
Zoo	0.0033	0.0032	0.0027	1.0000	1.0000	1.0000	005.33	005.17	004.33
Ranking (W T L))	5 0 13	5 0 13	8 0 10	4 5 9	4 4 10	5 4 9	4 11 13	7 0 11	6 11 11

Therefore, in terms of the mean fitness values, iBSSA-S_{v4} ranks first among all methods.

5.4.2 Assessment using the SVM classifier

Table 5 represents the mean fitness values (μ_{Fit}) based on the proposed iBSSA with the SVM classifier across the nine TFs. In terms of the mean fitness value in Table 5, although iBSSA-V_{v2} and iBSSA-V_{v3} seemingly share the first place on 50% of the datasets based on the number of wins and ties, iBSSA-V_{v2} experienced the best overall performance in comparison with iBSSA-V_{v3} based on the number of wins. In addition, iBSSA-V_{v1}, iBSSA-S_{v4}, and iBSSA-V_{v4} ranked first on 6, 5, and 5 datasets, respectively, but they were all less than iBSSA-V_{v2}. Thus, it can be concluded that the iBSSA-V_{v2} method is generally performing the best with SVM based on the mean fitness values.

5.4.3 Assessment using the RF classifier

When testing the iBSSA algorithm with nine TFs for FS in classification using RF, iBSSA-V_{v2} had strong

competitiveness against peers. In Table 6, among the 18 datasets, 9 datasets of iBSSA-V_{v2} are marked as winner and tied, which ranks the method as the first over the other nine methods in terms of the mean fitness value. Inspecting the above comparison, it is well shown that iBSSA-V_{v2} is the best-performing binary variant with the RF classifier in the proposed iBSSA algorithm.

All in all, Table 7 shows the best combinations of iBSSA variant and the three classifiers with the nine TFs, based on which the upcoming experiments in this section were conducted. Now, the finally selected, most effective models can be termed as iBSSA-V_{v4}-*k*-NN, iBSSA-V_{v2}-SVM, and iBSSA-V_{v2}-RF. In this regard, it should be noted that, for simplicity, these models are abbreviated based on the three classifiers respectively as iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF, and used consistently hereafter without including the respective TF's name.

5.4.4 Overall assessment

In Table 8, iBSSA-*k*-NN is compared with iBSSA-SVM and iBSSA-RF. In terms of the mean fitness value, it can be seen that the iBSSA-RF ranks first among 8 of the 18

datasets. For the other methods, iBSSA- k -NN and iBSSA-SVM both have 5 datasets ranking first. In the final statistical ranking, iBSSA-RF ranked first, followed by iBSSA-SVM and iBSSA- k -NN, for the value of mean fitness. In comparison of mean classification accuracy, iBSSA-RF won over 5 while tied over 4 of the 18 datasets in the combativeness of classification. Other methods only have 4 winning datasets, which are all less than iBSSA-RF. Especially on the datasets M-of-n, Vote, WineEW, and Zoo, the mean accuracy of iBSSA- k -NN and iBSSA-SVM has reached 100%, while iBSSA-RF reached that mean accuracy on the late 3 datasets only. iBSSA-RF also ranks first in the final statistics, followed by iBSSA- k -NN and iBSSA-SVM. The experimental results on the mean number of selected features for all methods are also shown in Table 8. As can be seen, iBSSA-SVM ranks first on 7 out of the 18 datasets. Especially from the numerical perspective, iBSSA-SVM can relatively find a smaller feature subset, for example, on the high-dimensional dataset PenglungEW, where the selected feature's mean number is nearly 30% of the original number of features. Similarly, in the final ranking, iBSSA-SVM ranked first, followed by iBSSA-RF and iBSSA- k -NN in terms of the mean number of features selected. From the experimental results analyzed above, it is asserted that both SVM and RF are

strongly competitive in comparison with k -NN in terms of mean fitness, mean classification accuracy, and mean number of selected features by the proposed iBSSA algorithm. The harmony between the mean fitness values and mean classification implies a slight advantage of iBSSA-RF among the three methods.

These findings can be justified by the nature of the classifiers themselves. k -NN can be described as a lazy classifier in which the learning step is not even required as the classification task is done by simply calculating the distance between the test and training sets. On the other hand, SVM is more complex by using the learning phase to update the bias and weights used in the testing step, in order to create the model. RF is also considered as a complex learning method which is based on training multiple deep decision trees on different parts of the same training set and averaging the results in order to reduce the variance. This comes at the expense of some loss of interpretability and a slight increase in the bias. Although not quite similar, forests act like a K -fold cross-validation. The aforementioned characteristics provide a greater capacity to SVM and RF than k -NN in determining the pertinent features with high discrimination.

Thus, in order to fully verify the effectiveness of the algorithm proposed in this study in feature selection and

Table 9 Performance comparisons of classification using the original k -NN classifier on all the features (before FS) and classification using the proposed iBSSA- k -NN (after FS) in terms of the mean classification accuracy (μ_{Acc}) and the mean number of selected features (μ_{Feat})

Benchmark	Accuracy			Feature		
	k -NN	iBSSA- k -NN (μ_{Acc})	Improvement (%)	k -NN	iBSSA- k -NN (μ_{Feat})	Reduction (%)
BreastCancer	0.6214	0.9857	58.63%	009.00	006.00	33.33%
BreastEW	0.9211	0.9649	04.76%	030.00	004.60	84.67%
CongressEW	0.9310	0.9770	04.94%	016.00	005.40	66.25%
Exactly	0.7350	0.9973	35.69%	013.00	006.13	52.85%
Exactly2	0.7400	0.7737	04.55%	013.00	006.47	50.23%
HeartEW	0.6296	0.9074	44.12%	013.00	004.77	63.31%
IonosphereEW	0.8451	0.9385	11.05%	034.00	010.63	68.74%
KrVsKpEW	0.9656	0.9852	02.03%	036.00	021.97	38.97%
Lymphography	0.7000	0.8378	19.69%	018.00	006.90	61.67%
M-of-n	0.8800	1.0000	13.64%	013.00	006.07	53.31%
PenglungEW	0.5333	0.6511	22.09%	325.00	103.57	68.13%
SonarEW	0.8571	0.9913	15.66%	060.00	022.43	62.62%
SpectEW	0.7778	0.8889	14.28%	022.00	006.77	69.23%
Tic-tac-toe	0.8542	0.8542	00.00%	009.00	009.00	00.00%
Vote	0.9333	1.0000	07.15%	016.00	003.37	78.94%
WaveformEW	0.8040	0.8484	05.52%	040.00	022.43	43.93%
WineEW	0.5833	1.0000	71.44%	013.00	004.07	68.69%
Zoo	0.9048	1.0000	10.52%	016.00	005.33	66.69%
Ranking (W T L)	0 1 17	17 1 0	*	0 1 17	17 1 0	*

Table 10 Performance comparisons of classification using the original SVM classifier on all the features (before FS) and classification using the proposed iBSSA–SVM (after FS) in terms of the mean classification accuracy (μ_{Acc}) and the mean number of selected features (μ_{Feat})

Benchmark	Accuracy			Feature		
	SVM	iBSSA–SVM (μ_{Acc})	Improvement (%)	SVM	iBSSA–SVM (μ_{Feat})	Reduction (%)
BreastCancer	0.6786	0.9786	44.21%	009.00	005.00	44.44%
BreastEW	0.8947	0.9474	05.89%	030.00	007.13	76.23%
CongressEW	0.9655	0.9770	01.19%	016.00	003.50	78.13%
Exactly	0.6600	0.7450	12.88%	013.00	006.20	52.31%
Exactly2	0.7500	0.7500	00.00%	013.00	001.00	92.31%
HeartEW	0.7963	0.9074	13.95%	013.00	004.33	66.69%
IonosphereEW	0.8873	0.9681	09.11%	034.00	012.23	64.03%
KrVsKpEW	0.9766	0.9854	00.90%	036.00	022.60	37.22%
Lymphography	0.7667	0.8500	10.86%	018.00	007.53	58.17%
M-of-n	1.0000	1.0000	00.00%	013.00	006.27	51.77%
PenglungEW	0.7333	0.8267	12.74%	325.00	096.90	70.18%
SonarEW	0.8333	0.9476	13.72%	060.00	022.27	62.88%
SpectEW	0.7407	0.8364	12.92%	022.00	008.43	61.68%
Tic-tac-toe	0.9062	0.9062	00.00%	009.00	008.00	11.11%
Vote	0.9333	1.0000	07.15%	016.00	003.60	77.50%
WaveformEW	0.8650	0.8791	01.63%	040.00	023.47	41.33%
WineEW	0.6667	1.0000	49.99%	013.00	002.27	82.54%
Zoo	0.8571	1.0000	16.67%	016.00	005.17	67.69%
Ranking (W T L)	0 3 15	15 3 0	*	0 0 18	18 0 0	*

due to their differential yet helpful nature, the three classifiers, k -NN, SVM, and RF, were considered inclusively independently in all other experiments in this section.

5.5 Performance comparison of classification using the original classifier with all features (before FS), and the iBSSA–classifier method (after FS)

In this experiment, to determine the breadth of influence on it, the number of selected features on average based on the proposed iBSSA algorithm was compared with the original number of features in each dataset. In addition, the magnitude of increase in the level of classification accuracy over the three classifiers was quantified.

5.5.1 Comparisons of k -NN and iBSSA– k -NN

On one side, Table 9 shows the classification accuracy based on the original k -NN classifier (before FS) along with the original number of features in each dataset, for each of the 18 datasets. On the other hand, the table shows the mean classification accuracy (μ_{Acc}) and the mean number of selected features (μ_{Feat}) based on the proposed iBSSA– k -NN method (after FS). It is remarkable that μ_{Acc}

based on the iBSSA– k -NN method increased by more than 10% in 10 out of the 18 datasets with a significant increase of up to 25% on 4 of them, while the mean accuracy itself reached up to more than 90% in 12 out of the 18 datasets, achieving 100% mean accuracy for 4 datasets. It is also most notable that μ_{Feat} based on the iBSSA– k -NN method was reduced by more than 50% on 15 out of the 18 datasets. Unfortunately, iBSSA– k -NN was unable to enhance both μ_{Acc} and μ_{Feat} on the dataset Tic-tac-toe. It may be the dataset's particular nature and data characteristics which have hindered iBSSA from improving the feature subset selection. Irrespective, in general, the iBSSA– k -NN method significantly excelled the original k -NN over 17 datasets in terms of the two metrics. Thus, it is very clear how iBSSA– k -NN has provided a really promising solution to the FS problem compared to only k -NN, across the selected datasets.

5.5.2 Comparisons of SVM and iBSSA–SVM

For the SVM classifier, Table 10 shows that iBSSA–SVM outperformed the original SVM in terms of μ_{Acc} and μ_{Feat} because iBSSA generally selected fewer features over the 18 datasets with a reduction in feature size by more than 50% in 14 out of the 18 datasets, improving at the same

Table 11 Performance comparisons of classification using the original RF classifier on all the features (before FS) and classification using the proposed iBSSA–RF (after FS) in terms of the mean classification accuracy (μ_{Acc}) and the mean number of selected features (μ_{Feat})

Benchmark	Accuracy			Feature		
	RF	iBSSA–RF (μ_{Acc})	Improvement (%)	RF	iBSSA–RF (μ_{Feat})	Reduction (%)
BreastCancer	0.9643	0.9857	02.22%	009.00	005.17	42.56%
BreastEW	0.9298	0.9944	06.95%	030.00	012.67	57.77%
CongressEW	0.9195	0.9808	06.67%	016.00	004.93	69.19%
Exactly	0.6650	0.7633	14.78%	013.00	005.37	58.69%
Exactly2	0.7500	0.7632	01.76%	013.00	005.07	61.00%
HeartEW	0.7963	0.9302	16.82%	013.00	007.23	44.38%
IonosphereEW	0.9014	0.9690	07.50%	034.00	013.10	61.47%
KrVsKpEW	0.7953	0.9487	19.29%	036.00	017.17	52.31%
Lymphography	0.7333	0.8800	20.01%	018.00	009.20	48.89%
M-of-n	0.8000	0.9993	24.91%	013.00	006.07	53.31%
PenglungEW	0.3333	0.7489	124.69%	325.00	150.33	53.74%
SonarEW	0.7857	0.9206	17.17%	060.00	027.33	54.45%
SpectEW	0.7407	0.8815	19.01%	022.00	006.57	70.14%
Tic-tac-toe	0.7500	0.8698	15.97%	009.00	007.00	22.22%
Vote	0.9000	1.0000	11.11%	016.00	002.73	82.94%
WaveformEW	0.7690	0.8201	06.64%	040.00	019.33	51.68%
WineEW	1.0000	1.0000	00.00%	013.00	003.13	75.92%
Zoo	0.9524	1.0000	05.00%	016.00	004.33	72.94%
Ranking (W T L)	011117	17 1 0	*	010118	18 0 0	*

time the μ_{Acc} in 15 out of 18 datasets with 100% accuracy for 3 datasets while preserving the μ_{Acc} in the remaining 3 datasets. It is also observed that while the μ_{Acc} in the datasets Exactly2, M-of-n, and Tic-tac-toe did not increase, the respective number of selected features was reduced on average, over the 3 datasets. From the analysis, the iBSSA–SVM method proposed herein was superior to the original SVM alone, in most datasets.

5.5.3 Comparisons of RF and iBSSA–RF

As shown in Table 11, iBSSA–RF outperformed the original RF classifier in terms of the accuracy and the number of selected features, respectively, in 17 and 18 out of all 18 datasets. Among the 18 datasets, the μ_{Acc} in 10 datasets increased by up to 10%, including 3 datasets attaining 100% classification accuracy. On the other hand, the μ_{Feat} in 15 datasets was reduced by 50%, thereby showing highly significant improvements in the two FS main metrics (μ_{Acc} and μ_{Feat}). This shows that iBSSA–RF has much better performance.

5.6 Performance comparison of BSSA and iBSSA

In this experiment, to quantify the extent of improvement in it, iBSSA was compared to the original BSSA based on the three classifiers, k -NN, SVM, and RF, in terms of three metrics: μ_{Fit} , μ_{Acc} , and μ_{Feat} .

5.6.1 Comparisons based on the k -NN classifier

As shown in Table 12, which presents the mean metrics based on iBSSA and BSSA both with k -NN, iBSSA outperforms the original BSSA algorithm over the 18 datasets in terms of the mean fitness values. Thus, iBSSA achieved much higher performance than the original BSSA algorithm based on the fitness. Moreover, iBSSA outperformed BSSA in terms of mean classification accuracy on 12 of the 18 datasets, accounting for 67% of all datasets, with 100% classification accuracies on 4 datasets (M-of-n, Vote, WineEW, and Zoo). Furthermore, in terms of the mean number of selected features, iBSSA outperformed BSSA because a smaller number of features was selected by iBSSA over 15 (83%) of all datasets. In addition, given the standard deviations, iBSSA was more stable on the vast majority of datasets, for the three metrics. From these results, it is clearly shown that iBSSA with k -NN has

Table 12 Performance comparisons of BSSA and iBSSA in the FS problem based on k -NN in terms of the mean fitness value (μ_{Fit}), the mean classification accuracy (μ_{Acc}), and the mean number of selected features (μ_{Feat})

Benchmark	Metric	Fitness		Accuracy		Feature	
		BSSA	iBSSA	BSSA	iBSSA	BSSA	iBSSA
BreastCancer	μ	0.0208	0.0201	0.9857	0.9857	006.70	006.00
	σ	0.0010	0.0000	0.0000	0.0000	001.01	000.00
BreastEW	μ	0.0372	0.0363	0.9649	0.9649	007.53	004.60
	σ	0.0007	0.0004	0.0000	0.0000	002.09	001.23
CongressEW	μ	0.0281	0.0261	0.9747	0.9770	004.90	005.40
	σ	0.0036	0.0005	0.0046	0.0000	001.85	000.88
Exactly	μ	0.0177	0.0074	0.9872	0.9973	006.47	006.13
	σ	0.0213	0.0072	0.0211	0.0070	000.56	000.34
Exactly2	μ	0.2331	0.2290	0.7697	0.7737	006.57	006.47
	σ	0.0044	0.0019	0.0046	0.0022	001.23	001.06
HeartEW	μ	0.0960	0.0953	0.9074	0.9074	005.67	004.77
	σ	0.0009	0.0006	0.0000	0.0000	001.11	000.84
IonosphereEW	μ	0.0792	0.0640	0.9235	0.9385	011.80	010.63
	σ	0.0112	0.0085	0.0113	0.0085	003.05	002.86
KrVsKpEW	μ	0.0249	0.0208	0.9808	0.9852	021.23	021.97
	σ	0.0045	0.0027	0.0045	0.0027	002.25	002.58
Lymphography	μ	0.1662	0.1644	0.8367	0.8378	008.03	006.90
	σ	0.0154	0.0108	0.0158	0.0113	001.64	001.27
M-of-n	μ	0.0060	0.0047	0.9988	1.0000	006.30	006.07
	σ	0.0041	0.0002	0.0040	0.0000	000.46	000.25
PenglungEW	μ	0.3760	0.3486	0.6244	0.6511	136.53	103.57
	σ	0.0314	0.0325	0.0321	0.0330	018.08	012.64
SonarEW	μ	0.0236	0.0124	0.9802	0.9913	023.63	022.43
	σ	0.0088	0.0112	0.0089	0.0115	004.16	003.77
SpectEW	μ	0.1172	0.1131	0.8852	0.8889	007.77	006.77
	σ	0.0108	0.0047	0.0111	0.0048	002.25	001.84
Tic-tac-toe	μ	0.1554	0.1544	0.8526	0.8542	008.50	009.00
	σ	0.0030	0.0000	0.0047	0.0000	001.50	000.00
Vote	μ	0.0024	0.0021	1.0000	1.0000	003.87	003.37
	σ	0.0006	0.0004	0.0000	0.0000	000.99	000.60
WaveformEW	μ	0.1629	0.1557	0.8414	0.8484	023.60	022.43
	σ	0.0051	0.0051	0.0051	0.0053	002.63	002.40
WineEW	μ	0.0032	0.0031	1.0000	1.0000	004.17	004.07
	σ	0.0003	0.0002	0.0000	0.0000	000.37	000.25
Zoo	μ	0.0036	0.0033	1.0000	1.0000	005.83	005.33
	σ	0.0005	0.0003	0.0000	0.0000	000.78	000.47
Ranking	W T L	0 0 18	18 0 0	0 6 12	12 6 0	3 0 15	15 0 3

significantly improved both the FS and classification tasks compared to the original BSSA algorithm.

5.6.2 Comparisons based on the SVM classifier

Table 13 shows the experimental results by using both BSSA and iBSSA with SVM for FS in the classification task. iBSSA had 16 (89%), 12 (67%), and 16 (89%) out of

18 datasets ranked first on the mean fitness, mean classification accuracy, and mean number of selected features, which confirms a highly considerable overall performance of the proposed iBSSA algorithm over BSSA. It is also worth mentioning that the mean classification accuracy reached 100% on 4 datasets (M-of-n, Vote, WineEW, and Zoo) with less number of features selected by iBSSA. Generally, by comparing the two methods in terms of

Table 13 Performance comparisons of BSSA and iBSSA in the FS problem based on SVM in terms of the mean fitness value (μ_{Fit}), the mean classification accuracy (μ_{Acc}), and the mean number of selected features (μ_{Feat})

Benchmark	Metric	Fitness		Accuracy		Feature	
		BSSA	iBSSA	BSSA	iBSSA	BSSA	iBSSA
BreastCancer	μ	0.0272	0.0262	0.9776	0.9786	005.07	005.00
	σ	0.0020	0.0000	0.0024	0.0000	000.57	000.00
BreastEW	μ	0.0574	0.0545	0.9447	0.9474	007.97	007.13
	σ	0.0041	0.0007	0.0040	0.0000	001.82	002.05
CongressEW	μ	0.0256	0.0249	0.9770	0.9770	004.57	003.50
	σ	0.0006	0.0004	0.0000	0.0000	000.96	000.72
Exactly	μ	0.2940	0.2572	0.7085	0.7450	007.07	006.20
	σ	0.0303	0.0003	0.0304	0.0000	001.59	000.40
Exactly2	μ	0.2483	0.2483	0.7500	0.7500	001.03	001.00
	σ	0.0001	0.0000	0.0000	0.0000	000.18	000.00
HeartEW	μ	0.0970	0.0950	0.9056	0.9074	004.50	004.33
	σ	0.0055	0.0004	0.0056	0.0000	000.76	000.47
IonosphereEW	μ	0.0438	0.0352	0.9596	0.9681	012.87	012.23
	σ	0.0078	0.0080	0.0079	0.0081	002.29	002.54
KrVsKpEW	μ	0.0270	0.0207	0.9791	0.9854	022.43	022.60
	σ	0.0042	0.0028	0.0045	0.0027	003.07	002.59
Lymphography	μ	0.1607	0.1527	0.8422	0.8500	008.13	007.53
	σ	0.0190	0.0163	0.0191	0.0167	001.89	001.28
M-of-n	μ	0.0050	0.0048	1.0000	1.0000	006.50	006.27
	σ	0.0005	0.0003	0.0000	0.0000	000.62	000.44
PenglungEW	μ	0.1953	0.1746	0.8067	0.8267	126.87	096.90
	σ	0.0197	0.0321	0.0200	0.0327	007.51	010.21
SonarEW	μ	0.0718	0.0556	0.9317	0.9476	025.10	022.27
	σ	0.0131	0.0142	0.0134	0.0143	003.56	003.02
SpectEW	μ	0.1985	0.1658	0.8037	0.8364	009.20	008.43
	σ	0.0247	0.0104	0.0251	0.0108	002.06	001.78
Tic-tac-toe	μ	0.1017	0.1017	0.9062	0.9062	008.00	008.00
	σ	0.0000	0.0000	0.0000	0.0000	000.00	000.00
Vote	μ	0.0025	0.0023	1.0000	1.0000	004.00	003.60
	σ	0.0007	0.0005	0.0000	0.0000	001.13	000.80
WaveformEW	μ	0.1295	0.1256	0.8753	0.8791	024.17	023.47
	σ	0.0041	0.0027	0.0041	0.0027	002.96	001.61
WineEW	μ	0.0022	0.0017	1.0000	1.0000	002.90	002.27
	σ	0.0009	0.0004	0.0000	0.0000	001.16	000.51
Zoo	μ	0.0036	0.0032	1.0000	1.0000	005.80	005.17
	σ	0.0007	0.0002	0.0000	0.0000	001.17	000.37
Ranking	W T L	0 2 16	16 2 0	6 0 12	12 0 6	1 1 16	16 1 1

standard deviation, iBSSA relatively has a higher stability. Thus, iBSSA with SVM outperformed significantly BSSA in solving the FS problem.

5.6.3 Comparisons based on the RF classifier

From Table 14, it is most notable that iBSSA with RF had the potential to find a smaller feature subset over 13 out of

the 18 datasets, yet achieving a higher (or maintaining the same) level of mean accuracy over those datasets. Furthermore, the mean fitness values obtained by iBSSA over the 18 datasets approved its substantial superiority compared to the original BSSA. Additionally, iBSSA–RF stability was very apparent in terms of the three metrics. Notably, iBSSA based on RF classifier showed a higher quality than BSSA alone.

Table 14 Performance comparisons of BSSA and iBSSA in the FS problem based on the RF classifier in terms of the mean fitness value (μ_{Fit}), the mean classification accuracy (μ_{Acc}), and the mean number of selected features (μ_{Feat})

Benchmark	Metric	Fitness		Accuracy		Feature	
		BSSA	iBSSA	BSSA	iBSSA	BSSA	iBSSA
BreastCancer	μ	0.0197	0.0193	0.9857	0.9857	005.57	005.17
	σ	0.0006	0.0004	0.0000	0.0000	000.56	000.37
BreastEW	μ	0.0150	0.0097	0.9895	0.9944	013.83	012.67
	σ	0.0051	0.0048	0.0053	0.0048	003.31	002.41
CongressEW	μ	0.0246	0.0220	0.9782	0.9808	004.77	004.93
	σ	0.0030	0.0045	0.0034	0.0054	001.23	001.53
Exactly	μ	0.2515	0.2384	0.7502	0.7633	005.40	005.37
	σ	0.0582	0.0574	0.0591	0.0584	000.66	000.48
Exactly2	μ	0.2451	0.2384	0.7563	0.7632	005.07	005.07
	σ	0.0035	0.0036	0.0036	0.0035	001.24	000.25
HeartEW	μ	0.0887	0.0746	0.9160	0.9302	007.23	007.23
	σ	0.0120	0.0147	0.0124	0.0156	001.59	001.17
IonosphereEW	μ	0.0399	0.0345	0.9638	0.9690	013.97	013.10
	σ	0.0064	0.0064	0.0070	0.0067	002.86	002.43
KrVsKpEW	μ	0.0642	0.0556	0.9396	0.9487	016.00	017.17
	σ	0.0053	0.0048	0.0054	0.0049	003.26	002.82
Lymphography	μ	0.1484	0.1239	0.8544	0.8800	007.80	009.20
	σ	0.0213	0.0213	0.0219	0.0221	002.14	002.06
M-of-n	μ	0.0108	0.0053	0.9940	0.9993	006.30	006.07
	σ	0.0134	0.0027	0.0131	0.0025	000.53	000.25
PenglungEW	μ	0.2887	0.2532	0.7133	0.7489	159.40	150.33
	σ	0.0346	0.0405	0.0351	0.0410	007.97	012.81
SonarEW	μ	0.1117	0.0831	0.8921	0.9206	029.13	027.33
	σ	0.0167	0.0151	0.0171	0.0155	004.43	003.82
SpectEW	μ	0.1285	0.1203	0.8735	0.8815	007.10	006.57
	σ	0.0118	0.0092	0.0118	0.0091	001.25	001.20
Tic-tac-toe	μ	0.1379	0.1367	0.8684	0.8698	006.83	007.00
	σ	0.0040	0.0000	0.0044	0.0000	000.52	000.00
Vote	μ	0.0023	0.0017	1.0000	1.0000	003.60	002.73
	σ	0.0008	0.0005	0.0000	0.0000	001.25	000.73
WaveformEW	μ	0.1890	0.1829	0.8139	0.8201	019.07	019.33
	σ	0.0063	0.0053	0.0061	0.0051	003.15	003.27
WineEW	μ	0.0026	0.0024	1.0000	1.0000	003.43	003.13
	σ	0.0004	0.0003	0.0000	0.0000	000.50	000.34
Zoo	μ	0.0031	0.0027	1.0000	1.0000	005.03	004.33
	σ	0.0005	0.0003	0.0000	0.0000	000.88	000.47
Ranking	W T L	0 0 18	18 0 0	0 4 14	14 4 0	5 0 13	13 0 5

5.7 Performance comparison with some other promising meta-heuristic algorithms

Previous experiments compared iBSSA with BSSA originated from the original SSA. iBSSA performed superiorly over the original BSSA approach. In fact, these improvements in performance are subjected to the iBSSA capability to balance exploration and exploitation over

iterations, as well as its ability to escape from local optima. In this section, in order to further confirm the superiority of iBSSA to other peers in literature, the best-performing iBSSA method with the three classifiers, *k*-NN, SVM, and RF, is compared with other advanced meta-heuristics implemented in the same conditions. Comparison with iBSSA includes binary variants of some promising optimization algorithms, such as BSSA, BABC, BPSO, BBA,

Table 15 Comparisons of iBSSA against a few promising algorithms based on k -NN in terms of the mean fitness value (μ_{Fit})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Fit}	0.0201	0.0203	0.0227	0.0246	0.0206	0.0210	0.0211	0.0201	0.0215	0.0208	0.0304	0.0232
	σ_{Fit}	0.0000	0.0005	0.0025	0.0037	0.0009	0.0016	0.0016	0.0000	0.0021	0.0015	0.0050	0.0022
BreastEW	μ_{Fit}	0.0363	0.0379	0.0394	0.0428	0.0375	0.0388	0.0381	0.0374	0.0388	0.0389	0.0473	0.0462
	σ_{Fit}	0.0004	0.0006	0.0027	0.0045	0.0005	0.0021	0.0007	0.0003	0.0029	0.0026	0.0066	0.0038
CongressEW	μ_{Fit}	0.0261	0.0274	0.0345	0.0350	0.0281	0.0313	0.0298	0.0260	0.0315	0.0280	0.0382	0.0380
	σ_{Fit}	0.0005	0.0024	0.0041	0.0041	0.0038	0.0047	0.0043	0.0005	0.0045	0.0033	0.0038	0.0007
Exactly	μ_{Fit}	0.0074	0.0132	0.1232	0.1546	0.0260	0.0472	0.0390	0.0147	0.0302	0.0439	0.2710	0.1476
	σ_{Fit}	0.0072	0.0097	0.0764	0.0915	0.0354	0.0518	0.0342	0.0178	0.0374	0.0480	0.0535	0.0796
Exactly2	μ_{Fit}	0.2290	0.2294	0.2332	0.2418	0.2324	0.2331	0.2324	0.2274	0.2369	0.2328	0.2450	0.2438
	σ_{Fit}	0.0019	0.0023	0.0049	0.0054	0.0041	0.0049	0.0037	0.0000	0.0053	0.0038	0.0063	0.0063
HeartEW	μ_{Fit}	0.0953	0.0958	0.1098	0.1178	0.0964	0.1022	0.0989	0.0951	0.0981	0.0986	0.1576	0.1179
	σ_{Fit}	0.0006	0.0008	0.0151	0.0176	0.0033	0.0104	0.0062	0.0005	0.0052	0.0062	0.0405	0.0144
IonosphereEW	μ_{Fit}	0.0640	0.0832	0.0970	0.1086	0.0806	0.0879	0.0878	0.0767	0.0788	0.0838	0.1100	0.1136
	σ_{Fit}	0.0085	0.0078	0.0109	0.0131	0.0141	0.0092	0.0099	0.0081	0.0102	0.0099	0.0083	0.0074
KrVsKpEW	μ_{Fit}	0.0208	0.0254	0.0407	0.0397	0.0265	0.0309	0.0307	0.0279	0.0265	0.0316	0.0611	0.0364
	σ_{Fit}	0.0027	0.0022	0.0085	0.0095	0.0044	0.0053	0.0050	0.0028	0.0057	0.0040	0.0185	0.0066
Lymphography	μ_{Fit}	0.1644	0.1681	0.1926	0.2249	0.1683	0.1806	0.1741	0.1656	0.1771	0.1756	0.2331	0.2156
	σ_{Fit}	0.0108	0.0059	0.0172	0.0225	0.0197	0.0176	0.0162	0.0096	0.0233	0.0129	0.0192	0.0196
M-of-n	μ_{Fit}	0.0047	0.0070	0.0484	0.0491	0.0049	0.0135	0.0124	0.0054	0.0080	0.0097	0.1338	0.0515
	σ_{Fit}	0.0002	0.0048	0.0334	0.0508	0.0010	0.0140	0.0127	0.0021	0.0066	0.0082	0.0408	0.0242
PenglungEW	μ_{Fit}	0.3486	0.3721	0.3962	0.3985	0.3763	0.3983	0.3830	0.3477	0.3803	0.3873	0.4048	0.4011
	σ_{Fit}	0.0325	0.0324	0.0162	0.0116	0.0315	0.0115	0.0288	0.0260	0.0299	0.0261	0.0163	0.0002
SonarEW	μ_{Fit}	0.0124	0.0243	0.0409	0.0571	0.0225	0.0342	0.0275	0.0167	0.0251	0.0274	0.0605	0.0534
	σ_{Fit}	0.0112	0.0084	0.0114	0.0166	0.0098	0.0118	0.0075	0.0114	0.0093	0.0113	0.0167	0.0111
SpectEW	μ_{Fit}	0.1131	0.1192	0.1369	0.1548	0.1153	0.1268	0.1297	0.1116	0.1217	0.1215	0.1611	0.1514
	σ_{Fit}	0.0047	0.0103	0.0122	0.0152	0.0072	0.0126	0.0116	0.0060	0.0113	0.0121	0.0129	0.0118
Tic-tac-toe	μ_{Fit}	0.1544	0.1544	0.1576	0.1611	0.1547	0.1547	0.1551	0.1544	0.1610	0.1554	0.1908	0.1613
	σ_{Fit}	0.0000	0.0000	0.0050	0.0099	0.0018	0.0018	0.0026	0.0000	0.0112	0.0031	0.0166	0.0103
Vote	μ_{Fit}	0.0021	0.0029	0.0108	0.0178	0.0024	0.0038	0.0039	0.0023	0.0060	0.0039	0.0220	0.0188
	σ_{Fit}	0.0004	0.0006	0.0079	0.0067	0.0005	0.0042	0.0040	0.0003	0.0062	0.0042	0.0100	0.0048
WaveformEW	μ_{Fit}	0.1557	0.1624	0.1762	0.1794	0.1636	0.1671	0.1672	0.1618	0.1622	0.1653	0.1932	0.1787
	σ_{Fit}	0.0051	0.0040	0.0053	0.0067	0.0057	0.0059	0.0048	0.0028	0.0066	0.0067	0.0077	0.0045
WineEW	μ_{Fit}	0.0031	0.0033	0.0075	0.0144	0.0034	0.0046	0.0035	0.0031	0.0037	0.0036	0.0325	0.0087
	σ_{Fit}	0.0002	0.0004	0.0091	0.0133	0.0004	0.0049	0.0004	0.0000	0.0008	0.0005	0.0153	0.0075
Zoo	μ_{Fit}	0.0033	0.0036	0.0042	0.0095	0.0036	0.0039	0.0039	0.0034	0.0040	0.0039	0.0112	0.0068
	σ_{Fit}	0.0003	0.0004	0.0006	0.0138	0.0004	0.0005	0.0005	0.0003	0.0006	0.0004	0.0154	0.0080
Ranking	W T L	10 3 5	0 1 17	0 0 18	0 0 18	0 0 18	0 0 18	0 0 18	5 3 10	0 0 18	0 0 18	0 0 18	0 0 18

BGWO, BWOA, BGOA, BSFO, BHHO, BBSA, BASO, and BHGSO. In the same way, the comparison and analysis of performance is according to the mean fitness value, the mean classification accuracy, and the mean number of selected features.

5.7.1 Comparisons based on the k -NN classifier

Table 15 reports the results of the mean fitness values for iBSSA and other advanced meta-heuristics in the FS problem based on k -NN. As can be seen in Table 15, iBSSA won over 10 and tied over 3 datasets in the FS task, thereby having a relatively good impact in 13 out of 18 datasets, accounting for 72% of all datasets. Additionally, there are large and small size datasets in the benchmark

Table 16 Comparisons of iBSSA against a few promising algorithms based on k -NN in terms of the μ_{Acc} classification accuracy (μ_{Acc})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Acc}	0.9857	0.9857	0.9836	0.9817	0.9857	0.9852	0.9852	0.9857	0.9848	0.9855	0.9757	0.9843
	σ_{Acc}	0.0000	0.0000	0.0033	0.0044	0.0000	0.0018	0.0018	0.0000	0.0024	0.0013	0.0057	0.0029
BreastEW	μ_{Acc}	0.9649	0.9649	0.9640	0.9608	0.9649	0.9643	0.9649	0.9649	0.9637	0.9640	0.9567	0.9585
	σ_{Acc}	0.0000	0.0000	0.0026	0.0044	0.0000	0.0022	0.0000	0.0000	0.0030	0.0026	0.0064	0.0039
CongressEW	μ_{Acc}	0.9770	0.9762	0.9682	0.9678	0.9747	0.9713	0.9732	0.9770	0.9705	0.9755	0.9651	0.9655
	σ_{Acc}	0.0000	0.0029	0.0049	0.0046	0.0046	0.0057	0.0054	0.0000	0.0057	0.0039	0.0036	0.0000
Exactly	μ_{Acc}	0.9973	0.9917	0.8818	0.8502	0.9788	0.9577	0.9660	0.9902	0.9747	0.9610	0.7333	0.8575
	σ_{Acc}	0.0070	0.0094	0.0764	0.0919	0.0353	0.0518	0.0341	0.0176	0.0373	0.0480	0.0541	0.0795
Exactly2	μ_{Acc}	0.7737	0.7737	0.7695	0.7607	0.7703	0.7698	0.7707	0.7750	0.7658	0.7695	0.7572	0.7590
	σ_{Acc}	0.0022	0.0022	0.0049	0.0057	0.0045	0.0051	0.0038	0.0000	0.0058	0.0039	0.0073	0.0066
HeartEW	μ_{Acc}	0.9074	0.9074	0.8938	0.8858	0.9068	0.9019	0.9049	0.9074	0.9056	0.9049	0.8457	0.8864
	σ_{Acc}	0.0000	0.0000	0.0158	0.0186	0.0033	0.0109	0.0063	0.0000	0.0056	0.0063	0.0415	0.0149
IonosphereEW	μ_{Acc}	0.9385	0.9197	0.9061	0.8944	0.9221	0.9150	0.9150	0.9258	0.9239	0.9192	0.8930	0.8901
	σ_{Acc}	0.0085	0.0083	0.0111	0.0135	0.0144	0.0093	0.0099	0.0081	0.0100	0.0102	0.0086	0.0076
KrVsKpEW	μ_{Acc}	0.9852	0.9808	0.9649	0.9660	0.9794	0.9751	0.9753	0.9781	0.9794	0.9742	0.9448	0.9701
	σ_{Acc}	0.0027	0.0022	0.0088	0.0101	0.0046	0.0055	0.0051	0.0029	0.0057	0.0042	0.0191	0.0067
Lymphography	μ_{Acc}	0.8378	0.8344	0.8100	0.7778	0.8344	0.8222	0.8289	0.8367	0.8256	0.8267	0.7700	0.7878
	σ_{Acc}	0.0113	0.0060	0.0175	0.0233	0.0202	0.0179	0.0166	0.0100	0.0239	0.0133	0.0199	0.0202
M-of-n	μ_{Acc}	1.0000	0.9978	0.9577	0.9563	0.9998	0.9917	0.9928	0.9995	0.9970	0.9955	0.8717	0.9545
	σ_{Acc}	0.0000	0.0046	0.0330	0.0508	0.0009	0.0136	0.0124	0.0020	0.0064	0.0079	0.0419	0.0239
PenglungEW	μ_{Acc}	0.6511	0.6289	0.6044	0.6022	0.6244	0.6022	0.6178	0.6533	0.6200	0.6133	0.5956	0.6000
	σ_{Acc}	0.0330	0.0330	0.0166	0.0120	0.0321	0.0120	0.0295	0.0267	0.0306	0.0267	0.0166	0.0000
SonarEW	μ_{Acc}	0.9913	0.9802	0.9635	0.9468	0.9817	0.9698	0.9770	0.9873	0.9786	0.9770	0.9437	0.9516
	σ_{Acc}	0.0115	0.0089	0.0119	0.0170	0.0101	0.0122	0.0075	0.0119	0.0094	0.0115	0.0168	0.0115
SpectEW	μ_{Acc}	0.8889	0.8840	0.8660	0.8481	0.8870	0.8759	0.8735	0.8914	0.8809	0.8815	0.8420	0.8525
	σ_{Acc}	0.0048	0.0106	0.0124	0.0154	0.0073	0.0128	0.0118	0.0063	0.0114	0.0123	0.0133	0.0122
Tic-tac-toe	μ_{Acc}	0.8542	0.8542	0.8493	0.8453	0.8536	0.8536	0.8531	0.8542	0.8460	0.8526	0.8135	0.8453
	σ_{Acc}	0.0000	0.0000	0.0075	0.0119	0.0028	0.0028	0.0039	0.0000	0.0128	0.0047	0.0167	0.0121
Vote	μ_{Acc}	1.0000	1.0000	0.9922	0.9850	1.0000	0.9989	0.9989	1.0000	0.9967	0.9989	0.9811	0.9850
	σ_{Acc}	0.0000	0.0000	0.0083	0.0066	0.0000	0.0042	0.0042	0.0000	0.0067	0.0042	0.0103	0.0050
WaveformEW	μ_{Acc}	0.8484	0.8424	0.8277	0.8249	0.8405	0.8369	0.8370	0.8426	0.8422	0.8390	0.8112	0.8258
	σ_{Acc}	0.0053	0.0040	0.0054	0.0071	0.0057	0.0061	0.0048	0.0029	0.0067	0.0067	0.0079	0.0046
WineEW	μ_{Acc}	1.0000	1.0000	0.9963	0.9898	1.0000	0.9991	1.0000	1.0000	1.0000	1.0000	0.9722	0.9972
	σ_{Acc}	0.0000	0.0000	0.0094	0.0134	0.0000	0.0050	0.0000	0.0000	0.0000	0.0000	0.0160	0.0083
Zoo	μ_{Acc}	1.0000	1.0000	1.0000	0.9952	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9937	0.9984
	σ_{Acc}	0.0000	0.0000	0.0000	0.0143	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0162	0.0085
Ranking	W T L	7 8 3	0 7 11	0 11 17	0 0 18	0 5 13	0 11 17	0 3 15	3 8 7	0 2 16	0 2 16	0 0 18	0 0 18

used, which proves iBSSA ability to consistently perform on all datasets regardless of the dataset size. For the 5 datasets lost by iBSSA, comparing the mean fitness values reveals that these values by iBSSA are very close to other algorithms. The same observation is highlighted for the 5 datasets won by BSFO in the FS problem, where their mean fitness values are very close to those in the case of iBSSA. This shows that the best performance is by iBSSA. Except BSFO, none of the other methods in comparison

with iBSSA ranked first in the 18 datasets, which confirms the remarkability of the proposed method with k -NN. Furthermore, the stability of iBSSA with k -NN is relatively strong, based on the standard deviations of the various methods. Now, it can be concluded that iBSSA is highly advantageous, with a proven capability of balancing between exploration and exploitation in the search space over iterations and avoiding local optima. Whereas, other algorithms may become trapped into local optima.

Table 17 Comparisons of iBSSA against a few promising algorithms based on k -NN in terms of the mean number of selected features (μ_{Feat})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Feat}	006.00	006.13	006.40	006.43	006.43	006.43	006.53	006.00	006.43	006.43	006.40	007.60
	σ_{Feat}	000.00	000.50	001.60	001.36	000.88	001.12	000.99	000.00	001.12	000.88	001.33	001.43
BreastEW	μ_{Feat}	004.60	009.43	011.27	012.00	008.43	010.57	010.23	007.97	008.73	009.87	013.43	015.23
	σ_{Feat}	001.23	001.73	001.75	002.03	001.56	001.98	002.06	001.05	002.06	001.63	002.91	001.45
CongressEW	μ_{Feat}	005.40	006.27	004.90	005.10	004.90	004.50	005.27	005.20	003.67	006.03	005.97	006.17
	σ_{Feat}	000.88	001.53	001.90	001.56	001.76	001.84	002.16	000.75	002.10	001.60	001.58	001.10
Exactly	μ_{Feat}	006.13	006.47	008.10	008.20	006.57	006.90	007.00	006.43	006.67	006.93	009.13	008.50
	σ_{Feat}	000.34	000.50	001.04	001.28	000.67	000.79	000.63	000.56	000.70	000.73	001.82	001.18
Exactly2	μ_{Feat}	006.47	006.97	006.47	006.37	006.53	006.77	007.00	006.00	006.53	006.00	005.97	006.83
	σ_{Feat}	001.06	001.02	001.02	001.74	001.09	001.23	001.39	000.00	001.61	001.15	002.23	001.85
HeartEW	μ_{Feat}	004.77	005.43	006.03	006.17	005.37	006.53	006.23	004.40	005.97	005.83	006.27	007.03
	σ_{Feat}	000.84	001.09	001.38	001.53	001.35	001.26	001.17	000.71	001.17	001.49	001.34	001.14
IonosphereEW	μ_{Feat}	010.63	012.57	013.70	013.80	011.57	012.97	012.57	011.13	011.93	013.07	013.73	016.53
	σ_{Feat}	002.86	002.63	002.95	002.93	002.54	001.87	002.93	002.81	002.41	002.89	003.44	003.11
KrVsKpEW	μ_{Feat}	021.97	023.13	021.77	021.87	021.77	022.47	022.60	022.40	021.90	021.73	023.23	024.33
	σ_{Feat}	002.58	002.35	002.65	003.52	002.50	002.28	002.65	001.89	002.74	002.45	004.34	003.01
Lymphography	μ_{Feat}	006.90	007.57	008.07	008.73	007.97	008.20	008.40	007.10	007.90	007.27	009.77	009.87
	σ_{Feat}	001.27	001.54	001.84	001.88	001.47	001.19	001.28	001.16	001.54	001.55	001.89	001.67
M-of-n	μ_{Feat}	006.07	006.37	008.50	007.67	006.20	006.83	006.90	006.37	006.50	006.77	008.73	008.40
	σ_{Feat}	000.25	000.55	001.48	001.27	000.40	000.86	000.75	000.48	000.56	000.72	002.02	000.99
PenglungEW	μ_{Feat}	103.57	152.07	148.27	152.00	146.20	145.17	148.60	147.20	134.10	146.73	143.63	165.53
	σ_{Feat}	012.64	015.28	011.73	019.21	015.03	012.77	014.88	016.38	013.30	014.05	022.31	007.81
SonarEW	μ_{Feat}	022.43	027.67	028.33	026.83	026.30	026.20	028.13	024.87	023.57	027.77	028.27	033.10
	σ_{Feat}	003.77	004.21	003.52	004.57	003.36	003.30	004.09	004.13	003.72	003.48	005.64	003.99
SpectEW	μ_{Feat}	006.77	009.40	009.43	009.83	007.57	008.70	009.70	008.93	008.27	009.20	010.33	011.83
	σ_{Feat}	001.84	001.62	001.80	002.10	001.71	002.02	001.77	002.61	001.84	001.96	002.33	002.33
Tic-tac-toe	μ_{Feat}	009.00	009.00	007.53	007.13	008.83	008.83	008.70	009.00	007.70	008.53	005.60	007.33
	σ_{Feat}	000.00	000.00	002.25	002.16	000.90	000.90	001.13	000.00	001.93	001.41	001.28	002.12
Vote	μ_{Feat}	003.37	004.67	005.03	004.70	003.80	004.33	004.53	003.70	004.33	004.53	005.33	006.30
	σ_{Feat}	000.60	000.94	001.25	001.46	000.83	001.08	000.96	000.46	001.35	000.96	001.37	001.07
WaveformEW	μ_{Feat}	022.43	025.30	022.60	024.20	022.83	022.53	023.40	023.70	023.93	023.57	025.10	025.03
	σ_{Feat}	002.40	003.84	002.68	003.95	003.01	003.04	002.63	002.44	002.74	002.60	004.46	002.54
WineEW	μ_{Feat}	004.07	004.33	005.03	005.67	004.37	004.73	004.53	004.00	004.80	004.63	006.47	007.73
	σ_{Feat}	000.25	000.47	001.02	001.27	000.55	001.03	000.56	000.00	000.98	000.71	001.69	001.77
Zoo	μ_{Feat}	005.33	005.77	006.73	007.67	005.80	006.30	006.30	005.37	006.33	006.17	007.87	008.43
	σ_{Feat}	000.47	000.62	001.00	001.45	000.65	000.86	000.74	000.48	000.91	000.64	002.05	001.33
Ranking	W T L	11 1 6	0 0 18	0 0 18	0 0 18	0 0 18	0 0 18	0 0 18	2 11 15	1 0 17	1 0 17	2 0 16	0 0 18

Table 16 shows the mean classification accuracy results of k -NN with iBSSA compared to other methods. From Table 16, it is observed that iBSSA excelled all other algorithms in terms of mean classification accuracy over 7 while tied with other algorithms over 8 datasets, achieving overall high accuracy in 15 out of the 18 datasets. Additionally, note that the second ranked optimizer (BSFO) acquired a slight advantage over 3 datasets only: Exactly2,

PenglungEW, and SpectEW, with a margin of 0.0013%, 0.0022%, and 0.25%, respectively.

In addition, as shown in Table 17, iBSSA has a better exploration ability than other methods based on the mean number of selected features by iBSSA, which was asserted by selecting less number of features over 12 (11 wins and one tie) out of the 18 datasets, rather than BSFO, BHHO, BBSA, and BASO which have selected a lower yet non-significant number of features than iBSSA and other

Table 18 Comparisons of iBSSA against a few promising algorithms based on SVM in terms of the mean fitness value (μ_{Fit})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Fit}	0.0262	0.0263	0.0314	0.0339	0.0271	0.0287	0.0274	0.0262	0.0291	0.0281	0.0382	0.0320
	σ_{Fit}	0.0000	0.0003	0.0032	0.0061	0.0020	0.0029	0.0020	0.0000	0.0030	0.0029	0.0037	0.0031
BreastEW	μ_{Fit}	0.0545	0.0606	0.0639	0.0711	0.0581	0.0624	0.0633	0.0561	0.0592	0.0607	0.0708	0.0654
	σ_{Fit}	0.0007	0.0041	0.0027	0.0092	0.0040	0.0034	0.0046	0.0020	0.0050	0.0043	0.0081	0.0007
CongressEW	μ_{Fit}	0.0249	0.0261	0.0284	0.0318	0.0256	0.0270	0.0266	0.0254	0.0259	0.0262	0.0359	0.0301
	σ_{Fit}	0.0004	0.0006	0.0029	0.0046	0.0006	0.0021	0.0020	0.0004	0.0007	0.0022	0.0046	0.0037
Exactly	μ_{Fit}	0.2572	0.2612	0.3006	0.3248	0.2835	0.2969	0.2892	0.2589	0.3044	0.3012	0.3321	0.3160
	σ_{Fit}	0.0003	0.0137	0.0311	0.0135	0.0301	0.0305	0.0318	0.0084	0.0310	0.0289	0.0040	0.0236
Exactly2	μ_{Fit}	0.2483	0.2486	0.2492	0.2494	0.2483	0.2487	0.2485	0.2483	0.2483	0.2487	0.2499	0.2497
	σ_{Fit}	0.0000	0.0004	0.0005	0.0006	0.0001	0.0005	0.0004	0.0000	0.0000	0.0004	0.0006	0.0006
HeartEW	μ_{Fit}	0.0950	0.0980	0.1187	0.1405	0.1053	0.1115	0.1108	0.0953	0.1078	0.1100	0.1669	0.1465
	σ_{Fit}	0.0004	0.0061	0.0169	0.0240	0.0123	0.0131	0.0138	0.0004	0.0129	0.0168	0.0340	0.0224
IonosphereEW	μ_{Fit}	0.0352	0.0451	0.0589	0.0671	0.0441	0.0504	0.0525	0.0412	0.0457	0.0533	0.0783	0.0641
	σ_{Fit}	0.0080	0.0074	0.0082	0.0121	0.0067	0.0080	0.0075	0.0064	0.0094	0.0067	0.0098	0.0076
KrVsKpEW	μ_{Fit}	0.0207	0.0266	0.0395	0.0402	0.0256	0.0315	0.0324	0.0284	0.0278	0.0304	0.0539	0.0362
	σ_{Fit}	0.0028	0.0031	0.0083	0.0083	0.0040	0.0048	0.0037	0.0023	0.0050	0.0031	0.0166	0.0037
Lymphography	μ_{Fit}	0.1527	0.1633	0.1852	0.1975	0.1654	0.1773	0.1726	0.1518	0.1740	0.1699	0.2188	0.1924
	σ_{Fit}	0.0163	0.0131	0.0178	0.0202	0.0140	0.0182	0.0141	0.0162	0.0182	0.0118	0.0218	0.0151
M-of-n	μ_{Fit}	0.0048	0.0053	0.0266	0.0419	0.0050	0.0102	0.0056	0.0051	0.0052	0.0056	0.1149	0.0116
	σ_{Fit}	0.0003	0.0004	0.0511	0.0642	0.0005	0.0251	0.0004	0.0004	0.0006	0.0006	0.0701	0.0269
PenglungEW	μ_{Fit}	0.1746	0.2004	0.2027	0.2027	0.1933	0.1980	0.1980	0.1889	0.1887	0.2002	0.2138	0.2032
	σ_{Fit}	0.0321	0.0118	0.0002	0.0170	0.0224	0.0164	0.0164	0.0260	0.0263	0.0118	0.0243	0.0001
SonarEW	μ_{Fit}	0.0556	0.0828	0.1132	0.1291	0.0784	0.0925	0.0951	0.0737	0.0781	0.0887	0.1362	0.1216
	σ_{Fit}	0.0142	0.0150	0.0198	0.0207	0.0132	0.0173	0.0160	0.0102	0.0158	0.0168	0.0218	0.0177
SpectEW	μ_{Fit}	0.1658	0.1854	0.2233	0.2320	0.1977	0.2072	0.1981	0.1754	0.2009	0.1962	0.2499	0.2382
	σ_{Fit}	0.0104	0.0186	0.0244	0.0259	0.0267	0.0281	0.0230	0.0097	0.0300	0.0232	0.0075	0.0173
Tic-tac-toe	μ_{Fit}	0.1017	0.1017	0.1206	0.1197	0.1017	0.1051	0.1018	0.1017	0.1018	0.1017	0.1602	0.1129
	σ_{Fit}	0.0000	0.0000	0.0191	0.0251	0.0000	0.0090	0.0004	0.0000	0.0004	0.0002	0.0320	0.0120
Vote	μ_{Fit}	0.0023	0.0033	0.0090	0.0136	0.0030	0.0042	0.0043	0.0026	0.0032	0.0033	0.0243	0.0107
	σ_{Fit}	0.0005	0.0008	0.0072	0.0101	0.0008	0.0032	0.0029	0.0006	0.0008	0.0010	0.0115	0.0076
WaveformEW	μ_{Fit}	0.1256	0.1315	0.1468	0.1435	0.1328	0.1357	0.1380	0.1333	0.1320	0.1352	0.1536	0.1412
	σ_{Fit}	0.0027	0.0029	0.0051	0.0076	0.0033	0.0045	0.0050	0.0028	0.0044	0.0046	0.0070	0.0048
WineEW	μ_{Fit}	0.0017	0.0027	0.0174	0.0205	0.0033	0.0075	0.0060	0.0019	0.0026	0.0038	0.0356	0.0189
	σ_{Fit}	0.0004	0.0007	0.0138	0.0135	0.0050	0.0104	0.0081	0.0004	0.0009	0.0051	0.0201	0.0131
Zoo	μ_{Fit}	0.0032	0.0038	0.0064	0.0170	0.0038	0.0055	0.0043	0.0034	0.0052	0.0039	0.0361	0.0129
	σ_{Fit}	0.0002	0.0005	0.0082	0.0201	0.0006	0.0083	0.0008	0.0003	0.0083	0.0008	0.0212	0.0167
Ranking	W T L	14 3 1	0 1 17	0 0 18	0 0 18	0 2 16	0 0 18	0 0 18	1 2 15	0 1 17	0 1 17	0 0 18	0 0 18

algorithms over only 2, 1, 1, and 2 datasets, respectively. This proves the ability of iBSSA with k -NN to explore the most feasible regions and avoid searching irrelevant search space areas. Thus, iBSSA based on k -NN can reduce the feature search space through the identification of the most informative features while maintaining high classification accuracy.

5.7.2 Comparisons based on the SVM classifier

By inspecting Table 18, iBSSA with SVM outperformed other optimizers assessed in the same experimental environment, in 14 out of 18 datasets because it has the minimal classification error compared to other methods, while BSFO provides the highest yet slight advantage over one dataset only. Additionally, the standard deviation values are close to the mean fitness, which implies a good balance

Table 19 Comparisons of iBSSA against a few promising algorithms based on SVM in terms of the mean classification accuracy (μ_{Acc})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Acc}	0.9786	0.9786	0.9733	0.9707	0.9776	0.9760	0.9776	0.9786	0.9755	0.9767	0.9664	0.9729
	σ_{Acc}	0.0000	0.0000	0.0037	0.0065	0.0024	0.0034	0.0024	0.0000	0.0035	0.0032	0.0042	0.0034
BreastEW	μ_{Acc}	0.9474	0.9424	0.9395	0.9325	0.9444	0.9404	0.9395	0.9468	0.9433	0.9421	0.9330	0.9386
	σ_{Acc}	0.0000	0.0043	0.0026	0.0096	0.0041	0.0035	0.0047	0.0022	0.0049	0.0043	0.0083	0.0000
CongressEW	μ_{Acc}	0.9770	0.9770	0.9759	0.9724	0.9770	0.9766	0.9766	0.9770	0.9770	0.9766	0.9682	0.9747
	σ_{Acc}	0.0000	0.0000	0.0034	0.0056	0.0000	0.0021	0.0021	0.0000	0.0000	0.0021	0.0049	0.0046
Exactly	μ_{Acc}	0.7450	0.7415	0.7022	0.6777	0.7187	0.7060	0.7135	0.7435	0.6973	0.7013	0.6683	0.6870
	σ_{Acc}	0.0000	0.0132	0.0311	0.0139	0.0305	0.0304	0.0318	0.0081	0.0316	0.0292	0.0051	0.0234
Exactly2	μ_{Acc}	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500	0.7500
	σ_{Acc}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HeartEW	μ_{Acc}	0.9074	0.9049	0.8840	0.8617	0.8975	0.8914	0.8920	0.9074	0.8951	0.8926	0.8358	0.8562
	σ_{Acc}	0.0000	0.0063	0.0172	0.0247	0.0124	0.0133	0.0144	0.0000	0.0129	0.0168	0.0347	0.0228
IonosphereEW	μ_{Acc}	0.9681	0.9592	0.9451	0.9371	0.9596	0.9535	0.9516	0.9624	0.9577	0.9502	0.9258	0.9404
	σ_{Acc}	0.0081	0.0076	0.0084	0.0125	0.0070	0.0083	0.0079	0.0066	0.0096	0.0070	0.0102	0.0079
KrVsKpEW	μ_{Acc}	0.9854	0.9799	0.9661	0.9657	0.9807	0.9742	0.9738	0.9773	0.9781	0.9757	0.9518	0.9700
	σ_{Acc}	0.0027	0.0033	0.0087	0.0089	0.0043	0.0052	0.0038	0.0025	0.0053	0.0031	0.0173	0.0038
Lymphography	μ_{Acc}	0.8500	0.8400	0.8178	0.8056	0.8378	0.8256	0.8311	0.8511	0.8289	0.8333	0.7844	0.8111
	σ_{Acc}	0.0167	0.0133	0.0187	0.0212	0.0142	0.0186	0.0147	0.0166	0.0187	0.0122	0.0223	0.0157
M-of-n	μ_{Acc}	1.0000	1.0000	0.9795	0.9642	1.0000	0.9953	1.0000	1.0000	1.0000	1.0000	0.8905	0.9950
	σ_{Acc}	0.0000	0.0000	0.0523	0.0650	0.0000	0.0251	0.0000	0.0000	0.0000	0.0000	0.0718	0.0269
PenglungEW	μ_{Acc}	0.8267	0.8022	0.8000	0.8000	0.8089	0.8044	0.8044	0.8133	0.8133	0.8022	0.7889	0.8000
	σ_{Acc}	0.0327	0.0120	0.0000	0.0172	0.0227	0.0166	0.0166	0.0267	0.0267	0.0120	0.0248	0.0000
SonarEW	μ_{Acc}	0.9476	0.9214	0.8905	0.8746	0.9254	0.9111	0.9087	0.9302	0.9254	0.9151	0.8675	0.8825
	σ_{Acc}	0.0143	0.0152	0.0200	0.0212	0.0134	0.0173	0.0164	0.0105	0.0160	0.0170	0.0219	0.0184
SpectEW	μ_{Acc}	0.8364	0.8173	0.7790	0.7710	0.8043	0.7951	0.8043	0.8272	0.8006	0.8062	0.7525	0.7648
	σ_{Acc}	0.0108	0.0190	0.0243	0.0260	0.0273	0.0283	0.0233	0.0100	0.0305	0.0238	0.0089	0.0173
Tic-tac-toe	μ_{Acc}	0.9062	0.9062	0.8868	0.8875	0.9062	0.9028	0.9062	0.9062	0.9062	0.9062	0.8457	0.8948
	σ_{Acc}	0.0000	0.0000	0.0199	0.0261	0.0000	0.0094	0.0000	0.0000	0.0000	0.0000	0.0332	0.0128
Vote	μ_{Acc}	1.0000	1.0000	0.9950	0.9906	1.0000	0.9994	0.9994	1.0000	1.0000	1.0000	0.9800	0.9933
	σ_{Acc}	0.0000	0.0000	0.0076	0.0103	0.0000	0.0030	0.0030	0.0000	0.0000	0.0000	0.0117	0.0082
WaveformEW	μ_{Acc}	0.8791	0.8738	0.8579	0.8611	0.8722	0.8694	0.8669	0.8718	0.8728	0.8696	0.8512	0.8639
	σ_{Acc}	0.0027	0.0028	0.0052	0.0079	0.0037	0.0045	0.0049	0.0028	0.0045	0.0047	0.0075	0.0051
WineEW	μ_{Acc}	1.0000	1.0000	0.9861	0.9833	0.9991	0.9954	0.9972	1.0000	1.0000	0.9991	0.9685	0.9852
	σ_{Acc}	0.0000	0.0000	0.0139	0.0136	0.0050	0.0104	0.0083	0.0000	0.0000	0.0050	0.0199	0.0139
Zoo	μ_{Acc}	1.0000	1.0000	0.9984	0.9873	1.0000	0.9984	1.0000	1.0000	0.9984	1.0000	0.9683	0.9921
	σ_{Acc}	0.0000	0.0000	0.0085	0.0211	0.0000	0.0085	0.0000	0.0000	0.0085	0.0000	0.0224	0.0177
Ranking	W T L	8 9 1	0 8 10	0 11 7	1 0 17	0 6 12	0 11 7	0 4 14	1 9 8	0 6 12	0 5 13	0 11 7	0 11 7

between exploration and exploitation of the SVM-based iBSSA method.

In Table 19, iBSSA is compared with other algorithms in terms of mean classification accuracy. By analyzing the results, iBSSA based on SVM had a perfect recognition in terms of 100% classification accuracy on the datasets M-of-n, Vote, WineEW, and Zoo; whereas, it achieved an accuracy varying from 94.76% to 98.54% on the datasets SonarEW, BreastEW, IonosphereEW, CongressEW,

BreastCancer, and KrVsKpEW. The results also revealed that iBSSA based on SVM obtained the best mean accuracy on 8 out of the 18 datasets, while the second ranked optimizer (BSFO) scored the best results over one dataset only.

In terms of the mean number of selected features, the results of iBSSA and other competitors based on SVM are depicted in Table 20. By analyzing the results, appealing observations are remarked for iBSSA with SVM, which

Table 20 Comparisons of iBSSA against a few promising algorithms based on SVM in terms of the mean number of selected features (μ_{Feat})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Feat}	005.00	005.10	004.97	004.93	004.97	004.93	005.20	005.00	004.83	005.00	004.97	005.10
	σ_{Feat}	000.00	000.30	000.91	000.89	000.48	000.68	000.65	000.00	000.69	000.52	001.40	000.75
BreastEW	μ_{Feat}	007.13	010.70	011.87	012.80	009.33	010.10	010.03	010.20	009.13	010.23	013.53	013.93
	σ_{Feat}	002.05	001.27	001.50	002.66	001.49	001.94	001.25	001.49	001.78	001.43	002.81	002.22
CongressEW	μ_{Feat}	003.50	005.33	007.13	007.20	004.47	006.10	005.57	004.20	004.97	004.87	007.00	008.03
	σ_{Feat}	000.72	001.01	002.06	002.34	000.92	001.70	001.50	000.65	001.14	001.06	001.57	002.37
Exactly	μ_{Feat}	006.20	006.87	007.53	007.40	006.53	007.60	007.27	006.43	006.23	007.17	004.83	008.03
	σ_{Feat}	000.40	000.85	001.67	002.32	001.54	001.58	001.59	000.67	002.69	001.49	002.65	001.62
Exactly2	μ_{Feat}	001.00	001.43	002.27	002.47	001.03	001.50	001.30	001.00	001.00	001.53	003.13	002.80
	σ_{Feat}	000.00	000.50	000.63	000.76	000.18	000.62	000.46	000.00	000.00	000.50	000.81	000.79
HeartEW	μ_{Feat}	004.33	005.00	005.00	004.67	005.07	005.10	005.00	004.67	005.03	004.77	005.60	005.33
	σ_{Feat}	000.47	001.00	001.18	001.49	001.03	001.01	001.00	000.54	001.11	000.96	001.56	001.16
IonosphereEW	μ_{Feat}	012.23	015.87	015.47	016.27	014.13	014.93	015.67	013.63	013.30	013.87	016.60	017.37
	σ_{Feat}	002.54	002.49	003.04	003.03	003.29	002.73	002.65	002.29	002.73	003.14	003.08	003.23
KrVsKpEW	μ_{Feat}	022.60	024.03	021.47	022.73	023.47	021.40	023.20	021.57	021.87	022.70	022.13	023.37
	σ_{Feat}	002.59	002.59	003.69	003.55	003.61	003.53	003.13	002.87	003.15	002.79	002.77	002.74
Lymphography	μ_{Feat}	007.53	008.83	008.57	009.07	008.70	008.33	009.67	007.93	008.37	008.87	009.80	009.73
	σ_{Feat}	001.28	001.07	002.04	002.71	001.37	002.01	002.34	001.31	001.30	001.65	002.06	001.97
M-of-n	μ_{Feat}	006.27	006.83	008.13	008.40	006.53	007.23	007.27	006.57	006.73	007.30	008.43	008.67
	σ_{Feat}	000.44	000.58	001.41	001.62	000.62	000.88	000.57	000.50	000.73	000.82	001.76	000.98
PenglungEW	μ_{Feat}	096.90	150.67	152.37	151.20	134.33	142.80	143.90	133.17	126.37	142.73	155.13	167.57
	σ_{Feat}	010.21	008.10	007.89	014.75	006.86	007.01	005.78	013.22	006.28	006.14	014.26	003.94
SonarEW	μ_{Feat}	022.27	030.20	028.90	029.53	026.97	026.83	028.30	027.63	025.67	027.83	030.00	031.67
	σ_{Feat}	003.02	004.00	004.32	004.38	003.34	003.53	003.26	003.72	004.22	002.85	004.11	004.68
SpectEW	μ_{Feat}	008.43	010.03	009.97	011.57	008.83	009.40	009.70	009.43	007.70	009.40	010.63	011.70
	σ_{Feat}	001.78	001.14	002.09	002.11	001.86	002.23	001.55	001.50	002.45	001.47	003.65	001.53
Tic-tac-toe	μ_{Feat}	008.00	008.00	007.67	007.53	008.00	008.00	008.13	008.00	008.13	008.03	006.63	007.87
	σ_{Feat}	000.00	000.00	000.83	000.92	000.00	000.45	000.34	000.00	000.34	000.18	001.14	000.85
Vote	μ_{Feat}	003.60	005.20	006.47	006.73	004.73	005.83	005.93	004.17	005.07	005.33	007.27	006.63
	σ_{Feat}	000.80	001.28	001.73	001.93	001.24	001.66	001.12	000.93	001.29	001.60	001.69	001.66
WaveformEW	μ_{Feat}	023.47	026.30	024.37	024.10	024.80	025.53	024.93	025.57	024.20	024.50	025.27	025.53
	σ_{Feat}	001.61	002.35	002.76	003.17	002.98	002.23	002.62	002.72	002.29	002.55	003.47	003.01
WineEW	μ_{Feat}	002.27	003.53	004.77	005.17	003.13	003.77	004.20	002.47	003.40	003.70	005.70	005.50
	σ_{Feat}	000.51	000.92	001.23	001.42	000.96	001.05	001.19	000.50	001.14	001.22	001.24	001.50
Zoo	μ_{Feat}	005.17	006.03	007.77	007.07	006.07	006.30	006.93	005.40	005.80	006.30	007.53	008.03
	σ_{Feat}	000.37	000.88	001.65	001.95	000.93	001.07	001.24	000.55	000.91	001.22	001.82	001.76
Ranking	W T L	12 1 5	010 18	010 18	010 18	010 18	110 17	010 18	011 17	211 15	010 18	210 16	010 18

achieved better results than other optimizers over 12 out of the 18 datasets adopted in this study. It is also observed that the second ranked method (BHHO) outperforms over two datasets only. The superiority of iBSSA with SVM in this regard was demonstrated thanks to the proposed 3RA and LSA methods, which implies that the proposed iBSSA would select fewer number of features, affirming its potential to explore the most important areas of the search

space, avoiding searching through non feasible spaces areas.

5.7.3 Comparisons based on the RF classifier

Table 21 compares the mean fitness and standard deviation values of iBSSA with other competitors based on RF, under the same implementation conditions. Notably, iBSSA based on RF shows a higher quality than other

Table 21 Comparisons of iBSSA against a few promising algorithms based on RF in terms of the mean fitness value (μ_{Fit})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Fit}	0.0193	0.0192	0.0206	0.0222	0.0194	0.0196	0.0196	0.0191	0.0201	0.0197	0.0248	0.0211
	σ_{Fit}	0.0004	0.0003	0.0020	0.0028	0.0004	0.0005	0.0006	0.0000	0.0012	0.0005	0.0035	0.0022
BreastEW	μ_{Fit}	0.0097	0.0119	0.0245	0.0305	0.0147	0.0186	0.0167	0.0114	0.0188	0.0154	0.0271	0.0277
	σ_{Fit}	0.0048	0.0049	0.0055	0.0096	0.0056	0.0044	0.0048	0.0032	0.0066	0.0057	0.0087	0.0070
CongressEW	μ_{Fit}	0.0220	0.0247	0.0267	0.0308	0.0245	0.0263	0.0252	0.0232	0.0261	0.0253	0.0361	0.0292
	σ_{Fit}	0.0045	0.0035	0.0028	0.0054	0.0030	0.0008	0.0032	0.0041	0.0009	0.0026	0.0056	0.0042
Exactly	μ_{Fit}	0.2384	0.2511	0.3038	0.3081	0.2787	0.2882	0.2839	0.2349	0.2758	0.2639	0.3179	0.3041
	σ_{Fit}	0.0574	0.0579	0.0088	0.0105	0.0391	0.0250	0.0331	0.0588	0.0510	0.0509	0.0126	0.0275
Exactly2	μ_{Fit}	0.2384	0.2416	0.2466	0.2479	0.2430	0.2449	0.2433	0.2378	0.2459	0.2431	0.2496	0.2475
	σ_{Fit}	0.0036	0.0044	0.0028	0.0021	0.0048	0.0037	0.0047	0.0022	0.0037	0.0042	0.0011	0.0039
HeartEW	μ_{Fit}	0.0746	0.0746	0.0984	0.1070	0.0880	0.0919	0.0839	0.0779	0.0951	0.0881	0.1192	0.1031
	σ_{Fit}	0.0147	0.0104	0.0109	0.0158	0.0107	0.0094	0.0138	0.0089	0.0111	0.0109	0.0136	0.0114
IonosphereEW	μ_{Fit}	0.0345	0.0351	0.0450	0.0513	0.0417	0.0442	0.0431	0.0353	0.0450	0.0415	0.0516	0.0519
	σ_{Fit}	0.0064	0.0047	0.0041	0.0089	0.0060	0.0056	0.0054	0.0053	0.0050	0.0062	0.0065	0.0062
KrVsKpEW	μ_{Fit}	0.0556	0.0617	0.0800	0.0886	0.0599	0.0685	0.0674	0.0596	0.0660	0.0662	0.1147	0.0830
	σ_{Fit}	0.0048	0.0045	0.0100	0.0210	0.0059	0.0077	0.0069	0.0041	0.0064	0.0075	0.0235	0.0087
Lymphography	μ_{Fit}	0.1239	0.1310	0.1729	0.1864	0.1445	0.1608	0.1485	0.1280	0.1556	0.1477	0.2119	0.1868
	σ_{Fit}	0.0213	0.0134	0.0254	0.0278	0.0235	0.0234	0.0224	0.0200	0.0311	0.0244	0.0289	0.0202
M-of-n	μ_{Fit}	0.0053	0.0249	0.1012	0.1016	0.0195	0.0577	0.0520	0.0153	0.0242	0.0348	0.1632	0.0951
	σ_{Fit}	0.0027	0.0183	0.0411	0.0468	0.0241	0.0383	0.0305	0.0129	0.0247	0.0291	0.0280	0.0312
PenglungEW	μ_{Fit}	0.2532	0.2695	0.2931	0.3348	0.2843	0.2953	0.2931	0.2532	0.3061	0.3019	0.2997	0.3507
	σ_{Fit}	0.0405	0.0294	0.0358	0.0339	0.0368	0.0401	0.0432	0.0277	0.0404	0.0370	0.0406	0.0324
SonarEW	μ_{Fit}	0.0831	0.0947	0.1225	0.1362	0.1077	0.1141	0.1108	0.0917	0.1208	0.1071	0.1545	0.1469
	σ_{Fit}	0.0151	0.0164	0.0228	0.0247	0.0196	0.0176	0.0224	0.0148	0.0208	0.0153	0.0222	0.0191
SpectEW	μ_{Fit}	0.1203	0.1406	0.1490	0.1594	0.1386	0.1393	0.1420	0.1288	0.1383	0.1442	0.1693	0.1565
	σ_{Fit}	0.0092	0.0090	0.0101	0.0131	0.0120	0.0155	0.0102	0.0093	0.0122	0.0120	0.0138	0.0104
Tic-tac-toe	μ_{Fit}	0.1367	0.1367	0.1487	0.1546	0.1370	0.1396	0.1385	0.1367	0.1420	0.1393	0.1731	0.1514
	σ_{Fit}	0.0000	0.0000	0.0123	0.0163	0.0015	0.0068	0.0044	0.0000	0.0079	0.0048	0.0134	0.0094
Vote	μ_{Fit}	0.0017	0.0028	0.0067	0.0124	0.0023	0.0028	0.0030	0.0020	0.0026	0.0028	0.0208	0.0119
	σ_{Fit}	0.0005	0.0005	0.0060	0.0078	0.0007	0.0006	0.0006	0.0004	0.0008	0.0006	0.0076	0.0078
WaveformEW	μ_{Fit}	0.1829	0.1920	0.2050	0.2117	0.1925	0.1970	0.1977	0.1891	0.1961	0.1930	0.2236	0.2064
	σ_{Fit}	0.0053	0.0048	0.0069	0.0078	0.0055	0.0058	0.0046	0.0055	0.0058	0.0064	0.0092	0.0074
WineEW	μ_{Fit}	0.0024	0.0028	0.0030	0.0037	0.0027	0.0030	0.0027	0.0023	0.0030	0.0026	0.0044	0.0042
	σ_{Fit}	0.0003	0.0005	0.0006	0.0009	0.0005	0.0006	0.0004	0.0001	0.0006	0.0005	0.0009	0.0006
Zoo	μ_{Fit}	0.0027	0.0033	0.0038	0.0044	0.0034	0.0034	0.0035	0.0030	0.0034	0.0035	0.0064	0.0046
	σ_{Fit}	0.0003	0.0004	0.0005	0.0006	0.0005	0.0004	0.0004	0.0003	0.0005	0.0004	0.0083	0.0006
Ranking	W T L	12 2 4	0 2 16	0 0 18	0 0 18	0 0 18	0 0 18	0 0 18	4 2 12	0 0 18	0 0 18	0 0 18	0 0 18

approaches. By analyzing Table 21, evidence shows that RF-based iBSSA obtained the lowest fitness values along with competitive standard deviations over 12 out of the 18 datasets. In addition, the second ranked optimizer (BSFO) acquired a slight advantage over 4 datasets only: Breast-Cancer, Exactly, Exactly2, and WineEW with a margin of 0.02%, 0.35%, 0.06%, and 0.01%, respectively.

Table 22 shows the comparison of iBSSA and other algorithms with RF classifier in terms of accuracy results

which are assessed fairly in the same environment. The experimental results reveal that RF-based iBSSA classifier is superior on 8 out of 18 datasets, where BSFO, which ranks second, gave the best results on two datasets only. It should also be pointed out that BSFO produced the same results on 6 datasets as iBSSA based on RF.

Table 23 presents the mean number of features selected by each algorithm. It is evident that iBSSA excelled other algorithms over 7 out of 18 datasets in feature reduction.

Table 22 Comparisons of iBSSA against a few promising algorithms based on RF in terms of the mean classification accuracy (μ_{Acc})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Acc}	0.9857	0.9857	0.9848	0.9833	0.9857	0.9857	0.9857	0.9857	0.9855	0.9857	0.9807	0.9845
	σ_{Acc}	0.0000	0.0000	0.0024	0.0034	0.0000	0.0000	0.0000	0.0000	0.0013	0.0000	0.0038	0.0027
BreastEW	μ_{Acc}	0.9944	0.9930	0.9801	0.9737	0.9895	0.9860	0.9880	0.9930	0.9857	0.9892	0.9775	0.9772
	σ_{Acc}	0.0048	0.0053	0.0060	0.0099	0.0057	0.0043	0.0053	0.0035	0.0070	0.0059	0.0090	0.0070
CongressEW	μ_{Acc}	0.9808	0.9785	0.9770	0.9728	0.9782	0.9770	0.9782	0.9797	0.9770	0.9778	0.9682	0.9747
	σ_{Acc}	0.0054	0.0039	0.0030	0.0055	0.0034	0.0000	0.0034	0.0049	0.0000	0.0029	0.0057	0.0046
Exactly	μ_{Acc}	0.7633	0.7508	0.6975	0.6932	0.7227	0.7133	0.7175	0.7670	0.7257	0.7375	0.6837	0.6975
	σ_{Acc}	0.0584	0.0587	0.0088	0.0104	0.0397	0.0252	0.0335	0.0596	0.0517	0.0517	0.0123	0.0278
Exactly2	μ_{Acc}	0.7632	0.7598	0.7550	0.7527	0.7585	0.7565	0.7583	0.7637	0.7547	0.7583	0.7505	0.7528
	σ_{Acc}	0.0035	0.0044	0.0032	0.0031	0.0049	0.0039	0.0045	0.0022	0.0046	0.0043	0.0015	0.0046
HeartEW	μ_{Acc}	0.9302	0.9309	0.9062	0.8975	0.9167	0.9130	0.9210	0.9272	0.9099	0.9167	0.8852	0.9019
	σ_{Acc}	0.0156	0.0106	0.0116	0.0164	0.0115	0.0097	0.0143	0.0095	0.0114	0.0115	0.0139	0.0119
IonosphereEW	μ_{Acc}	0.9690	0.9695	0.9592	0.9526	0.9620	0.9596	0.9606	0.9685	0.9587	0.9624	0.9526	0.9526
	σ_{Acc}	0.0067	0.0052	0.0042	0.0093	0.0065	0.0060	0.0056	0.0060	0.0051	0.0066	0.0068	0.0068
KrVsKpEW	μ_{Acc}	0.9487	0.9431	0.9244	0.9157	0.9443	0.9361	0.9369	0.9446	0.9384	0.9382	0.8893	0.9217
	σ_{Acc}	0.0049	0.0048	0.0103	0.0212	0.0061	0.0077	0.0069	0.0042	0.0065	0.0076	0.0234	0.0089
Lymphography	μ_{Acc}	0.8800	0.8733	0.8300	0.8167	0.8589	0.8422	0.8556	0.8756	0.8478	0.8556	0.7911	0.8167
	σ_{Acc}	0.0221	0.0133	0.0263	0.0282	0.0239	0.0242	0.0233	0.0210	0.0318	0.0248	0.0297	0.0206
M-of-n	μ_{Acc}	0.9993	0.9802	0.9042	0.9035	0.9853	0.9477	0.9533	0.9897	0.9807	0.9702	0.8413	0.9110
	σ_{Acc}	0.0025	0.0182	0.0413	0.0475	0.0238	0.0382	0.0303	0.0128	0.0245	0.0288	0.0287	0.0309
PenglungEW	μ_{Acc}	0.7489	0.7333	0.7089	0.6667	0.7178	0.7067	0.7089	0.7489	0.6956	0.7000	0.7022	0.6511
	σ_{Acc}	0.0410	0.0298	0.0364	0.0344	0.0373	0.0407	0.0438	0.0282	0.0410	0.0375	0.0412	0.0330
SonarEW	μ_{Acc}	0.9206	0.9095	0.8810	0.8675	0.8960	0.8897	0.8929	0.9119	0.8825	0.8968	0.8484	0.8571
	σ_{Acc}	0.0155	0.0167	0.0230	0.0251	0.0199	0.0179	0.0228	0.0152	0.0212	0.0155	0.0226	0.0194
SpectEW	μ_{Acc}	0.8815	0.8617	0.8537	0.8432	0.8636	0.8630	0.8605	0.8735	0.8636	0.8580	0.8340	0.8469
	σ_{Acc}	0.0091	0.0092	0.0100	0.0133	0.0122	0.0156	0.0104	0.0097	0.0122	0.0120	0.0139	0.0106
Tic-tac-toe	μ_{Acc}	0.8698	0.8698	0.8569	0.8510	0.8694	0.8667	0.8677	0.8698	0.8639	0.8667	0.8312	0.8540
	σ_{Acc}	0.0000	0.0000	0.0127	0.0169	0.0019	0.0072	0.0050	0.0000	0.0084	0.0055	0.0142	0.0096
Vote	μ_{Acc}	1.0000	1.0000	0.9972	0.9911	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9833	0.9917
	σ_{Acc}	0.0000	0.0000	0.0062	0.0083	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0075	0.0083
WaveformEW	μ_{Acc}	0.8201	0.8118	0.7980	0.7915	0.8107	0.8063	0.8054	0.8140	0.8070	0.8102	0.7796	0.7971
	σ_{Acc}	0.0051	0.0046	0.0069	0.0082	0.0054	0.0058	0.0047	0.0053	0.0058	0.0067	0.0092	0.0073
WineEW	μ_{Acc}	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	σ_{Acc}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Zoo	μ_{Acc}	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9984	1.0000
	σ_{Acc}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0085	0.0000
Ranking	W T L	8 6 4	2 5 11	0 2 16	0 2 16	0 4 14	0 4 14	0 4 14	2 6 10	0 3 15	0 4 14	0 11 7	0 2 16

Furthermore, BASO excelled other algorithms over only 4 datasets, BSFO excelled other algorithms over only 3 datasets, and BGWO excelled other algorithms over only 2 datasets; whereas, BWOA and BBSA excelled other algorithms for only one dataset. The iBSSA ability to select fewer number of features is believed to be attributed to the superiority of the proposed 3RA and LSA methods in selecting the most informative features through searching

within feasible areas while at the same time considering improved classification accuracy.

5.8 Convergence analysis

This subsection presents an asymptotic analysis of the proposed iBSSA with the three classifiers, k -NN, SVM, and RF, in FS on different datasets. These analyses reveal the convergence capacity of the proposed iBSSA in the form of

Table 23 Comparisons of iBSSA against a few promising algorithms based on RF in terms of the mean number of selected features (μ_{Feat})

Benchmark	Metric	iBSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	μ_{Feat}	005.17	005.10	005.47	005.73	005.23	005.47	005.50	005.00	005.77	005.57	005.70	005.73
	σ_{Feat}	000.37	000.30	000.76	000.96	000.42	000.50	000.56	000.00	000.84	000.50	000.90	000.77
BreastEW	μ_{Feat}	012.67	014.90	014.37	013.37	012.90	014.20	014.40	013.33	013.77	014.03	014.43	015.30
	σ_{Feat}	002.41	002.71	003.05	002.63	002.44	002.59	003.12	003.06	002.50	002.94	003.03	002.90
CongressEW	μ_{Feat}	004.93	005.47	006.33	006.13	004.63	005.60	005.67	004.90	005.30	005.33	007.40	006.73
	σ_{Feat}	001.53	000.85	000.94	001.38	001.05	001.28	000.98	001.27	001.37	001.08	001.94	001.36
Exactly	μ_{Feat}	005.37	005.70	005.67	005.67	005.40	005.70	005.43	005.53	005.53	005.27	006.13	006.00
	σ_{Feat}	000.48	000.82	000.83	001.11	000.84	000.97	000.76	000.62	000.85	000.73	001.23	001.00
Exactly2	μ_{Feat}	005.07	005.03	005.23	004.00	005.10	005.03	005.30	005.00	003.93	005.00	003.37	003.63
	σ_{Feat}	000.25	000.18	001.63	001.77	000.94	001.08	000.46	000.00	002.34	000.63	001.08	001.62
HeartEW	μ_{Feat}	007.23	008.03	007.20	007.23	007.20	007.40	007.40	007.53	007.63	007.23	007.13	007.70
	σ_{Feat}	001.17	000.98	001.68	001.67	001.60	001.20	000.95	001.02	001.43	001.17	001.36	001.35
IonosphereEW	μ_{Feat}	013.10	016.70	015.57	014.87	013.73	014.40	013.80	014.23	014.00	014.83	015.80	016.87
	σ_{Feat}	002.43	002.77	002.63	002.57	002.91	002.30	001.87	003.30	002.31	002.50	003.50	002.45
KrVsKpEW	μ_{Feat}	017.17	019.23	018.40	018.43	016.90	018.97	017.90	017.07	017.87	018.23	018.53	019.90
	σ_{Feat}	002.82	003.13	003.14	002.82	002.89	002.59	002.84	002.19	002.74	002.59	003.25	002.72
Lymphography	μ_{Feat}	009.20	010.07	008.33	008.83	008.63	008.27	009.90	008.57	008.73	008.47	009.10	009.57
	σ_{Feat}	002.06	001.95	002.09	002.44	001.76	002.21	002.37	002.14	001.86	002.08	002.23	001.86
M-of-n	μ_{Feat}	006.07	006.80	008.20	007.93	006.50	007.60	007.57	006.53	006.63	006.90	008.00	009.07
	σ_{Feat}	000.25	000.48	001.25	001.29	000.67	001.05	000.76	000.50	000.66	000.87	001.79	001.26
PenglungEW	μ_{Feat}	150.33	178.63	159.10	155.03	158.57	159.07	157.77	149.40	151.33	158.63	158.60	173.83
	σ_{Feat}	012.81	015.84	014.04	016.78	008.94	010.31	012.53	011.79	010.41	011.92	011.57	012.64
SonarEW	μ_{Feat}	027.33	030.73	027.80	029.90	028.87	029.47	028.40	027.07	027.10	030.03	026.63	032.80
	σ_{Feat}	003.82	004.34	003.31	005.11	003.99	004.23	003.49	003.64	003.88	003.85	004.47	003.30
SpectEW	μ_{Feat}	006.57	008.17	009.23	009.27	007.87	007.97	008.63	007.73	007.13	008.07	010.87	010.80
	σ_{Feat}	001.20	001.39	002.81	001.41	000.92	000.88	001.92	001.34	001.45	001.73	002.67	002.50
Tic-tac-toe	μ_{Feat}	007.00	007.00	006.33	006.43	006.93	006.83	006.73	007.00	006.53	006.53	005.43	006.17
	σ_{Feat}	000.00	000.00	000.87	001.02	000.36	000.45	000.68	000.00	000.76	000.85	001.09	000.78
Vote	μ_{Feat}	002.73	004.47	006.37	005.77	003.67	004.53	004.83	003.27	004.23	004.53	006.90	005.80
	σ_{Feat}	000.73	000.81	001.87	001.94	001.11	000.88	001.04	000.63	001.33	000.99	001.51	001.58
WaveformEW	μ_{Feat}	019.33	022.57	019.93	021.27	020.57	020.90	020.10	020.10	020.03	020.40	021.63	022.07
	σ_{Feat}	003.27	003.27	001.91	003.49	002.45	003.28	002.70	003.49	002.97	003.29	003.30	002.02
WineEW	μ_{Feat}	003.13	003.60	003.93	004.87	003.57	003.93	003.53	003.03	003.93	003.43	005.77	005.43
	σ_{Feat}	000.34	000.61	000.73	001.18	000.67	000.73	000.56	000.18	000.73	000.62	001.17	000.72
Zoo	μ_{Feat}	004.33	005.33	006.07	007.10	005.40	005.50	005.57	004.80	005.37	005.67	007.73	007.30
	σ_{Feat}	000.47	000.70	000.73	000.98	000.76	000.72	000.67	000.40	000.88	000.65	001.48	000.90
Ranking	W T L	7 0 11	0 0 18	0 0 18	0 0 18	2 0 16	1 0 17	0 0 18	3 0 15	0 0 18	1 0 17	4 0 14	0 0 18

a convergence graph. Since the mean classification accuracy can be adopted to quantify the performance of iBSSA in FS problems, yet the algorithm convergence behavior strongly determines its efficiency. Figures 3, 4, and 5 depict the relation between the number of iterations and the optimal fitness value obtained so far with k -NN, SVM, and RF, respectively, for iBSSA against binary variants of other optimizers, including BSSA, BABC, BPSO, BBA, BGWO, BWOA, BGOA, BSFO, BHHO, BBSA, BASO, and

BHGSO, which are executed in the same experimental settings, including the number of agents and number of iterations.

Based on Figs. 3, 4, and 5, iBSSA with k -NN excelled other algorithms in terms of fast convergence over the 18 datasets. Thus, the superior performance of iBSSA with k -NN is revealed through its speedy convergent behavior in comparison with other methods, especially for large datasets like, for example, PenglungEW where convergent

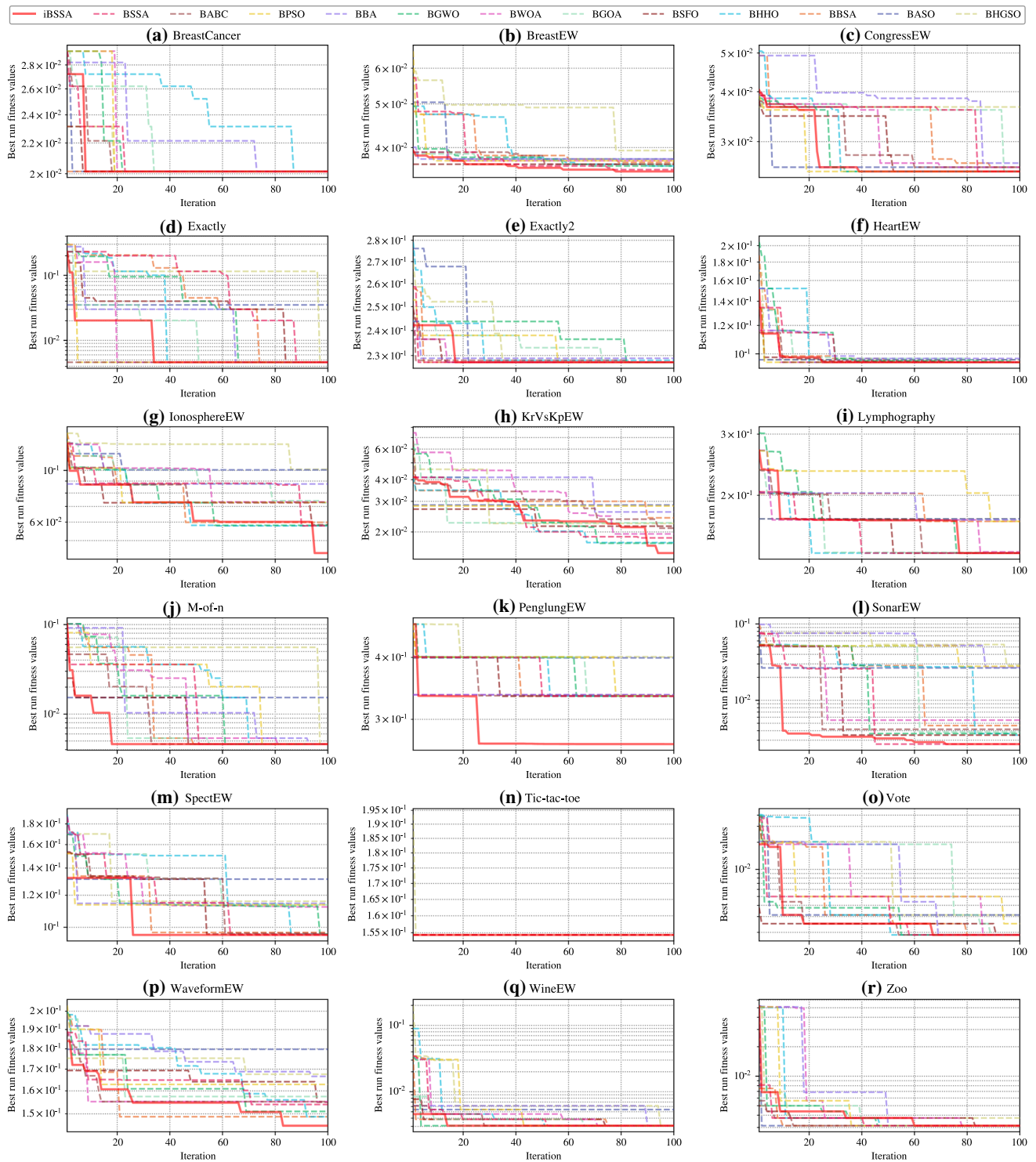


Fig. 3 Convergence curves of the proposed iBSSA and other algorithms over all datasets with the *k*-NN classifier

behavior was exhibited after only 25 iterations. This is comparable to both: the iBSSA with SVM, which indicates speedy convergent behavior over 16 datasets except BreastEW and SonarEW, and the iBSSA with RF which indicates divergent behavior over only the dataset

SonarEW. Moreover, the convergence behavior shows that the proposed iBSSA effectively balances between exploitation and exploration, that is well demonstrated by the perfect coincidence between the optimal mean fitness values and the optimal mean accuracy values. This really

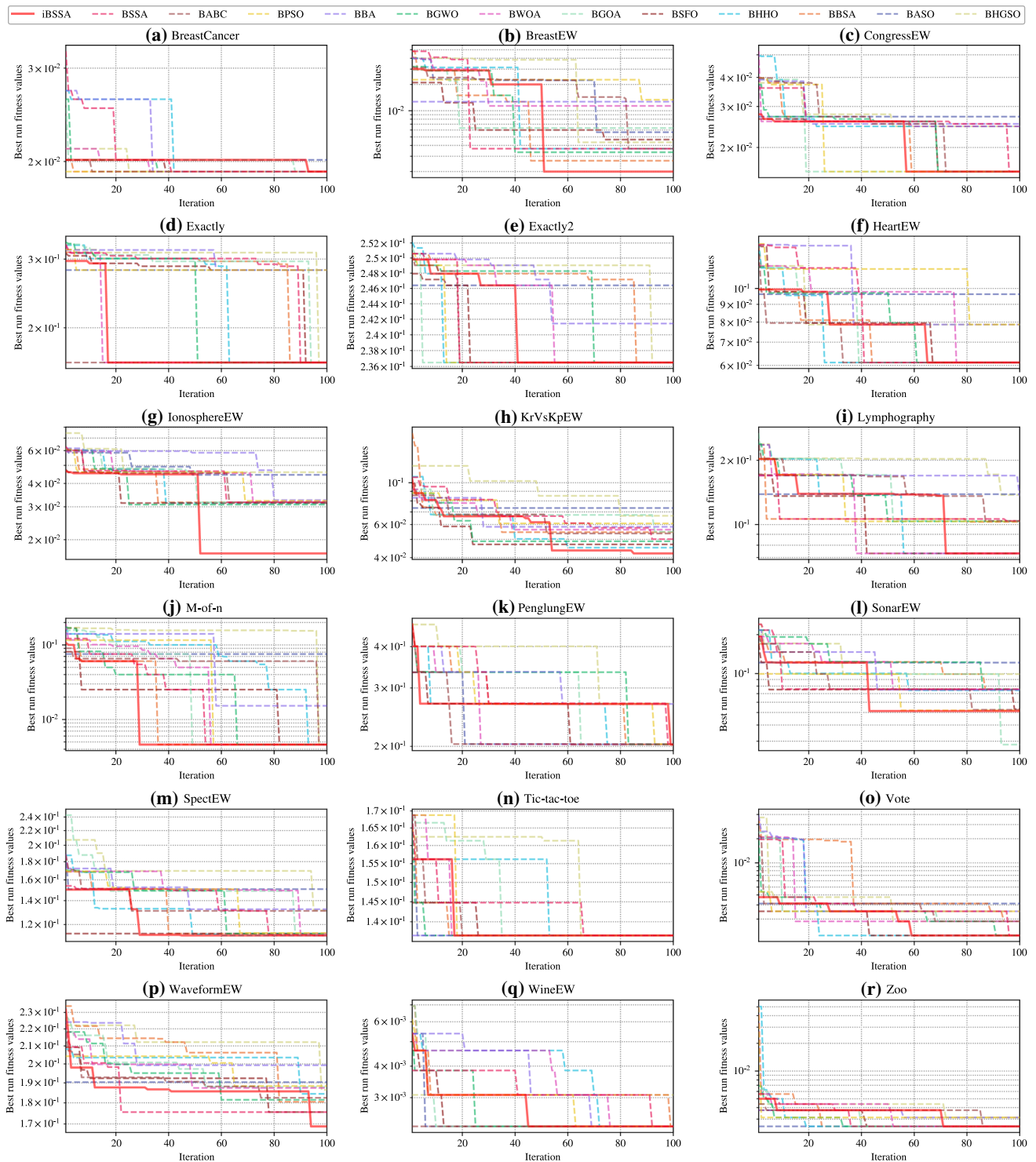


Fig. 4 Convergence curves of the proposed iBSSA and other algorithms over all datasets with the SVM classifier

contributed a lot to solving the problem of premature convergence compared to other optimization algorithms. Notably, this stability in iBSSA performance is reflected by incorporating two improvements, 3RA and LSA, into the

original BSSA algorithm. It is worth mentioning that iBSSA based on the three classifiers: *k*-NN, SVM, and RF, exhibited convergent behavior, in most cases, throughout the 100 iterations, which were thus sufficient for

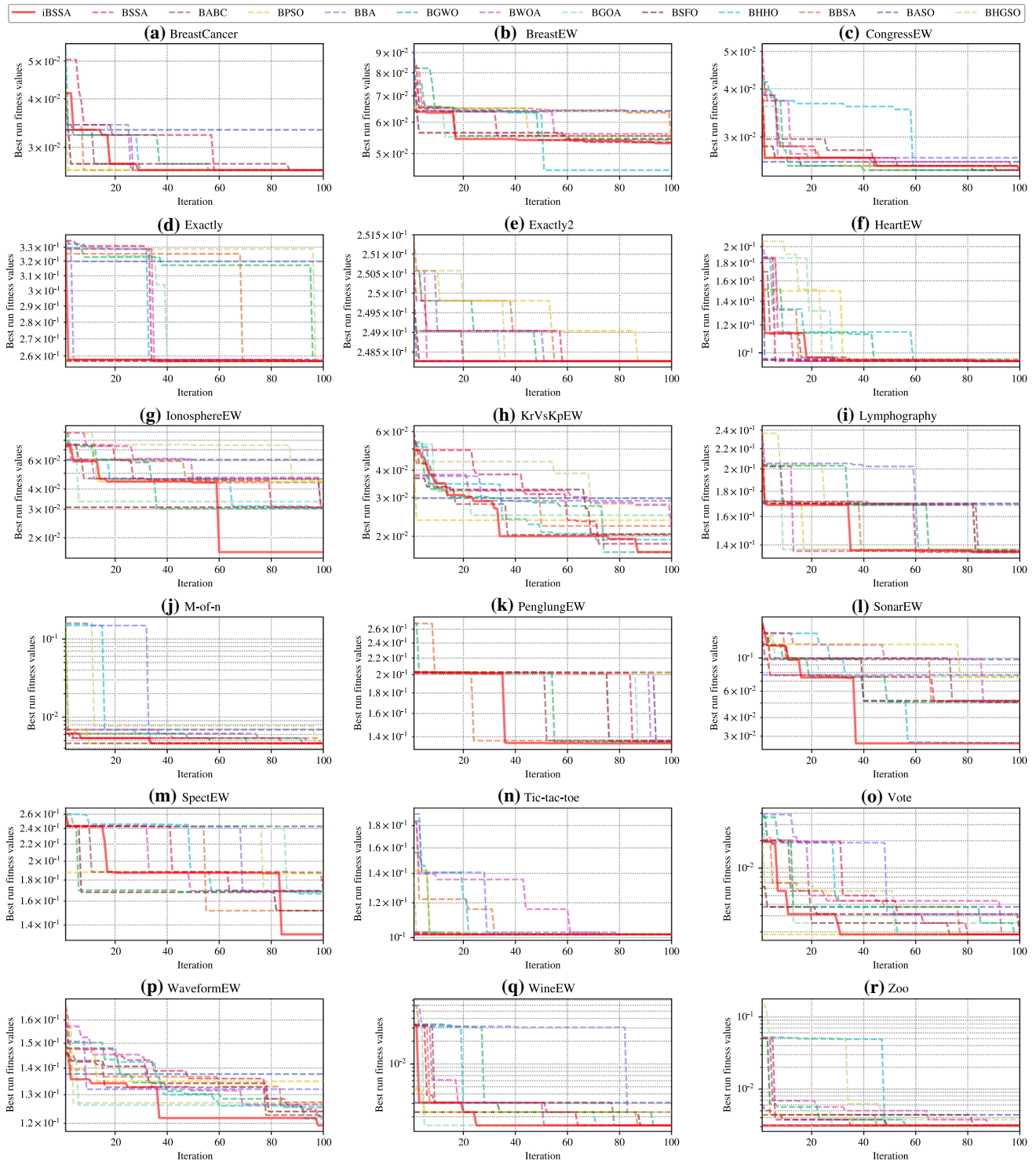


Fig. 5 Convergence curves of the proposed iBSSA and other algorithms over all datasets with the RF classifier

convergence. In addition, the fastest convergence was relatively exhibited by iBSSA with the three classifiers on the vast majority of datasets.

5.9 Statistical analysis (Wilcoxon’s rank-sum test)

Based on Tables 15, 18, and 21, it is obvious that the iBSSA with *k*-NN, SVM, and RF, respectively, shows a

Table 24 p -values of the Wilcoxon's rank-sum test for the mean fitness values (μ_{Fit}) of the iBSSA vs. other approaches based on k -NN ($\alpha < 0.05$ are **bold**)

Benchmark	BSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	6.99E-03	9.54E-02	1.43E-04	1.59E-05	2.80E-02	2.63E-02	1.48E-02	N/A	9.81E-03	5.87E-02	4.44E-10	1.37E-06
BreastEW	5.59E-05	7.06E-11	4.38E-05	9.41E-07	4.10E-09	2.87E-05	6.04E-11	8.76E-11	1.31E-03	2.69E-04	5.50E-08	1.19E-12
CongressEW	2.94E-02	3.49E-02	6.06E-10	1.47E-10	3.75E-02	6.83E-05	1.30E-03	3.29E-01	2.14E-05	2.48E-02	4.34E-15	2.66E-36
Exactly	6.27E-02	5.28E-02	3.86E-07	1.16E-07	3.98E-02	3.67E-03	7.32E-04	1.09E-01	1.89E-02	4.05E-03	5.59E-21	1.78E-08
Exactly2	1.29E-03	3.36E-01	2.42E-03	5.81E-11	4.03E-03	2.99E-03	1.98E-03	1.50E-03	1.48E-06	8.20E-04	6.59E-12	5.00E-11
HeartEW	1.19E-02	4.50E-02	3.80E-04	7.31E-06	1.50E-01	1.02E-02	2.29E-02	2.18E-01	3.36E-02	3.51E-02	2.67E-07	1.83E-07
IonosphereEW	8.65E-05	5.45E-08	1.07E-11	7.88E-14	2.11E-04	2.72E-09	7.55E-09	8.59E-05	5.70E-05	3.47E-07	8.95E-18	1.58E-19
KrVsKpEW	2.95E-03	4.23E-06	6.13E-11	2.54E-09	6.50E-05	3.58E-08	2.07E-08	7.43E-09	7.36E-04	5.68E-11	1.46E-10	1.01E-10
Lymphography	3.64E-01	1.75E-01	1.78E-06	7.14E-12	3.00E-01	2.85E-03	4.56E-02	3.72E-01	4.70E-02	9.91E-03	6.22E-15	3.23E-11
M-of-n	1.60E-01	5.34E-02	4.95E-06	1.04E-03	2.78E-01	1.43E-02	1.76E-02	1.47E-01	4.51E-02	1.71E-02	4.06E-15	1.99E-09
PenglungEW	1.76E-02	4.07E-02	4.80E-06	8.58E-07	1.67E-02	9.07E-07	2.62E-03	3.94E-01	5.81E-03	5.50E-04	2.42E-07	1.01E-07
SonarEW	2.79E-03	1.37E-03	1.24E-08	6.06E-11	8.70E-03	3.31E-06	5.32E-05	2.05E-01	1.06E-03	4.64E-04	1.01E-11	1.08E-12
SpectEW	1.33E-01	3.24E-02	7.89E-09	8.69E-13	2.18E-01	1.86E-04	3.94E-06	2.78E-01	6.80E-03	1.19E-02	2.06E-16	1.66E-14
Tic-tac-toe	1.46E-01	N/A	1.28E-02	8.86E-03	3.07E-01	3.07E-01	2.05E-01	N/A	2.07E-02	1.55E-01	9.51E-11	9.50E-03
Vote	8.54E-02	6.09E-05	6.85E-05	1.82E-11	5.78E-02	9.35E-02	6.78E-02	9.54E-02	1.44E-02	7.92E-02	1.04E-09	2.65E-16
WaveformEW	2.39E-04	1.55E-04	1.57E-13	1.19E-13	1.57E-04	7.02E-07	7.21E-08	1.29E-04	3.01E-03	4.15E-05	2.03E-18	5.62E-16
WineEW	1.97E-01	6.80E-02	5.14E-02	1.38E-03	9.41E-03	1.70E-01	8.20E-04	3.96E-01	5.25E-03	5.50E-04	2.31E-09	4.30E-03
Zoo	3.99E-02	1.87E-02	3.24E-06	6.71E-02	1.87E-02	1.64E-04	1.64E-04	2.38E-01	1.37E-04	1.95E-05	4.04E-02	7.33E-02
Ranking (W T L)	11 0 7	11 1 7	17 0 1	17 0 1	12 0 6	15 0 3	16 0 2	5 1 13	18 0 0	15 0 3	18 0 0	17 0 1

clear superiority over other competing algorithms in 13, 14, and 12 out of the 18 datasets. Therefore, in this section, the Wilcoxon's rank-sum test is assessed for iBSSA with the three classifiers, k -NN, SVM, and RF, compared to BSSA, BABC, BPSO, BBA, BGWO, BWOA, BGOA, BSFO, BHHO, BBSA, BASO, and BHGSO. In order to statistically detect the significant difference between the means of two samples, the Wilcoxon's rank-sum test [26] is used. Therefore, for assessing the significance of the proposed iBSSA against counterparts, Wilcoxon's rank-sum statistical analysis is conducted at a 0.05 significance level under

which a statistically significant difference of the proposed algorithm is exhibited compared to other algorithms (values with $p < 0.05$ are **bold**). The results shown in Tables 24, 25, and 26 reveal that the suggested iBSSA approach is arguably the best compared to other optimizers over the three classifiers, k -NN, SVM, and RF, based on the fitness value results.

By inspecting Table 24's results, iBSSA with k -NN is significantly different from BHHO and BASSO over the 18 datasets since p -value is less than 0.05. Additionally, compared to BPSO, BBA, and BHGSO, iBSSA with k -NN

Table 25 p -values of the Wilcoxon's rank-sum test for the mean fitness values (μ_{Fit}) of the iBSSA vs. other approaches based on SVM ($\alpha < 0.05$ are **bold**)

Benchmark	BSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	4.50E-02	1.46E-01	8.92E-08	8.89E-06	6.66E-02	1.19E-03	1.87E-02	N/A	3.49E-04	1.10E-02	1.96E-15	4.26E-09
BreastEW	7.20E-03	6.61E-07	6.23E-16	1.02E-08	8.99E-04	3.70E-11	3.33E-09	3.92E-03	5.34E-04	1.15E-06	8.52E-10	7.09E-33
CongressEW	3.33E-04	5.47E-08	2.07E-05	4.64E-07	3.33E-04	2.89E-04	1.65E-03	9.27E-04	1.17E-05	2.22E-02	1.13E-11	1.60E-06
Exactly	1.63E-05	1.83E-01	1.63E-06	2.49E-21	1.04E-03	5.40E-06	2.17E-04	2.72E-01	3.25E-07	3.25E-07	7.91E-41	3.44E-12
Exactly2	3.96E-01	4.14E-03	1.01E-08	6.74E-09	3.96E-01	2.40E-03	4.50E-02	N/A	N/A	2.34E-04	5.40E-13	1.94E-11
HeartEW	1.22E-01	4.85E-02	1.50E-06	3.16E-09	1.59E-03	9.27E-06	3.93E-05	3.49E-02	2.58E-04	8.38E-04	2.35E-10	2.86E-11
IonosphereEW	3.34E-03	6.96E-04	4.01E-10	8.84E-11	1.32E-03	3.16E-06	1.62E-07	2.14E-02	1.36E-03	2.27E-08	4.49E-16	8.86E-13
KrVsKpEW	1.06E-05	1.32E-06	1.63E-10	6.50E-11	2.24E-04	1.77E-09	2.49E-12	2.08E-10	1.19E-05	2.20E-11	1.26E-09	8.12E-16
Lymphography	1.58E-01	4.25E-02	3.04E-06	2.53E-08	2.03E-02	2.15E-04	5.84E-04	3.90E-01	1.07E-03	1.30E-03	6.82E-12	1.19E-08
M-of-n	1.38E-01	2.34E-04	7.92E-02	2.29E-02	1.38E-01	2.59E-01	1.22E-07	1.87E-02	1.94E-02	2.15E-05	1.77E-07	2.21E-01
PenglungEW	2.95E-02	3.95E-03	1.02E-03	3.21E-03	5.38E-02	1.17E-02	1.17E-02	1.35E-01	1.40E-01	4.20E-03	3.21E-04	8.54E-04
SonarEW	1.56E-03	4.44E-06	1.38E-11	3.87E-14	2.64E-05	6.63E-08	5.74E-09	1.52E-04	1.13E-04	4.07E-07	7.55E-15	4.69E-14
SpectEW	1.52E-05	6.10E-04	1.27E-10	1.27E-11	5.80E-05	1.95E-06	7.13E-06	9.02E-03	6.39E-05	2.06E-05	3.28E-25	9.45E-17
Tic-tac-toe	N/A	N/A	2.65E-04	5.86E-03	N/A	1.11E-01	2.24E-01	N/A	2.24E-01	3.96E-01	7.18E-09	5.19E-04
Vote	2.42E-01	1.12E-04	5.51E-04	5.51E-05	4.51E-03	2.12E-02	8.61E-03	1.06E-01	4.06E-04	8.20E-04	2.62E-09	6.60E-05
WaveformEW	2.55E-03	4.97E-07	4.54E-17	7.02E-11	4.02E-08	2.23E-09	1.07E-10	1.15E-09	1.18E-05	1.02E-08	2.80E-17	9.95E-14
WineEW	4.22E-02	1.17E-05	4.30E-05	1.71E-06	1.58E-01	2.76E-02	3.49E-02	1.29E-01	6.54E-04	8.88E-02	4.15E-08	4.70E-06
Zoo	2.96E-02	5.74E-05	1.02E-01	8.03E-03	4.33E-04	1.97E-01	3.57E-06	2.82E-02	2.33E-01	1.39E-03	2.24E-07	2.23E-02
Ranking (W T L)	12 1 6	15 1 3	16 0 2	18 0 0	12 1 6	15 0 3	17 0 1	10 3 8	14 1 4	16 0 2	18 0 0	17 0 1

remains statistically significant on 17 datasets. While iBSSA is significantly different from BGOA, BWOA, BBSA, BGWO, BABC, BSSA, and BSFO with k -NN over 16, 15, 15, 12, 11, 11, and 5 datasets, respectively. On the other hand, iBSSA and BSFO exhibit the same performance with k -NN for the dataset BreastCancer because the mean fitness value is 0.0201 and the standard deviation is zero. The same observation can be made for iBSSA, BABC, and BSFO, where the same performance is also exhibited with k -NN for the dataset Tic-tac-toe as the mean fitness value is 0.1544 and the standard deviation is zero.

Furthermore, iBSSA with k -NN does not perform significantly for the datasets Lymphography and M-of-n compared to BSSA, BABC, BGWO, and BSFO. In addition, iBSSA with k -NN are not significantly different from the results of BSSA, BGWO, BWOA, BGOA, BSFP, and BBSA on the datasets Tic-tac-toe and Vote. Lastly, iBSSA with k -NN does not also differ significantly from BSSA, BABC, BPSO, BWOA, and BSFO on the dataset WineEW. In general, it is clear from the table that iBSSA has statistically significant differences on most of the datasets compared to other algorithms.

Table 26 p -values of the Wilcoxon's rank-sum test for the mean fitness values (μ_{Fit}) of the iBSSA vs. other approaches based on RF ($\alpha < 0.05$ are bold)

Benchmark	BSSA	BABC	BPSO	BBA	BGWO	BWOA	BGOA	BSFO	BHHO	BBSA	BASO	BHGSO
BreastCancer	2.82E-02	2.74E-01	1.31E-02	1.72E-04	2.97E-01	5.78E-02	8.54E-02	4.50E-02	1.37E-02	1.48E-02	1.99E-07	2.27E-03
BreastEW	3.85E-03	1.57E-01	6.53E-10	1.90E-09	8.76E-03	2.35E-06	1.60E-04	1.80E-01	5.67E-05	3.52E-03	1.85E-08	2.18E-10
CongressEW	5.25E-02	5.56E-02	8.96E-04	1.01E-05	6.07E-02	4.75E-04	2.26E-02	2.77E-01	8.30E-04	1.34E-02	1.41E-09	2.86E-05
Exactly	3.13E-01	3.17E-01	4.94E-05	2.10E-05	2.24E-02	2.52E-03	8.02E-03	3.89E-01	4.99E-02	1.46E-01	2.81E-06	1.58E-04
Exactly2	3.55E-06	2.63E-02	1.04E-08	3.55E-11	3.46E-03	9.25E-06	1.76E-03	3.29E-01	7.87E-07	1.37E-03	2.44E-14	2.91E-08
HeartEW	4.46E-03	3.96E-01	5.47E-06	4.25E-07	4.75E-03	2.59E-04	6.12E-02	2.82E-01	5.83E-05	4.72E-03	6.43E-11	2.88E-07
IonosphereEW	1.93E-02	3.76E-01	1.95E-06	2.86E-07	1.90E-03	4.12E-05	1.68E-04	3.64E-01	6.03E-06	2.81E-03	4.08E-09	1.59E-09
KrVsKpEW	1.89E-05	5.59E-04	8.78E-11	2.89E-07	2.57E-02	1.17E-06	1.47E-06	1.36E-02	5.50E-06	2.21E-05	4.62E-12	2.12E-13
Lymphography	2.07E-03	1.92E-01	5.70E-07	1.23E-08	1.16E-02	2.99E-05	2.51E-03	3.30E-01	1.52E-03	4.86E-03	5.32E-12	1.69E-10
M-of-n	9.39E-02	1.13E-04	2.06E-11	4.75E-10	2.14E-02	2.41E-06	3.14E-07	3.77E-03	3.69E-03	2.08E-04	5.84E-23	7.03E-14
PenglungEW	9.83E-03	1.52E-01	4.69E-03	2.45E-07	2.50E-02	4.67E-03	9.13E-03	3.96E-01	5.75E-04	8.86E-04	2.13E-03	3.82E-09
SonarEW	8.03E-06	3.80E-02	9.18E-07	6.67E-09	2.50E-04	3.45E-06	1.72E-04	9.11E-02	6.59E-07	5.58E-05	5.82E-13	8.74E-13
SpectEW	2.99E-02	1.63E-07	2.76E-10	5.84E-12	1.72E-05	1.21E-04	1.58E-07	1.16E-02	2.58E-05	1.56E-07	2.99E-14	1.01E-12
Tic-tac-toe	1.75E-01	N/A	3.14E-04	7.00E-05	2.74E-01	7.93E-02	8.96E-02	N/A	9.41E-03	3.16E-02	3.23E-13	1.93E-07
Vote	1.33E-02	2.14E-07	1.71E-03	2.29E-06	7.18E-03	1.39E-06	5.44E-08	5.78E-02	4.06E-04	1.39E-06	2.89E-12	5.16E-06
WaveformEW	4.56E-03	7.79E-06	2.04E-12	1.15E-14	9.52E-06	1.08E-08	2.51E-10	2.11E-03	4.48E-08	1.61E-05	1.22E-17	1.31E-12
WineEW	9.54E-02	8.07E-03	8.20E-04	2.25E-06	3.99E-02	8.20E-04	1.87E-02	1.61E-01	8.20E-04	1.38E-01	2.53E-10	4.54E-13
Zoo	8.07E-03	1.95E-05	3.53E-09	2.17E-12	1.94E-05	1.54E-06	1.22E-07	6.50E-03	1.94E-05	1.22E-07	6.89E-02	9.98E-14
Ranking (W T L)	13 0 5	9 1 9	18 0 0	18 0 0	15 0 3	16 0 2	15 0 3	6 1 12	18 0 0	16 0 2	17 0 1	18 0 0

From the p -values in Table 25, it can be seen that based on the mean fitness values, iBSSA with SVM have a statistically significant difference compared to BBA and BASO over the 18 datasets. In addition, the iBSSA with SVM is still more significant than BGOA, BHGSO, BPSO, BBSA, BABC, BWOA, BHHO, BSSA, and BGWO respectively over 17, 17, 16, 16, 15, 15, 14, 12, and 12 out of the 18 datasets. On the other hand, iBSSA, BSFO, and BHHO exhibit the same performance with SVM for the dataset Exactly2 as the mean fitness value is 0.2483 and the standard deviation is zero. The same observation is for

iBSSA, BABC, BGWO, and BSFO, where the same performance is also exhibited with SVM for the dataset Tic-tac-toe as the mean fitness value is 0.1017 and the standard deviation is zero. Moreover, it is very notable that iBSSA with SVM do not have a statistically significant difference from BSSA, BPSO, BGWO, BWOA, and BHGSO on the dataset M-of-n. Furthermore, iBSSA with SVM does not have a statistically significant difference from BWOA, BGOA, BHHO, and BBSA on the dataset Tic-tac-toe.

Table 26 shows the p -values of iBSSA with RF compared to other algorithms. It is remarkable that iBSSA has

Table 27 Comparison of iBSSA and other algorithms from the specialized literature in terms of the mean classification accuracy (μ_{Acc})

Benchmark	iBSSA- <i>k</i> -NN	iBSSA-SVM	iBSSA-RF	BGOA-EPD [72]	TCSSA3 [6]	HGSA [107]	BSSA-CP [35]	GA [32]	SA [42]	WOASAT-2 [74]
BreastCancer	0.9857	0.9786	0.9857	0.9800	N/A	0.9740	0.9768	0.9680	0.9699	0.9700
BreastEW	0.9649	0.9474	0.9944	0.9470	N/A	0.9710	0.9484	0.9390	N/A	0.9800
CongressEW	0.9770	0.9770	0.9808	0.9640	0.9705	0.9660	0.9628	0.9320	N/A	0.9800
Exactly	0.9973	0.7450	0.7633	0.9990	0.9969	1.0000	0.9803	0.6740	N/A	1.0000
Exactly2	0.7737	0.7500	0.7632	0.7800	0.7672	0.7700	0.7582	0.7460	N/A	0.7500
HeartEW	0.9074	0.9074	0.9302	0.8330	0.8331	0.8560	0.8605	0.7800	N/A	0.8500
IonosphereEW	0.9385	0.9681	0.9690	0.8990	0.9377	0.9340	0.9182	0.8140	0.9205	0.9600
KrVsKpEW	0.9852	0.9854	0.9487	0.9680	0.9692	0.9780	0.9644	0.9200	N/A	0.9800
Lymphography	0.8378	0.8500	0.8800	0.8680	0.8444	0.8920	0.8900	0.6960	N/A	0.8900
M-of-n	1.0000	1.0000	0.9993	1.0000	0.9992	1.0000	0.9918	0.8610	N/A	1.0000
PenglungEW	0.6511	0.8267	0.7489	0.9270	0.9072	0.9560	0.8775	0.5840	N/A	0.9400
SonarEW	0.9913	0.9476	0.9206	0.9120	0.9481	0.9580	0.9372	0.7540	0.4925	0.9700
SpectEW	0.8889	0.8364	0.8815	0.8260	0.8333	0.9190	0.8361	0.7930	N/A	0.8800
Tic-tac-toe	0.8542	0.9062	0.8698	0.8080	0.7975	0.7880	0.8205	0.7190	N/A	0.7900
Vote	1.0000	1.0000	1.0000	0.9660	0.9549	0.9730	0.9511	0.9040	N/A	0.9700
WaveformEW	0.8484	0.8791	0.8201	0.7370	0.7364	0.8150	0.7335	0.7730	N/A	0.7600
WineEW	1.0000	1.0000	1.0000	0.9890	0.9978	0.9890	0.9933	0.9370	0.9787	0.9900
Zoo	1.0000	1.0000	1.0000	0.9930	0.9928	0.9320	1.0000	0.8550	0.6829	0.9700
Ranking (W T L)	115 12	314 11	4 4 10	111 16	0 0 16	4 11 3	0 11 7	0 0 18	0 0 5	0 2 16

statistically significant differences from other algorithms over the vast majority of datasets, especially in the cases of BPSO, BBA, BHHO, and BHGSO, where a statistically outstanding difference is exhibited over the 18 datasets.

5.10 Comparison with various FS methods from the literature

This subsection compares the mean classification accuracy results for the three models proposed in this article (i.e., iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF) versus various algorithms from the literature, including BGOA-EPD_Tour [72], TCSSA3 [6], HGSA [107], BSSA_S3_CP [35], GA [32], SA [42], and WOASAT-2 [74], which were conducted under the same experimental settings. Table 27 reports the experimental data of the mean classification accuracy of iBSSA-*k*-NN, iBSSA-SVM, iBSSA-RF, and various methods in the literature.

By inspecting Table 27, it can be seen that in 18 datasets, iBSSA-RF proposed in this article dominated all approaches in terms of the mean classification accuracy on 4 datasets. Moreover, iBSSA-SVM and iBSSA-*k*-NN take the second and third places with, respectively, 3 and 1 datasets, followed by HGSA, BGOA-EPD_Tour, WOA-SAT-2, BSSA_S3_CP, GA, and TCSSA3. In addition, by

comparing specific data, it is important to highlight that the mean classification accuracy of iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF have great advantages over competitors. Specifically, on the datasets M-of-n, Vote, WineEW, and Zoo, it reaches up to 100%, and those datasets are of low dimensionality, demonstrating the high performance of iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF with low-dimensional datasets. Although HGSA outperformed iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF on 4 datasets, the numerical margin was relatively small. Therefore, the iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF proposed in this work performed better in solving the FS problem than other FS methods in the literature, ultimately taking the top three in the competition.

5.11 Discussions

This study fused an improved binary version of the SSA algorithm and three well-known classifiers into three models (i.e., iBSSA-*k*-NN, iBSSA-SVM, and iBSSA-RF) for FS in the classification task. Based on the empirical analytics performed above in this section, it can be seen that the proposed iBSSA algorithm with the three classifiers typically performs better than other state-of-the-art and well-known algorithms. Specifically, iBSSA with RF

efficiently achieved higher mean accuracy over the majority of datasets, followed by iBSSA with SVM and iBSSA with k -NN. Moreover, the iBSSA algorithm performed the search more stably in comparison with peers, based on the small standard deviation values in the results. This subsection analyzes the underlying reasons.

First, in an attempt to resolve the feature selection problem more effectively, the original SSA algorithm in its continuous version was first mapped into binary using a transfer function. In order to make this approach as much fruitful as possible, the present study compared nine S-shaped and V-shaped transfer functions. A probability is obtained using the transfer function to map the real value into a binary one. Moreover, three well-known classifiers: k -Nearest Neighbor (k -NN), Support Vector Machine (SVM), and Random Forest (RF) are utilized as fitness evaluators both with the proposed iBSSA and many other competitor algorithms, using 18 multifaceted, multi-scale benchmark datasets from UCI repository. Through experimental comparisons, iBSSA- V_{v_2} -RF performed the best in iBSSA with the transfer function V_{v_2} and the RF classifier. In this work, nine transfer functions were compared because different transfer functions have different curve slopes. When the transfer function generates the probability, the probability of numerical mapping is made more reasonable through the better slope, so that the algorithm can perform better.

Second, based on the above introduction, the SSA algorithm can be understood as a novel swarm-based meta-heuristic algorithm. It has strong exploration and exploitation capabilities which were further boosted via the proposed iBSSA method by incorporating a strategy for random re-positioning of roaming agents as well as a novel local search algorithm. Moreover, the variable parameters and many mechanisms in the algorithm balance its exploration and development. Herein, the iBSSA only maps real values into binary ones as well as has two improvements (3RA and LSA) introduced into its original structure. Thus, the proposed iBSSA algorithm still has a superior performance.

Moreover, apart from the minimized feature subset and the better mean accuracy, iBSSA with k -NN has exhibited better convergence behavior than other optimizers on most of the datasets. Although iBSSA has, in general, achieved better performance compared to other methods, including the original BSSA algorithm, the convergence graphs reveal that iBSSA still has a problem with high-dimensional datasets, specifically trapping into local optima. Additionally, the mean accuracy of iBSSA with k -NN is inferior to other approaches on high-dimensional datasets. In the future, these shortcomings will be deeply examined. Apart from iBSSA, k -NN was used in this study as a classification algorithm due to its ease of implementation.

However, k -NN performance is often degraded when compared to SVM and RF in terms of mean accuracy and mean number of selected features. This can be justified by the adopted wrapper-based feature selection approach, wherein iBSSA works with k -NN as a lazy classifier and with SVM and RF as complex classifiers. Lastly, it should be noted that, because of the non-exact repeatability of the optimization results, different executions of the algorithm can give a different subset of features, which may confuse the user. Hence, on different applications or occasions, iBSSA or other meta-heuristic algorithms applied herein may find a different subset of features.

Summarizing, a three fine-tuned classification algorithms are presented in this study with respect to their algorithmic design perspective, as well as the general framework. First, we briefly discussed the conventional SSA algorithm, different binary conversion techniques, and proposed learning algorithms. Then, proposed iBSSA was duly presented and a theoretical study was conducted on FS approaches. We also presented an adequate analysis of some of the state-of-the-art FS approaches in a comparative manner. Furthermore, a comparison of existing FS algorithms was conducted based on standard performance evaluation metrics. Finally, based on the FS problem formulation, research challenges were discussed with some future directions. This study could be a good starting point for researchers new to the FS community by providing them with a better understanding of recent developments on the FS problem. The current experimental study on FS identifies several research possibilities in data classification. However, several pieces had been done in FS, but still many more things to be uncovered. Future works in FS should boost increasing in efficiency, robustness, scalability, and effectiveness in large-scale datasets. Along with scalability, one of the more challenging issues is the context in FS, which must assert in the future.

6 Conclusions and future directions

A novel approach was proposed in this study for solving the FS problem using an improved binary version of the SSA algorithm (iBSSA). First, a suitable method for binary transformation of continuous SSA to the binary version was determined through the form of probability using common S-shaped and V-shaped transfer functions. In addition, a strategy for Random Re-positioning of Roaming Agents (3RA) along with a novel Local Search Algorithm (LSA) were embedded into the proposed iBSSA algorithm, in order to increase the algorithm capability to search within feasible regions as well as to prevent it from becoming trapped into local optima, respectively. Then, the resulting methods were tested on 18 multifaceted, multi-

scale benchmark UCI datasets with three well-known classifiers: k -NN, SVM, and RF. The experimental results showed that iBSSA- V_{v4} - k -NN, iBSSA- V_{v2} -SVM, and iBSSA- V_{v2} -RF were the best-performing methods in the proposed algorithms. Then, each of these methods was compared with the better performing and most popular methods in the literature, independently based on every classifier. Comparisons were driven by a number of measures, including mean classification accuracy, mean fitness, and mean number of selected features. By analyzing the experimental results, the three models proposed in this article (i.e., iBSSA- k -NN, iBSSA-SVM, and iBSSA-RF) relatively had the highest performance among the competing algorithms for solving FS problems. Specifically, iBSSA with RF (iBSSA-RF) performed better than other methods. Therefore, iBSSA-RF can be considered more highly when solving FS problems.

Moreover, the proposed iBSSA wrapper feature selection model has an important practical implication. Based on the various types of datasets used over the conducted experiments, iBSSA can be introduced into other domains, including medical applications, engineering applications, data mining, data science, and many more. However, the proposed iBSSA has one limitation: selecting more features than those selected by competitors over 6, 5, and 11 out of the 18 datasets based on iBSSA- k -NN, iBSSA-SVM, and iBSSA-RF, respectively. Therefore, the proposed algorithm can be reinforced by using a new selection strategy to select fewer features, especially on high-dimensional small instance datasets, which calls for further potential future research.

For future work, different conversion forms of iBSSA can be attempted for classification purposes. In addition, a binary version was successfully yielded from the continuous SSA, which can help tackle other discrete optimization problems; e.g., task scheduling, Travelling Salesman Problem (TSP) problem, Knapsack Problem (KP), etc., as well as investigate real-world problems, including Intrusion Detection System (IDS), cancer detection, and sentiment analysis. On the other hand, a sensitivity analysis of the main parameters of the proposed algorithm can be explored as a future extension of this work. Moreover, the embedding of the proposed LSA algorithm with other optimization algorithms could also be investigated. Lastly, other classifiers (e.g., Artificial Neural Networks (ANNs), Decision Tree (DT), etc.) could be employed to further investigate the performance of iBSSA in feature selection for classification.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Declarations

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abdel-Basset M, El-Shahat D, El-henawy I, de Albuquerque VHC, Mirjalili S (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst Appl* 139:112824
2. Agarwal P, Ahsan S, Akbar M, Nawaz R, Cesarano C (2021a) A reliable algorithm for solution of higher dimensional nonlinear $(1+ 1)$ and $(2+ 1)$ dimensional volterra-fredholm integral equations. *Dolomites Research Notes on Approximation* 14(2)
3. Agarwal P, Filali D, Akram M, Dilshad M (2021) Convergence analysis of a three-step iterative algorithm for generalized set-valued mixed-ordered variational inclusion problem. *Symmetry* 13(3):444
4. Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW (2021) Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019). *IEEE Access* 9:26766–26791
5. Aha DW, Bankert RL (1996) A comparative evaluation of sequential feature selection algorithms. In: *Learning from data*, Springer, pp 199–206
6. 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
7. Alsouda Y, Pllana S, Kurti A (2019) Iot-based urban noise identification using machine learning: Performance of svm, knn, bagging, and random forest. In: *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, pp. 62–67
8. Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. *Expert Syst Appl* 116:147–160
9. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23(3):715–734
10. Backer E (1977) On the max-min approach for feature ordering and selection. In: *The Seminar on Pattern Recognition*, Liege University, Sart-Tilmap
11. Baig MZ, Aslam N, Shum HP, Zhang L (2017) Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery eeg. *Expert Syst Appl* 90:184–195
12. Barnard CJ, Sibly RM (1981) Producers and scroungers: a general model and its application to captive flocks of house sparrows. *Anim Behav* 29(2):543–550
13. Barta Z, Liker A, Mónus F (2004) The effects of predation risk on the use of social foraging tactics. *Anim Behav* 67(2):301–308
14. Bouktif S, Fiaz A, Ouni A, Serhani MA (2018) Optimal deep learning lstm model for electric load forecasting using feature

- selection and genetic algorithm: comparison with machine learning approaches. *Energies* 11(7):1636
15. Budgey R (1998) The three dimensional structure of bird flocks and its implications for birdstrike tolerance in aircraft. *Int Bird Strike Committee Proc* 24:207–220
 16. Cao H, Bernard S, Sabourin R, Heutte L (2019) Random forest dissimilarity based multi-view learning for radiomics application. *Pattern Recogn* 88:185–197
 17. Chen K, Zhou FY, Yuan XF (2019) Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection. *Expert Syst Appl* 128:140–156
 18. Cover T, Hart P (1967) Nearest neighbor pattern classification. *IEEE Trans Inf Theory* 13(1):21–27
 19. Cover TM, Van Campenhout JM (1977) On the possible orderings in the measurement selection problem. *IEEE Trans Syst Man Cybern* 7(9):657–661
 20. Criminisi A, Shotton J, Konukoglu E (2012) Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision* 7(2–3): 81–227
 21. Das AK, Das S, Ghosh A (2017) Ensemble feature selection using bi-objective genetic algorithm. *Knowl-Based Syst* 123:116–127
 22. Dash M, Liu H (1997) Feature selection for classification. *Intell Data Anal* 1(1–4):131–156
 23. De León-Aldaco SE, Calleja H, Alquicira JA (2015) Metaheuristic optimization methods applied to power converters: a review. *IEEE Trans Power Electron* 30(12):6791–6803
 24. De Souza RCT, dos Santos Coelho L, De Macedo CA, Pierezan J (2018) A v-shaped binary crow search algorithm for feature selection. In: 2018 IEEE congress on evolutionary computation (CEC), IEEE, pp. 1–8
 25. Demidova L, Klyueva I, Pylkin A (2019) Hybrid approach to improving the results of the svm classification using the random forest algorithm. *Procedia Comput Sci* 150:455–461
 26. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
 27. Devijver PA, Kittler J (1982) *Pattern recognition: a statistical approach*. Prentice hall, Hoboken
 28. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. *Comput Ind Eng* 137:106040
 29. Dong H, Li T, Ding R, Sun J (2018) A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Appl Soft Comput* 65:33–46
 30. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Ieee, pp. 39–43
 31. Eid HF (2018) Binary whale optimisation: an effective swarm algorithm for feature selection. *Int J Metaheuristics* 7(1):67–79
 32. Emary E, Zawbaa HM, Hassanien AE (2016) Binary ant lion approaches for feature selection. *Neurocomputing* 213:54–65
 33. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. *Neurocomputing* 172:371–381
 34. Faris H, Hassonah MA, Ala'M AZ, Mirjalili S, Aljarah I (2018) A multi-verse optimizer approach for feature selection and optimizing svm parameters based on a robust system architecture. *Neural Comput Appl* 30(8):2355–2369
 35. Faris H, Mafarja MM, Heidari AA, Aljarah I, Ala'M AZ, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl-Based Syst* 154:43–67
 36. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. *AI Mag* 17(3):37–37
 37. Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (1996b) *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence
 38. Frank A (2010) Uci machine learning repository. <https://archive.ics.uci.edu/ml>
 39. Fukunaga K, Koontz WL (1970) Application of the karhunen-loeve expansion to feature selection and ordering. *IEEE Trans Comput* 100(4):311–318
 40. Gao Y, Zhou Y, Luo Q (2020) An efficient binary equilibrium optimizer algorithm for feature selection. *IEEE Access* 8:140936–140963
 41. Gu S, Cheng R, Jin Y (2018) Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Comput* 22(3):811–822
 42. Guha R, Ghosh M, Kapri S, Shaw S, Mutsuddi S, Bhateja V, Sarkar R (2021) Deluge based genetic algorithm for feature selection. *Evol Intel* 14(2):357–367
 43. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. *J Mach Learn Res* 3:1157–1182
 44. Hafez AI, Zawbaa HM, Emary E, Hassanien AE (2016) Sine cosine optimization algorithm for feature selection. In: 2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), IEEE, pp 1–5
 45. Hamilton WD (1971) Geometry for the selfish herd. *J Theor Biol* 31(2):295–311
 46. Han J, Pei J, Kamber M (2011) *Data mining: concepts and techniques*. Elsevier, Amsterdam
 47. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. *Futur Gener Comput Syst* 101:646–667
 48. Hegazy AE, Makhlof M, El-Tawel GS (2020) Improved salp swarm algorithm for feature selection. *J King Saud Univ-Comput Inf Sci* 32(3):335–344
 49. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872
 50. Houssein EH, Gad AG, Hussain K, Suganthan PN (2021) Major advances in particle swarm optimization: theory, analysis, and application. *Swarm Evol Comput* 63:100868
 51. Houssein EH, Gad AG, Wazery YM (2021b) Jaya algorithm and applications: a comprehensive review. *Metaheuristics Optim Comput Electr Eng*, pp. 3–24
 52. Houssein EH, Gad AG, Wazery YM, Suganthan PN (2021c) Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends. *Swarm and Evolutionary Computation* p 100841
 53. Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. *IEEE Trans Pattern Anal Mach Intell* 19(2):153–158
 54. Jain AK, Vailaya A (1996) Image retrieval using color and shape. *Pattern Recogn* 29(8):1233–1244
 55. Jiang S, Chin KS, Wang L, Qu G, Tsui KL (2017) Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. *Expert Syst Appl* 82:216–230
 56. Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G (2015) A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. *Oper Res Perspect* 2:62–72
 57. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132

58. Kashef S, Nezamabadi-pour H (2015) An advanced aco algorithm for feature subset selection. *Neurocomputing* 147:271–279
59. Katuwal R, Suganthan PN, Zhang L (2018) An ensemble of decision trees with random vector functional link networks for multi-class classification. *Appl Soft Comput* 70:1146–1153
60. 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
61. Kira K, Rendell LA (1992) A practical approach to feature selection. In: *Machine learning proceedings 1992*, Elsevier, pp 249–256
62. Kittler J (1978) Feature set search algorithms. *Pattern Recogn Signal Process*
63. Kohavi R (1996) Wrappers for performance enhancement and oblivious decision graphs. PhD thesis, stanford university
64. Kohavi R, John GH (1997) Wrappers for feature subset selection. *Artif Intell* 97(1–2):273–324
65. Lanzi PL (1997) Fast feature selection with genetic algorithms: a filter approach. In: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, IEEE, pp 537–540
66. Lendvai AZ, Barta Z, Liker A, Bókony V (2004) The effect of energy reserves on social foraging: hungry sparrows scrounge more. *Proc R Soc Lond B* 271(1556):2467–2472
67. Li T, Dong H, Sun J (2019) Binary differential evolution based on individual entropy for feature subset optimization. *IEEE Access* 7:24109–24121
68. Liker A, Barta Z (2002) The effects of dominance on social foraging tactic use in house sparrows. *Behaviour* 139(8):1061–1076
69. Liu H, Motoda H (2012) *Feature selection for knowledge discovery and data mining*, vol 454. Springer Science & Business Media, Berlin
70. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. *Appl Soft Comput* 62:441–453
71. Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li X, Mirjalili S (2018) Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl-Based Syst* 161:185–204
72. Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Ala'M AZ, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowl-Based Syst* 145:25–45
73. 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
74. Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* 260:302–312
75. Marill T, Green D (1963) On the effectiveness of receptors in recognition systems. *IEEE Trans Inf Theory* 9(1):11–17
76. Meng XB, Gao XZ, Lu L, Liu Y, Zhang H (2016) A new bio-inspired optimisation algorithm: Bird swarm algorithm. *J Exp Theor Artif Intell* 28(4):673–687
77. Mirjalili S, Lewis A (2013) S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm Evol Comput* 9:1–14
78. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
79. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
80. Murugan A, Nair SAH, Kumar KS (2019) Detection of skin cancer using svm, random forest and knn classifiers. *J Med Syst* 43(8):269
81. Narendra PM, Fukunaga K (1977) A branch and bound algorithm for feature subset selection. *IEEE Trans Comput* 26(09):917–922
82. Neggaz N, Houssein EH, Hussain K (2020) An efficient henry gas solubility optimization for feature selection. *Expert Syst Appl*, p 113364
83. Obimbo C, Nyakundi E (2017) Comparison of svms with radial-basis function and polynomial kernels in classification of categories in intrusion detection. In: *Proceedings of the International Conference on Security and Management (SAM)*, The Steering Committee of The World Congress in Computer Science, Computer, pp 240–246
84. Ojha VK, Abraham A, Snašel V (2017) Metaheuristic design of feedforward neural networks: a review of two decades of research. *Eng Appl Artif Intell* 60:97–116
85. Paul S, Das S (2015) Simultaneous feature selection and weighting-an evolutionary multi-objective optimization approach. *Pattern Recogn Lett* 65:51–59
86. Pudil P, Ferri FJ, Novovicova J, Kittler J (1994a) Floating search methods for feature selection with nonmonotonic criterion functions. In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5), IEEE, vol 2, pp 279–283
87. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. *Pattern Recogn Lett* 15(11):1119–1125
88. Pulliam HR (1973) On the advantages of flocking. *J Theor Biol* 38(2):419–422
89. Punch III WF, Goodman ED, Pei M, Chia-Shun L, Hovland PD, Enbody RJ (1993) Further research on feature selection and classification using genetic algorithms. In: *ICGA*, pp 557–564
90. Raman MG, Somu N, Kirthivasan K, Liscano R, Sriram VS (2017) An efficient intrusion detection system based on hyper-graph-genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowl-Based Syst* 134:1–12
91. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
92. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2010) Bgsa: binary gravitational search algorithm. *Nat Comput* 9(3):727–745
93. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
94. Sayed GI, Darwish A, Hassanien AE (2018) A new chaotic whale optimization algorithm for features selection. *J Classif* 35(2):300–344
95. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl Intell* 48(10):3462–3481
96. Sayed SAF, Nabil E, Badr A (2016) A binary clonal flower pollination algorithm for feature selection. *Pattern Recogn Lett* 77:21–27
97. Schael M (2001) Texture defect detection using invariant textural features. In: *Joint Pattern Recognition Symposium*, Springer, pp. 17–24
98. Shadravan S, Naji H, Bardsiri VK (2019) The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Eng Appl Artif Intell* 80:20–34
99. Sharawi M, Zawbaa HM, Emary E (2017) Feature selection approach based on whale optimization algorithm. In: *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, IEEE, pp. 163–168
100. Sharma M, Kaur P (2020) A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem. *Archives Comput Methods Eng*, pp. 1–25

101. Sharma M, Kaur P (2021) A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem. *Archiv Comput Methods Eng*, 28(3)
102. Sharma V, Baruah D, Chutia D, Raju P, Bhattacharya D (2016) An assessment of support vector machine kernel parameters using remotely sensed satellite data. In: 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, pp. 1567–1570
103. Siedlecki W, Sklansky J (1993) A note on genetic algorithms for large-scale feature selection. In: *Handbook of pattern recognition and computer vision*, World Scientific, pp 88–107
104. Streams S (1976) On selecting features for pattern classifiers. *Proc ICPR 1976*:71–75
105. Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. *Eng Appl Artif Intell* 32:112–123
106. Tang J, Alelyani S, Liu H (2014) Feature selection for classification: A review. *Data classification: algorithms and applications* p 37
107. 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
108. Thaher T, Heidari AA, Mafarja M, Dong JS, Mirjalili S (2020) Binary harris hawks optimizer for high-dimensional, low sample size feature selection. In: *Evolutionary Machine Learning Techniques*, Springer, pp. 251–272
109. Tharwat A, Hassanien AE, Elnaghi BE (2017) A ba-based algorithm for parameter optimization of support vector machine. *Pattern Recogn Lett* 93:13–22
110. Tsai CW, Tsai PW, Pan JS, Chao HC (2015) Metaheuristics for the deployment problem of wsn: a review. *Microprocess Microsyst* 39(8):1305–1317
111. Tubishat M, Idris N, Shuib L, Abushariah MA, Mirjalili S (2020) Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst Appl* 145:113122
112. Van Laarhoven PJ, Aarts EH (1987) *Simulated annealing*. In: *Simulated annealing: Theory and applications*, Springer, pp. 7–15
113. Vapnik V (2013) *The nature of statistical learning theory*. Springer science & business media, Berlin
114. Wang H, Jing X, Niu B (2017) A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data. *Knowl-Based Syst* 126:8–19
115. Whitney AW (1971) A direct method of nonparametric measurement selection. *IEEE Trans Comput* 100(9):1100–1103
116. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
117. Wong W, Ming CI (2019) A review on metaheuristic algorithms: recent trends, benchmarking and applications. In: 2019 7th International Conference on Smart Computing and Communications (ICSCC), IEEE, pp. 1–5
118. Wu Y, Ianakiev K, Govindaraju V (2002) Improved k-nearest neighbor classification. *Pattern Recogn* 35(10):2311–2318
119. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. *Syst Sci Control Eng* 8(1):22–34
120. Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. In: *Feature extraction, construction and selection*, Springer, pp 117–136
121. Yang XS (2009) Firefly algorithms for multimodal optimization. In: *International symposium on stochastic algorithms*, Springer, pp. 169–178
122. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, pp. 65–74
123. Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World congress on nature and biologically inspired computing (NaBIC), IEEE, pp. 210–214
124. Yusta SC (2009) Different metaheuristic strategies to solve the feature selection problem. *Pattern Recogn Lett* 30(5):525–534
125. Zakeri A, Hokmabadi A (2019) Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Syst Appl* 119:61–72
126. Zaki MJ, Meira W (2014) *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, Cambridge
127. Zawbaa HM, Hassanien AE, Emary E, Yamany W, Parv B (2015) Hybrid flower pollination algorithm with rough sets for feature selection. In: 2015 11th International Computer Engineering Conference (ICENCO), IEEE, pp 278–283
128. Zhang H, Sun G (2002) Feature selection using tabu search method. *Pattern Recogn* 35(3):701–711
129. Zhang L, Mistry K, Lim CP, Neoh SC (2018) Feature selection using firefly optimization for classification and regression models. *Decis Support Syst* 106:64–85
130. Zhang Y, Agarwal P, Bhatnagar V, Balochian S, Yan J (2013) *Swarm intelligence and its applications*
131. Zhang Y, Agarwal P, Bhatnagar V, Balochian S, Zhang X (2014) *Swarm intelligence and its applications 2014*
132. Zhang Y, Liu R, Wang X, Chen H, Li C (2021) Boosted binary harris hawks optimizer and feature selection. *Eng Comput* 37(4):3741–3770
133. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl-Based Syst* 163:283–304

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ahmed G. Gad¹  · Karam M. Sallam^{2,3}  · Ripon K. Chakraborty⁴  · Michael J. Ryan⁵  · Amr A. Abohany¹ 

✉ Ahmed G. Gad
ahmed.gad@fci.kfs.edu.eg;
https://ahmedgad.com

Karam M. Sallam
karam_sallam@zu.edu.eg; karam.sallam@canberra.edu.au

Ripon K. Chakraborty
r.chakraborty@adfa.edu.au

Michael J. Ryan
mike.ryan@ieee.org

Amr A. Abohany
amrabohany8@gmail.com;
<https://www.amrabohany.software>

- ¹ Faculty of Computers and Information, Kafrelsheikh University, Kafrelsheikh, Egypt
- ² Faculty of Computers and Information, Zagazig University, Zagazig, Egypt

- ³ Faculty of Science and Technology, School of IT and Systems, University of Canberra, Canberra, ACT 2601, Australia
- ⁴ Capability Systems Centre, School of Engineering and IT, UNSW Canberra at ADFA, Canberra, Australia
- ⁵ Capability Associates Canberra, Canberra, Australia