Applied Intelligence (2019) 49:1580–1596
https://doi.org/10.1007/s10489-018-1348-2

# Hybridization of feature selection and feature weighting for high dimensional data

Dalwinder Singh[1] · Birmohan Singh[1]

Published online: 20 November 2018

## Abstract

The classification of high dimensional data is a challenging problem due to the presence of redundant and irrelevant features in a higher amount. These unwanted features degrade accuracy and increase the computational complexity of machine learning algorithms. In this paper, we propose a hybrid method that integrates the complementary strengths of feature selection and feature weighting approaches for improving the classification of high dimensional data on the Nearest Neighbor classifier. Specifically, we suggest four strategies that combine filter and wrapper methods of feature selection and feature weighting. Experiments are performed on 12 high dimensional datasets and outcomes are supported by Friedman as well as Holm statistical tests for validation. Extended Adjusted Ratio of Ratios is used to recognize the best method considering accuracy, feature selection, and runtime. The results show that two proposed strategies outperform other well-known methods in accuracy and features reduction. The hybrid feature selection-feature weighting wrapper method is the best among all in accuracy while the hybrid feature selection filter-feature weighting wrapper method is the most suitable for reducing features and runtime. Thus, the promising outcomes validate the importance of hybridizing feature selection and feature weighting while dealing with high dimensional data.

**Keywords** Feature selection · Feature weighting · Hybrid method · Optimization algorithm

## 1 Introduction

Improvement in the learning ability of machine learning algorithms is still a challenge in the field of pattern recognition. The problem becomes even more complex when data contains a large number of features. This curse of dimensionality undermines the learning ability of a classifier, especially when instances are less than the features [1]. The presence of irrelevant and redundant features in the data confuses the learning algorithms and results in degraded performance of the classifier (also known as the Hughes phenomenon [2]). The other implications are over-fitting of the classifier and learning overheads because of higher computations [3, 4]. The problem has become even more challenging with the boom in stored and streaming data of classification during recent times. This tremendous increase in the data demands a more effective approach to tackle performance issues in machine learning algorithms. Therefore, many researchers are working on this active problem, and many efforts have been made [5–8] with the eventual goal of developing a well generalized and efficient machine learning algorithms from the high dimensional offline or online data.

Feature Selection (FS) and Feature Weighting (FW) are two widely adopted approaches for improving performance and reducing the dimensions of data. Feature selection is a combinatorial search problem where the feature is either accepted or rejected. It is suitable for the data containing redundant and irrelevant features only [9]. While feature weighting is a continuous search problem where weights are assigned to features according to their relevance [10, 11]. FW approaches are a good choice when relevancy of features vary in data [12].

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s10489-018-1348-2) contains supplementary material, which is available to authorized users.

✉ Birmohan Singh
birmohansingh@sliet.ac.in

Dalwinder Singh
dalwindercheema@outlook.com

[1] Department of Computer Science and Engineering,
Sant Longowal Institute of Engineering and Technology,
Longowal, Punjab, India

The methods for the selection of relevant feature subsets are of three types: filter, wrapper, and hybrid. Filter methods inspect intrinsic properties of the data to evaluate and select a subset of features. Wrapper methods use heuristic search algorithms where classifiers themselves act as an evaluator and, hybrid methods are based on a combination of the filter and wrapper methods. Feature weighting, on the other hand, is based mostly on wrapper methods [6, 11].

The filter methods suffer from lower classification accuracy whereas the performance of wrapper methods directly depend upon search ability of the optimization algorithm. However, these optimization algorithms also suffer from the curse of dimensionality. Therefore, replication of the same classification performance with the existing methods is still a difficult task when data has higher dimensions. Considering this as a motivation, we introduce the hybrid method that combines feature selection and feature weighting. Specifically, in this paper, four strategies based on the hybrid method are proposed for improving the classification performance on high dimensional data. In the paper, we investigate hybrid strategies for higher classification performance by combining filter and wrapper methods of feature selection and feature weighting approaches. Furthermore, the Ant Lion Optimization (ALO) with blend crossover for feature selection and feature weighting is also presented.

Rest of the paper is organized as: The related work is discussed in Section 2. The need for the hybrid method that combines feature selection and feature weighting approaches and proposed strategies are discussed in Section 3. The results and discussion of proposed method are provided in Section 4. The comparison of the work with other well-known methods, its limitations and future scope is also included in the section. Finally, Section 5 concludes the work.

## 2 Related work

A plethora of successful efforts have been made to improve classification performance and to select the best subset of features from lower dimensional data. Numerous filter, wrapper, and hybrid methods have been suggested in the literature. However, only a few have worked on high dimensional data, and their work is discussed as follows.

The ranking of features by assigning weights to them based on relevancy criterion, a RELIEF method [13]. Various variants have suggested for further advancement of this method which includes RELIEF-F [14], Iterative RELIEF [15], Fuzzy-theoretic Margin-maximization (FM-RELIEF) [16]. However, these algorithms are unable to identify features that are entirely redundant and get trapped in local optima while optimizing their objective function on higher dimensions. The same problem was observed for Simba algorithm [17] when data has large irrelevant features as pointed out by Sun et al. [18]. A Local leaning based feature selection method was proposed by Sun et al. [18] to find the best feature subsets from high dimensional data. Hall et al. [19] proposed correlation based feature selection which is based on the idea of searching a subset of features that are highly correlated within a class but are uncorrelated with the other class. Advancing the algorithm for higher dimensions, a fast correlation based filter (FCBF) was proposed by Yu and Liu [20] to select the subset of features by measuring the correlation between feature-class and feature-feature.

Recently, clustering based filter methods have gained a lot of attention due to their success in reducing features from high dimensional datasets. DeSarbo et al. [21] applied $k$-means Clustering to group features by computing weights assigned by the user. Huang et al. [22] automated the weight assignment in $k$-means clustering to group features. The weighted $k$-means clustering algorithm updates the weights of features based on the current partition at each iteration. Domeniconi et al. [23] proposed a Locally Adaptive Clustering (LAC) algorithm for finding a relevant subset of features. But, Liping et al. [24] pointed out that the objective function of LAC is not differentiable since it was a maximum function and they proved that replacing the largest average value with a constant value would lead to the same convergence of objective function. An entropy-based weighted $k$-means clustering (EWKM) was proposed by Liping et al. [24] to cluster the group of features. In their approach, weights for features were inversely proportional to the variance of the feature within-cluster. The work was further extended by Chen et al. [25] by introducing FG-$k$-means clustering algorithm where individual features, as well as feature groups, were weighted to find the best feature subset. Besides $k$-means clustering, recently Song et al. [26] used graph-theoretic clustering to find the best feature subset. They proposed a fast clustering-based feature selection (FAST) algorithm for the high dimensional datasets. It was a two-step approach in which clusters of features were formed using Minimum Spanning Tree (MST) based clustering method in the first step. In the next step, features strongly related to a class were selected from each cluster to estimate the final feature subset. Recently, Revanasiddappa and Harish [27] proposed Intuitionistic Fuzzy Entropy (IFE) method for selecting a subset of features from text categorization data. Initially, the Intuitionistic Fuzzy C-Means (IFCM) clustering method is used to measure the intuitionistic membership values for the features. Then, the intuitionistic fuzzy entropy is calculated with a matching degree using these membership values. The features that have low entropy are selected for the classification of text documents.

Wrapper methods have been studied extensively and have been used for feature selection, feature weighting, and simultaneous feature selection and feature weighting. In feature selection, Liu et al. [28] proposed the improved feature selection approach that combines both feature selection and kernel parameters simultaneously using modified Multi-Swarm Optimization. Ghamisi et al. [29] proposed a wrapper based approach for feature selection by integrating Genetic algorithm (GA) and Particle swarm optimization (PSO). Hancer et al. [30] used Binary Artificial Bee Colony (BABC) optimization algorithm whereas Hafez et al. [31] sine cosine algorithm (SCA) for selecting features.

In feature weighting, Kelly and Davis [10] used the combination of genetic algorithm and $k$-NN classifier for the improvement of the accuracy. Paredes et al. [32] applied the gradient descent optimization algorithm to weight the class and features (Class-dependent Weighting), individual Prototype (Prototype-dependent Weighting) and the combination of both. Additionally, they considered Euclidean and Class-Dependent Mahalanobis (CDM) distance schemes for measuring the distances in the $k$-NN classifier. AlSukker et al. [2] used Differential Evolution (DE) optimization algorithm for feature weighting, neighbors weighting, class weighting, and hybrid (feature and class) weighting. The studies for the individual feature weighting are very few, and it is combined with instances or feature selection.

Tahir et al. [33] proposed a simultaneous selection and weighting of the features using a Tabu search optimization algorithm. The scope for improvement in classification performance had motivated many researchers to work on optimizing feature subsets and feature weights simultaneously. Barros et al. [34] introduced a new adaptive distance scheme to further improve the performance of the work of Tahir et al. [33]. Derrac et al. [35] combined instance selection, instance weighting, and feature weighting using cooperative coevolution algorithm. Recently, Rodrguez et al. [6] studied 15 combinations of feature and instance selection and weighting. A framework using an evolutionary approach was presented that combines the selection and weighting of features as well as instances. The evolutionary approach used binary cross-generational elitist selection genetic algorithm for selection and differential evolution for weighting.

Feature selection based hybrid methods studied for high dimensional data includes the work of Chuang et al. [36] that combined correlation based feature selection and the Taguchi-Genetic algorithm for DNA microarray data. Derrac et al. [37] presented a hybrid method based on the combination of fuzzy rough set theory and Genetic Algorithm (GA) for instance and feature selection. Recently, Apolloni [38] proposed two hybrid methods that combine Information Gain (IG) and Binary Differential Evolution (BDE) for high-dimensional microarray data.

## 3 Hybridization of feature selection and feature weighting

### 3.1 The need for hybrid method

Feature selection and feature weighting are two well-proven approaches for improving classification accuracy and reducing data dimensionality. Feature selection is meant for the data that have redundant and irrelevant features only, whereas feature weighting suits for the data where features vary in relevance [12].

A classification problem has been shown in Fig. 1 from the viewpoint of feature selection and feature weighting. Two features, petal length, and petal width are used from Iris data to plot the classification accuracy of the 1-NN classifier. Figure 1a shows the accuracies for individual features as well as their combination. The data has many global optimum solutions as depicted in Fig. 1b. It can be seen that feature selection, being a subset of feature weighting, is a relatively less computational complex problem which gives a fair accuracy. On the other hand, feature weighting promises good accuracy, but searching for optimal solutions is a very time-consuming task. Filter methods can eliminate the irrelevant and redundant features based on the information measures of the data itself, but these methods are independent of the machine learning algorithms [39]. Moreover, these methods do not address the problem of parameters optimality of the classifier. Therefore, an improvement in the classification accuracy may not be guaranteed for every machine learning algorithm. When higher classification accuracy is the primary objective, wrapper methods are more effective [9] since these can address the data and parameters of classifiers simultaneously. However, these methods have two downsides; first is the selection of an optimization algorithm that converges towards a global solution avoiding local optimal solutions. Second is the execution time for evaluating the newly generated solutions. The execution cost depends upon total instances as well as dimensions of the input data. The data having large dimensions and instances requires more processing time and vice-versa. Therefore, wrapper methods are computationally intensive, and outcomes of performance entirely depend upon the exploration capability of the optimization method. Such approaches perform well for the data that have fewer features because it is easier to search for lower dimensions and requires less computational time. When data is high dimensional, the performance of wrapper

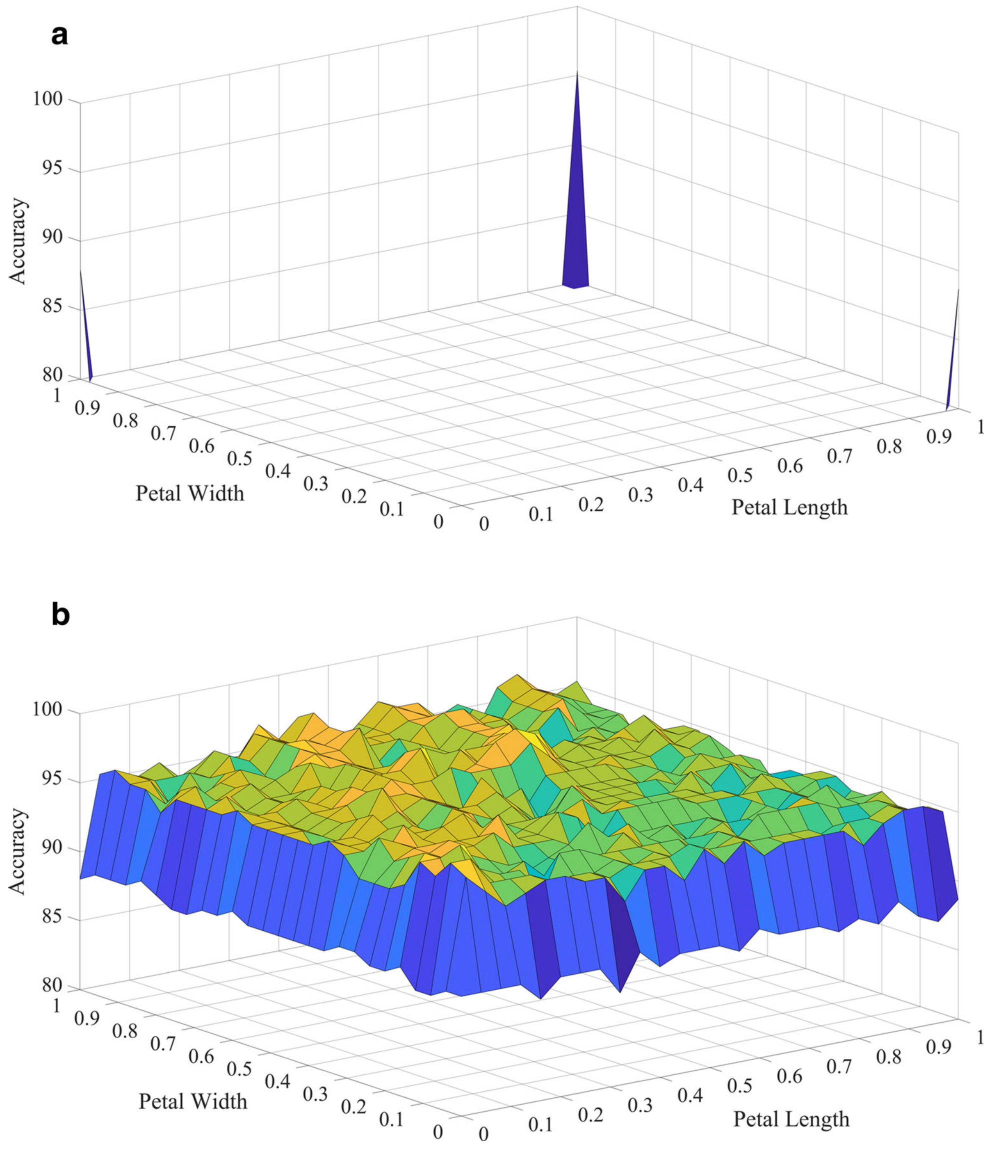


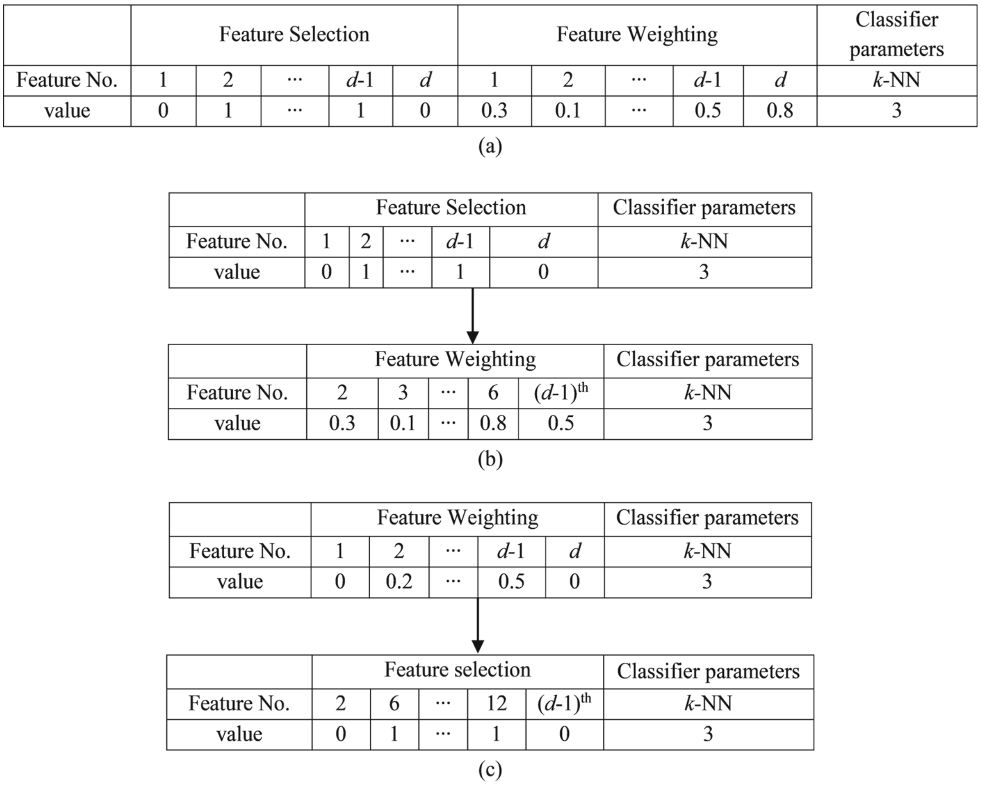
**Fig. 1** Visualization of feature selection and feature weighting problem considering two features of Iris data

methods deteriorates due to the implications of the curse of dimensionality for optimization algorithms [40]. Higher dimensions limit the searching capability of the optimization algorithms for a global solution due to an increase in search space and hence, stagnate at local optimal solutions.

The simultaneous feature selection and feature weighting methods show improvements in accuracy on lower dimensional data only, as compared to individual feature selection and feature weighting. But, the dimensions of the data are doubled when feature selection and feature weighting are integrated into a single problem. This two-fold increase in data dimensions make the problem more computationally complex to solve, and optimization algorithms fail to search for a global optimum solution thereby results in lower classification accuracy.

### 3.2 The Hybrid method combining feature selection and feature weighting

The limitation in previous studies motivates us to develop the hybrid method that explores the strengths of feature selection and feature weighting approaches. In this paper, we present four strategies that combine various methods of feature selection and feature weighting. We consider feature selection and feature weighting as an independent problem, which needs to be optimized independently instead of combining these into a single problem. The idea of hybridizing feature selection and feature weighting is illustrated in Fig. 2 with the help of an example. Figure 2a shows the wrapper based simultaneous search vector of feature selection and feature weighting having a size of ($2 * d+$ NN′s Parameter), where $d$ is the dimensions of



| | Feature Selection | | | | | Feature Weighting | | | | | Classifier parameters |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Feature No. | 1 | 2 | ··· | *d*-1 | *d* | 1 | 2 | ··· | *d*-1 | *d* | *k*-NN |
| value | 0 | 1 | ··· | 1 | 0 | 0.3 | 0.1 | ··· | 0.5 | 0.8 | 3 |

(a)

| | Feature Selection | | | | | Classifier parameters |
|---|---|---|---|---|---|---|
| Feature No. | 1 | 2 | ··· | *d*-1 | *d* | *k*-NN |
| value | 0 | 1 | ··· | 1 | 0 | 3 |

| | Feature Weighting | | | | | Classifier parameters |
|---|---|---|---|---|---|---|
| Feature No. | 2 | 3 | ··· | 6 | ($d$-1)th | *k*-NN |
| value | 0.3 | 0.1 | ··· | 0.8 | 0.5 | 3 |

(b)

| | Feature Weighting | | | | | Classifier parameters |
|---|---|---|---|---|---|---|
| Feature No. | 1 | 2 | ··· | *d*-1 | *d* | *k*-NN |
| value | 0 | 0.2 | ··· | 0.5 | 0 | 3 |

| | Feature selection | | | | | Classifier parameters |
|---|---|---|---|---|---|---|
| Feature No. | 2 | 6 | ··· | 12 | ($d$-1)th | *k*-NN |
| value | 0 | 1 | ··· | 1 | 0 | 3 |

(c)

**Fig. 2** An example of the hybridization of feature selection and feature weighting. **a** Simultaneous search vector of feature selection and feature weighting as given in existing work. **b** Feature selection followed by feature weighting and **c** feature weighting followed by feature selection

data. Figure 2b and c show the hybridization of feature selection and feature weighting in two possible ways along with how an individually optimized search vector might appear. The initial search vector consists of dimensions of the data along with the parameter of the NN classifier while the subsequent search vector has lower data dimensions in addition to the parameter of the classifier. The advantages of the proposed hybrid method are two-fold: reduction of dimensions through initial approach either by selecting a subset of features or searching for the weights which will help the successive approach to search for a better optimal solution from the resulting lower dimensional data. Second, different combinations of filter and wrapper methods can be tried to find the best performing hybrid method.

The feature selection and feature weighting approaches can be combined in two orderly ways. Figure 3 shows the possible combinations of feature selection and feature weighting using the filter and wrapper methods. In this paper, we suggest four hybrid strategies based on these two ways. The strategies are:

1. Feature selection with filter method followed by feature weighting with wrapper method (FFS-WFW)
2. Feature weighting with wrapper method followed by feature selection with filter method (WFW-FFS)
3. Feature selection followed by feature weighting with wrapper method (WFS-WFW)
4. Feature weighting followed by feature selection with wrapper method (WFW-WFS)

Although, many filter and wrapper methods have been presented in literature, but in this paper, we consider a clustering based FAST filtering method proposed by Song

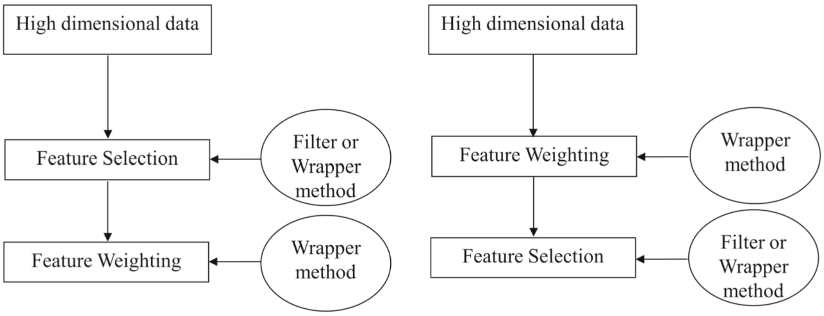


**Fig. 3** Proposed hybrid strategies for the high dimensional data

et al. [26], for selecting relevant features. The method identifies irrelevant features and is more successful for high dimensional datasets as compared to the other filter based methods. The wrapper method that we have used for feature selection as well as for feature weighting is the Ant Lion optimization algorithm with blend crossover. The crossover operation is used to improve exploration and exploitation capabilities for searching better solutions. The linear feature weighting method is employed in this work in which weights are multiplied with the features directly.

### 3.3 FAST clustering method

FAST method is a clustering-based feature selection (FAST) algorithm for high-dimensional data. It is a two-step procedure of feature selection which involves the elimination of irrelevant and redundant features. In the first step, the irrelevant features are discarded from the full set of features and then, in the second step, the redundant features are eliminated by selecting representative features from each cluster to a obtain the final subset of features. The method is based on the mutual information among feature or feature and target class where symmetric uncertainty $(SU)$ is used as a measure of correlation. It is given as follows:

$$S(X,Y)=\frac{2*IG(X|Y)}{H(X)+H(Y)} \tag{1}$$

where $S$ denotes the symmetric uncertainty, $H$ denotes the entropy and $IG$ denotes the information gain. The entropy measure for the variable $X$ is given as:

$$H(X)=-\sum_{x\in X}p(x)\log_2 p(x) \tag{2}$$

Similarly, the conditional entropy is measured for the variable $X$ after observing the values of another variable $Y$ as follows:

$$H(X|Y)=-\sum_{y\in Y}p(y)\sum_{x\in X}p(x|y)\log_2 p(x|y) \tag{3}$$

where $p(x)$ denotes the prior probabilities for all values of $X$ and $p(x|y)$ denotes the posterior probabilities of $X$ given the values of $Y$. The Information Gain measures the amount of decrease in the entropy of $X$ which reveals the additional information about variable $X$ as provided by $Y$. It is a symmetrical measure where gained information about $X$ after observing $Y$ is equal to the gained information about $Y$ after observing $X$ [20]. It is defined as:

$$IG(X|Y)=H(X)-H(X|Y) \tag{4}$$

The output of the symmetric uncertainty lies with the range of [0, 1] where the value 0 denotes that $X$ and $Y$ are independent variables and the value 1 denotes that either one of variable predicts the value of other variable completely.

Suppose $d$ number of features in data are represented as: $D=\{F_1,F_2,...,F_d\}$. Initially, irrelevant features are removed by measuring the correlation between the feature $F_i$ and class $C$. If $S(F_i,C)$ is greater than the specified threshold $\theta$, then the feature is considered as relevant else discarded from full feature set to produce $D'=\{F'_1,F'_2,...,F'_c|c\le d\}$. Then, the redundant features are eliminated by constructing a complete weighted graph, $G=(V,E)$ where features $F_i$ and $F_j$ acts as vertices and the correlation among these features are the weights of the edges. The vertices of the undirected graph represent the correlation between feature $F_i$ and class $C$, and edges represent the correlation between features $F_i$ and $F_j$. Then, the Prim algorithm is used to build the minimum spanning tree (MST) from the graph. The clusters are formed by partitioning the MST based on the removal of edges whose weight is less than its vertices. The representative features are selected from each cluster which has a higher value of correlation between feature and class. This selection of feature from each cluster produces the final feature subset.

### 3.4 Ant lion optimization

Recently proposed Ant Lion optimization algorithm by Mirjalili [41] is attracting researchers for solving optimization problems. It is a nature-inspired algorithm which imitates the hunting behavior of the antlions. The algorithm has been applied successfully to many engineering problems due to its good exploration and exploitation capabilities [42–44]. It balances the searching of optimal solutions through a single parameter which controls the exploration as well as exploitation. Exploration is realized by employing the concept of random walks and the roulette wheel strategy for generating diverse solutions. Exploitation capability is achieved by shrinking the searching area of the problem adaptively and with elitism. The ALO is modified by using the crossover operator to improve the search capabilities of the algorithm.

In the optimization algorithm, the act of catching Ant Lion's prey is simulated mathematically for finding the optimal solution of the given problem. The hunting simulation of antlions $(AL)$ is depicted by moving the ants $(AN)$ over the search space where antlions entrap them and become fitter. To model this idea, the ants are required to perform the random walk in the locality of the antlions so that they can be captured and consumed by the antlions.

Initially, the positions of antlions are selected randomly within the search space of the problem. Then, the random walk of an ant $(AN_l)$ is performed around an antlion $(AL_j)$ which is defined as follows:

$$W^t=\left[0,\sum_{i=1}^{t_1}2r-1,\sum_{i=1}^{t_2}2r-1,...,\sum_{i=1}^{t_T}2r-1\right] \tag{5}$$

where $t$ denotes the current iteration, $T$ is the maximum iterations and $r$ is a random variable which is given as follows:

$$r = \begin{cases} 1 \; if \; rand > 0.5 \\ 0 \; otherwise \end{cases} \tag{6}$$

where $rand$ denotes the randomly generated real values in the range of [0, 1] uniformly. Thus, the random walks of the ants denote the cumulative sum of random variable up to $t$ iterations. Further, min-max normalization is used to confine these random walks of ants within the search boundaries of the problem as follows:

$$W_i^t = \frac{\left(W_i^t - a_i\right)\left(h_i^t - g_i^t\right)}{b_i - a_i} + g_i^t \tag{7}$$

where $a_i$ denotes the lower bound and $b_i$ denotes the upper bound of random walk's $i^{th}$ dimension, $g_i^t$ is lower bound and $h_i^t$ is the upper bound of $i^{th}$ dimension at $t^{th}$ iteration. At each iteration, the lower and upper bounds are updated to simulate the trapping of ants in the antlion's pit. For an ant ($AN_l$) at $t^{th}$ iteration, it is defined as follows:

$$g_i^t = AL_j^t + g_i^t, \quad \text{and} \quad h_i^t = AL_j^t + h_i^t \tag{8}$$

where $AL_j^t$ represents the position of the selected $j^{th}$ antlion at $t^{th}$ iteration around which ants are trapping, and $g_i^t$ and $h_i^t$ represents the lower and upper bounds of $i^{th}$ dimension for $l^{th}$ ant at $t^{th}$ iteration respectively. The antlion catches their prey by sliding them downwards into the pit. This act has represented by reducing the radius of ant walks adaptively as follows:

$$g_i^t = \frac{g_i}{I}, \quad \text{and} \quad h_i^t = \frac{h_i}{I} \tag{9}$$

where $g_i$ is the lower bound of $i^{th}$ dimension for the given problem, $h_i$ is the upper bound of $i^{th}$ dimension for the given problem and $I$ is the ratio which is defined as:

$$I = 10^w \frac{t}{T} \tag{10}$$

where $w$ is the parameter that helps to adjust the level of exploitation. The newer positions of the ants are updated using elitism; a technique used in optimization algorithms to maintain the best solutions at each iteration. It involves an elite antlion ($AL_{ET}$) that has the fittest solution and an antlion ($AL_{RW}$) which is selected through roulette wheel strategy. The elite antlion affects the movement of all ants whereas the selected antlion affects the movements of nearby ants only. The newer position of an ant was determined by measuring the average of a random walk around elite and selected antlion. However, we discourage the averaging operation and has employed the blend crossover operation ($BLX$) [45] for determining the newer positions of ants. This crossover operation has implemented in many evolutionary algorithms for its success in the global exploration of the solutions. The $BLX$ operator has preferred over the averaging operation because the newer solutions are generated randomly within the uniformly extended search range of random walks around elite and the selected antlion. On the contrary, the averaging operation gives deterministic newer solutions representing the mid-point of the ant walk around elite and the selected antlion. Let $R_{l,j}^t$ denote the normalized random walk of $l^{th}$ ant around $j^{th}$ antlion at $t^{th}$ iteration. Then, the $BLX$ operation is given as follows:

$$\begin{aligned} x_1 &= min\left(R_{l,ET}^t, R_{l,RW}^t\right) - \eta \cdot m \\ x_2 &= max\left(R_{l,ET}^t, R_{l,RW}^t\right) + \eta \cdot m \end{aligned} \tag{11}$$

where $m$ is given as $|R_{l,ET}^t - R_{l,RW}^t|$ and $\eta$ is the positive constant which controls the exploration and exploitation of the search space. A uniform random number is selected in-between $x_1$ and $x_2$ to determine the new position of the ant as follows:

$$Sn = rand(x_1, x_2)$$

Figure 4 depicts the working of the $BLX$ operator assuming that $R_{l,ET} < R_{l,RW}$. The extended search range for generating new solutions is shown which varies from $R_{l,ET} - \eta \cdot m$ to $R_{l,RW} + \eta \cdot m$. The stretching of the range depends upon the parameter $\eta$. The newer solution can be selected randomly within this extended search range, for instance, $AN_{BLX}$. The averaging operation is also depicted as $AN_{Avg.}$. The $BLX$ operator helps to improve the exploration as well as exploitation ability of the ALO. It helps to avoid the local optima by generating solutions randomly at each iteration which are not tried earlier.

In case of feature weighting which is a continuous search problem $AN_l^t = Sn$. For feature subset selection, we have used an S-shaped transfer function to convert the real values into binary values for selecting or rejecting the features. The output of the function is compared with the random number generated in the interval of [0, 1] for conversion and is given as follows:

$$P = \frac{1}{1 + e^{-Sn}} \quad \text{and} \quad AN_l^t = \begin{cases} 1 \; P > rand \\ 0 \; otherwise \end{cases} \tag{12}$$

The fitness of newer solutions is determined to update the global best solution if a better solution is obtained. It is

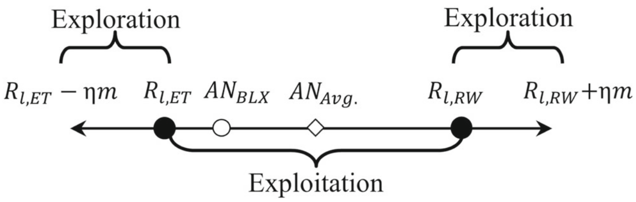


**Fig. 4** Blend Crossover operator

equivalent to the consumption of prey by antlion and laying of its new trap for next prey which is defined as follows:

$$AL_j^{t+1} = AN_l^t \quad \text{if} \quad f\left(AN_l^t\right) > f\left(AL_j^t\right) \tag{13}$$

where $f$ denotes fitness function which is the classification accuracy measured with $k$-NN classifier and is given as follows:

$$f = \frac{number\ of\ correctly\ classified\ instances}{Total\ number\ of\ instances} \tag{14}$$

## 4 Results and discussions

In this section, the experiments and analysis of the above-discussed strategies have been provided. We discuss the findings and also compare the results with similar work in the field to validate the proposed strategies.

### 4.1 Data and evaluation criteria

The experiments have been performed on 12 high-dimensional datasets containing features ranging from 274 to 10,304. These datasets have obtained from the ASU feature selection [46] and the UCI machine learning [47] repositories. These benchmarking datasets belong to different application fields, and their details are provided in Table 1. These datasets are evaluated using a 10-fold cross-validation procedure. This procedure involves the splitting of original data into 10 sets randomly such that each class of the data shares the same proportion in each set. The training data is obtained by selecting 9 sets while the remaining one set is used as testing data. The process is then repeated 10 times so that each set will be used as testing data once. The performance is measured with Nearest Neighbor (NN) classifier in terms of classification accuracy, the percentage of selected features, and runtime. We have used $k$-nearest neighbor classifier due to its efficiency in many practical applications, and it requires only one parameter (i.e., $k$) to optimize. We have implemented the proposed strategies on a system having Intel Xeon CPU, 8 GB RAM and MATLAB 2016 development environment. Furthermore, we have carried out 20 independent runs of each dataset wherever wrapper method is employed to eliminate the random discrepancies in the outcomes. It allows better analysis of the results leading to robust conclusions. It is in contrast to the FAST filter method that gives deterministic results.

From the above discussed existing works, five methods can be identified based on the feature selection and/or feature weighting. The experiments are performed considering these five methods and the four proposed strategies to provide a proper comparative study. The goal is to imitate the current work of field on the considered benchmarking datasets to deliver a fair assessment of results when compared with proposed strategies. In the paper, besides a full set of features (FULLSET), nine methods are evaluated which include: feature selection using filter and wrapper method (denoted as FFS and WFS respectively), wrapper based feature weighting (WFW), the hybrid method for feature selection (FFS-WFS), wrapper based simultaneous feature selection and feature weighting (SFSFW) and the proposed strategies FFS-WFW, WFW-FFS, WFS-WFW and WFW-WFS. We have implemented these nine methods using only FAST and ALO techniques to keep the baseline of the experiments uniform. It is worth noting that FFS denotes the FAST method while feature selection, feature weighting, hybrid feature selection and simultaneous feature selection and feature weighting methods are inspired from the recent works.

**Table 1** High dimensional datasets used in the study

| Dataset | Instances | Features | Classes | Feature type | Nature of data |
|---|---|---|---|---|---|
| Arcene | 200 | 10000 | 2 | Continuous | Mass Spectrometry |
| Arrhythmia | 452 | 274 | 16 | Continuous | Medical data |
| Basehock | 1993 | 4862 | 2 | Continuous | Text Classification |
| Isolet | 1560 | 617 | 26 | Continuous | Spoken letter recognition data |
| Leukemia | 72 | 7070 | 2 | Discrete | Medical data |
| Lung | 203 | 3312 | 5 | Continuous | Medical data |
| Lung Discrete | 73 | 325 | 7 | Discrete | Medical data |
| Madelon | 2600 | 500 | 2 | Continuous | Artificial data |
| Micromass | 571 | 1300 | 20 | Continuous | Mass Spectrometry |
| Nci9 | 60 | 9712 | 9 | Continuous | Biological data |
| Orlraws10P | 100 | 10304 | 10 | Continuous | Face Image data |
| Yale | 165 | 1024 | 15 | Continuous | Face Image data |

We have also used Extended Adjusted Ratio of Ratios (EARR) [48], a multi-criteria metric which unifies classification accuracy, the percentage of selected features, and runtime. The metric allows computation of relative importance of classification accuracy, feature selection and runtime of different methods. Considering $M$ datasets $\{A_1, A_2, \ldots A_M\}$ which are being evaluated on set of $N$ methods represented $\{B_1, B_2, \ldots B_N\}$. Let classification accuracy, the percentage of selected features and runtime of method $B_u$ on data $A_i$ are denoted as $acc_u^i$, $fs_u^i$ and $rt_u^i$ respectively. Then, EARR of $B_u$ with respect to rest of $(N-1)$ methods is given as follows:

$$EARR_{B_u}^{A} = \frac{1}{N-1}\sum_{v=1\wedge v\neq u}^{N} EARR_{B_u,B_v}^{A} \tag{15}$$

$$EARR_{B_u,B_v}^{A_i} = \frac{acc_u^i/acc_v^i}{1+\beta\cdot\log_{10}(fs_u^i/fs_v^i)+\gamma\cdot\log_{10}(rt_u^i/rt_v^i)} \tag{16}$$

where $\beta$ and $\gamma$ donates the relative importance that user wants to give to feature selection and runtime over accuracy when comparing various methods. To determine whether $B_u$ is the best, equal or the worst method as compared to $B_v$ on data $A_i$, $EARR_{B_u,B_v}^{A_i}$ is equated to $EARR_{B_v,B_u}^{A_i}$ for higher, equal or lower value respectively. In experiments, the values of both $\beta$ and $\gamma$ are set at 0.1%, 1%, 10%, representing the will of the user for trading accuracy with features as well as the runtime in proportions of 10:1, 1:1 and 1:10 respectively.

For validation, a comparative analysis of the proposed strategies has also been supported by various statistical tests. We use $\chi_F^2$ Friedman test [49], a non-parametric approach which measures the average rank $(rank_i)$ of each method. The acceptance or the rejection of the null hypothesis (i.e., the performance of each method is not statistically different) depends upon the $p$-value which is estimated from chi-square distribution. Additionally, another statistical test, Holm post hoc test [9] is applied considering a minimum ranked method to determine whether its performance is statistically significant with respect to the rest of the methods. Considering $N$ methods, the test performed pairwise comparison as $z = (rank_i - rank_j)/\sqrt{N(N+1)/6M}$ , where the method that has a minimum Friedman rank is considered as a control method. The $p$-value is calculated from the normal distribution using the value of $z$. The Holm step down procedure compares the smallest $p$-value with $\alpha/(N-1)$. If $p$ is less, the hypothesis is rejected, and the next higher value is tried with $\alpha/(N-2)$. If this hypothesis is also rejected, next higher $p$-values are tried. The procedure continues until all $p$-values are tried, or hypothesis gets accepted.

## 4.2 Parameter setting

The parameters of the clustering based FAST method is kept the same as mentioned in [26]. The parameter settings of the modified Ant Lion optimization algorithm for feature selection and feature weighting are presented in Table 2. The population size and a maximum number of iterations are chosen to be 20 and 200 respectively [50]. It is because searching through the large solution space of high dimensional data requires higher iterations to avoid the stagnation. The values of $w$ are kept same as mentioned in [41] whereas the value of blend crossover is set to 0.5 [45]. The search range of $k$ neighborhood is set in the range of [1, 11] experimentally.

## 4.3 Results and analysis

The results have evaluated and discussed based on classification accuracy, the percentage of selected features and runtime.

**Table 2** Parameters setting for ant lion optimization

| Parameter | Settings | |
|---|---|---|
| Ant Lion optimization for continuous search | Feature Weighting | |
| Ant Lion optimization for combinational search | Feature Selection | |
| Feature weights | [0, 1] | |
| Population of Antlions and Ants | 20 | |
| Iterations | 100 | FFS-WFS, FFS-WFW, WFW-FFS |
| | 200 | WFS, WFW and SFSFW |
| | 200 (100 for FS and 100 for FW) | WFW-WFS and WFS-WFW |
| $w$ (parameter for controlling the level of exploitation) | 2 | $t > 0.10T$ |
| | 3 | $t > 0.50T$ |
| | 4 | $t > 0.75T$ |
| | 5 | $t > 0.90T$ |
| | 6 | $t > 0.95T$ |

### 4.3.1 Classification accuracy

Table 3 shows the classification accuracy of nine methods for the high dimensional datasets along with the overall average accuracy. It also provides the accuracies when all features (FULLSET) are used for the evaluation of datasets. Further, the standard deviation of the accuracies for the independent executions is also presented except FULLSET and FFS method due to the deterministic outcomes. The results show that feature selection methods lack in accuracy as compared to feature weighting which upholds the findings of Wettschereck et al. [12]. Maximum average accuracy is 83.15% for feature selection and 88.47% for feature weighting approaches. Moreover, in the case of feature selection, FAST filter method (72.99%) is unable to compete with its wrapper counterpart (83.15%) that uses Ant Lion optimization for selecting features. The overall best method is one of the proposed strategies, i.e., WFS-WFW, which achieves 85.64% accuracy. This method performs better than WFS, which is the second-best method. A comparison of WFS-WFW with the other proposed strategies, FFS-WFW, WFW-FFS, and WFW-WFS shows the gain in accuracy of 7.93%, 14.06%, and 1.13% respectively.

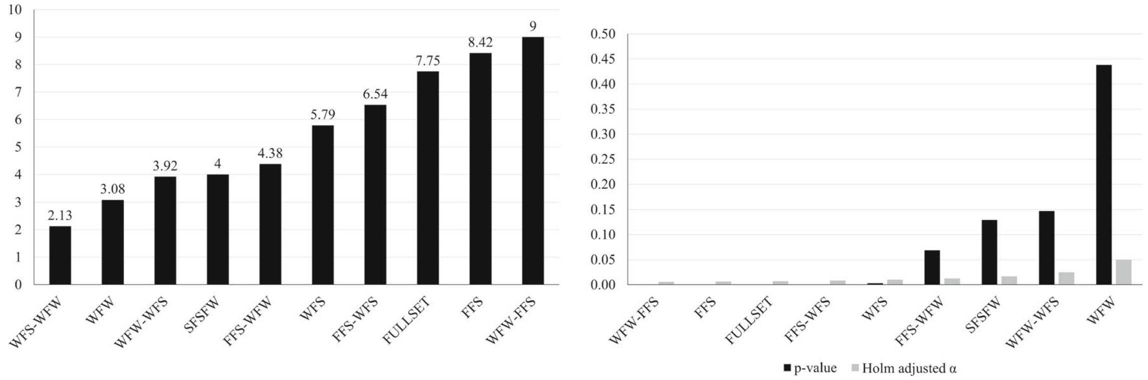
We further perform Friedman test followed by Holm post hoc test to analyse the outcomes of experiments. Figure 5a shows the Friedman ranks of ten methods. The obtained $p$-value is 9.90E-11, which is less than the assumed significance level, $\alpha = 0.05$ and therefore, it shows the significant differences in classification accuracies of the methods. WFS-WFW attains the best rank with a difference of 0.95 from the subsequent method. Furthermore, considering the best method as a control method, Holm post hoc test is applied to the rest of the methods, and the results are presented in Fig. 5b. It can be seen that the results of the best method are significantly better than FULLSET, FFS, WFS, FFS-WFS and WFW-FFS methods.

### 4.3.2 Percentage of selected features

Table 4 outlines the percentage of selected features for all methods excluding FULLSET. The results show that feature selection outperforms feature weighting by selecting minimum features. The results are expected because, in feature selection approach, features are rejected completely even though their relevancy is less. Hybrid feature selection method reduces maximum features while ALO based feature weighting reduces approximately 7% features only. A large difference of more than 92% has observed when compared to the best (FFS-WFS) performing method with the worst (WFW) performing method. Nonetheless, the FAST filter method lacks in classification accuracy despite selecting the minimum number of features. The results also

**Table 3** Average classification accuracy (and standard deviation) of various methods on high dimensional data

| Datasets | Methods | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FULLSET | FFS | WFS | WFW | FFS-WFS | SFSFW | FFS-WFW | WFW-FFS | WFS-WFW | WFW-WFS |
| Arcene | 87.14 | 84.59 | 91.11 (0.35) | 91.72 (0.66) | 89.22 (0.40) | 92.06 (0.89) | 91.88 (1.42) | 83.08 (2.68) | 92.34 (1.15) | 91.74 (0.74) |
| Arrhythmia | 62.86 | 63.56 | 63.45 (0.69) | 67.47 (1.19) | 63.24 (0.77) | 67.44 (0.98) | 65.74 (0.88) | 61.91 (1.35) | 67.80 (0.75) | 63.91 (1.27) |
| Basehock | 88.11 | 90.56 | 90.30 (0.13) | 91.50 (1.28) | 90.59 (0.20) | 90.45 (0.68) | 93.61 (0.19) | 91.61 (0.53) | 91.37 (0.84) | 90.87 (0.84) |
| Isolet | 89.68 | 49.10 | 91.97 (0.14) | 92.40 (0.62) | 46.26 (1.00) | 91.88 (0.54) | 50.13 (0.17) | 44.78 (7.03) | 92.39 (0.68) | 91.90 (0.43) |
| Leukemia | 90.36 | 97.50 | 97.52 (0.38) | 98.46 (1.19) | 100 (0) | 98.18 (1.07) | 100 (0) | 98.03 (1.03) | 99.23 (1.09) | 98.65 (1.30) |
| Lung | 95.60 | 96.09 | 96.80 (0.32) | 97.58 (0.50) | 97.59 (0.03) | 97.47 (0.30) | 97.61 (0.28) | 95.54 (0.71) | 97.97 (0.47) | 96.87 (0.35) |
| Lung Discrete | 89.72 | 73.00 | 92.51 (0.23) | 94.43 (1.63) | 76.95 (1.36) | 93.30 (1.10) | 81.85 (1.60) | 72.14 (7.41) | 93.86 (1.16) | 92.72 (1.51) |
| Madelon | 74.62 | 67.73 | 72.96 (1.38) | 80.96 (1.62) | 67.10 (0.26) | 79.95 (0.70) | 68.57 (0.24) | 66.79 (1.72) | 82.14 (2.06) | 77.22 (2.46) |
| Micromass | 72.02 | 49.60 | 76.05 (0.36) | 77.46 (0.96) | 52.47 (0.44) | 75.50 (0.81) | 54.49 (1.16) | 47.66 (3.54) | 77.76 (1.13) | 75.97 (1.19) |
| Nci9 | 50.32 | 49.18 | 56.57 (0.69) | 64.20 (3.70) | 53.85 (0.97) | 64.56 (2.62) | 59.46 (1.99) | 49.51 (2.94) | 63.00 (2.56) | 64.72 (2.29) |
| Orlraws10P | 96.00 | 95.00 | 99.00 (0) | 98.80 (0.42) | 98.90 (0.32) | 99.40 (0.52) | 98.20 (0.63) | 93.00 (3.27) | 99.00 (0) | 99.10 (0.32) |
| Yale | 64.98 | 60.03 | 69.53 (0.38) | 70.62 (1.73) | 69.40 (0.67) | 69.89 (0.77) | 71.00 (1.99) | 54.91 (5.07) | 70.88 (0.78) | 70.45 (1.22) |
| Average | 80.12 | 72.99 | 83.15 | 85.47 | 75.46 | 85.01 | 77.71 | 71.58 | 85.64 | 84.51 |



(a) Friedman Ranks

(b) Holm test

**Fig. 5** Statistical analysis for classification accuracy

show the shortcoming of feature weighting approach for not reducing features despite achieving good classification accuracy.

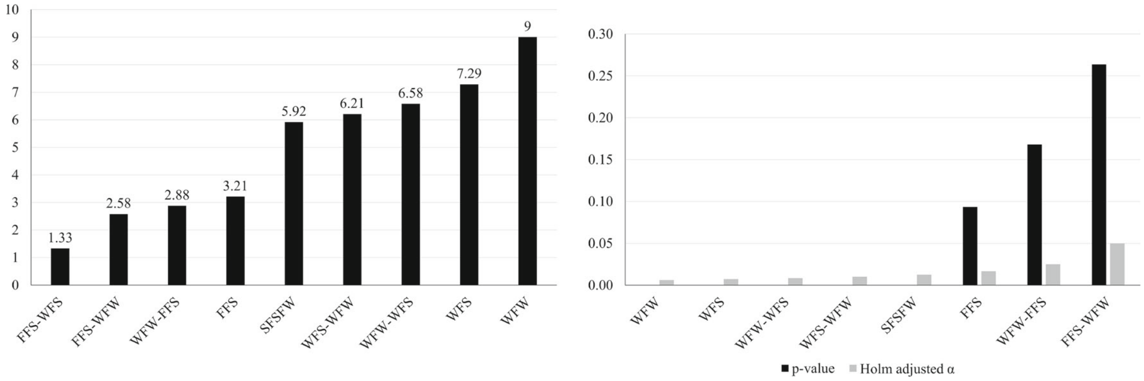
Further, we apply the Friedman test along with post hoc Holm test to the results. Figure 6a shows the Friedman ranks of the nine methods. The $p$-value obtained for the test is 2.03E-15 which is less than the presumed significance level, $\alpha = 0.05$ . Therefore, the percentage of selected features by all methods are significantly different. The overall best rank obtains by hybrid feature selection method, and this method acts as a control method in Holm post hoc test to realize whether its performance is statistically different as compared to the rest of the methods. The outcomes in Fig. 6b shows that FFS-WFS method is significantly better than WFS, WFW, SFSFW, WFS-WFW, and WFW-WFS.

### 4.3.3 Runtime

The runtime has shown in Table 5 after selecting a subset of features. The results are computed by evaluating the resultant data using 10-fold cross-validation procedure for 20 independent executions and are shown in $seconds$. The outcomes show that FFS-WFS method has minimum execution time on the datasets whereas WFW method has maximum execution time. FFS-WFS is 12 times faster than WFW and this large difference in runtime could play a vital role while designing practical applications. Figure 7a shows the Friedman rank of nine methods. The obtained $p$-value is 6.95E-15 for the test which is less than the presumed significance level, $\alpha = 0.05$. Therefore, the runtime of the methods is also significantly different. FFS-WFS attains

**Table 4** Percentage of selected features by various methods

| Datasets | Methods | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FFS | WFS | WFW | FFS-WFS | SFSFW | FFS-WFW | WFW-FFS | WFS-WFW | WFW-WFS |
| Arcene | 0.21 | 63.75 | 89.30 | 0.12 | 61.52 | 0.21 | 0.19 | 59.98 | 63.59 |
| Arrhythmia | 1.46 | 50.66 | 99.85 | 1.39 | 64.71 | 1.46 | 1.46 | 50.15 | 52.70 |
| Basehock | 0.72 | 64.27 | 97.90 | 0.59 | 63.52 | 0.71 | 0.72 | 64.15 | 63.86 |
| Isolet | 0.81 | 66.08 | 99.16 | 0.81 | 63.84 | 0.81 | 0.83 | 64.73 | 65.62 |
| Leukemia | 0.35 | 66.09 | 81.81 | 0.18 | 49.36 | 0.30 | 0.29 | 58.97 | 61.19 |
| Lung | 2.72 | 63.85 | 97.97 | 1.82 | 62.19 | 2.71 | 2.64 | 62.94 | 64.01 |
| Lung Discrete | 1.54 | 63.42 | 91.85 | 1.32 | 61.26 | 1.48 | 1.72 | 65.17 | 61.20 |
| Madelon | 0.40 | 51.68 | 99.68 | 0.40 | 63.90 | 0.40 | 0.40 | 51.68 | 50.56 |
| Micromass | 0.92 | 63.85 | 97.72 | 0.76 | 64.01 | 0.92 | 0.93 | 63.88 | 64.67 |
| Nci9 | 0.04 | 65.44 | 96.80 | 0.04 | 56.55 | 0.04 | 0.04 | 58.04 | 60.30 |
| Orlraws10P | 0.77 | 62.95 | 74.94 | 0.46 | 37.97 | 0.60 | 0.65 | 46.75 | 50.96 |
| Yale | 1.07 | 64.92 | 95.75 | 0.82 | 61.04 | 1.06 | 1.04 | 61.31 | 60.49 |
| Average | 0.92 | 62.25 | 93.56 | 0.73 | 59.15 | 0.89 | 0.91 | 58.98 | 59.93 |



(a) Friedman Ranks

(b) Holm test

**Fig. 6** Statistical analysis for percentage of selected features

the overall best rank, and therefore, we use it as a control method in the Holm test. Figure 7b shows that the best method is significantly different from all methods except FFS and FFS-WFW.

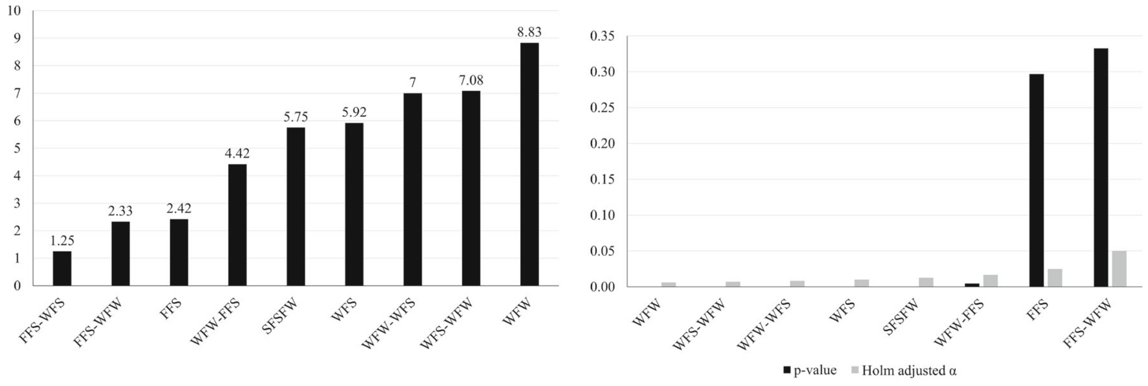
Figure 8 outlines the comparison between the accuracy and feature selection for the methods that were put to experiments in this paper. The plot is shown in order of increasing average accuracies for the better understanding of the results. It can be seen that features selected through filter methods do not yield efficient learning models as compared to the wrapper methods. The filter methods have reduced a large number of features from the data which results in a loss of reliable information as well. On the contrary, the wrapper methods keep relevant features in the resultant data but are unable to eliminate most of the irrelevant and redundant features due to limitations of optimization algorithms in high dimensional data. A sudden rise in the percentage of the feature selected along with accuracy can be observed. The proposed hybrid strategies, integrating the strengths of feature selection and feature weighting, yield more efficient learning models with lesser features (FFS-WFW and WFS-WFW).

From the outcomes of above-discussed analysis approaches, we can conclude the following:

1. No method is superior to others in all three analysis approaches. However, proposed strategies obtain best ranks in classification accuracy and second ranks in the percentage of selected features and runtime.
2. No method is significant in all three analysis approaches.
3. Simultaneous feature selection and feature weighting method always lack in all three performance analysis approaches.

**Table 5** Runtime of the resultant datasets after selecting features with nine methods

| Datasets | Methods | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | FFS | WFS | WFW | FFS-WFS | SFSFW | FFS-WFW | WFW-FFS | WFS-WFW | WFW-WFS |
| Arcene | 2.62 | 248.53 | 375.56 | 1.49 | 239.01 | 2.10 | 6.97 | 256.59 | 252.13 |
| Arrhythmia | 6.79 | 15.55 | 21.10 | 11.04 | 19.16 | 10.21 | 14.95 | 17.14 | 22.41 |
| Basehock | 91.08 | 1408.86 | 2355.42 | 61.36 | 1492.24 | 105.52 | 113.83 | 1531.42 | 1352.47 |
| Isolet | 70.01 | 154.71 | 227.06 | 58.08 | 162.61 | 102.32 | 114.42 | 171.36 | 163.25 |
| Leukemia | 1.62 | 68.56 | 86.85 | 1.52 | 52.92 | 1.57 | 5.01 | 66.71 | 63.62 |
| Lung | 4.91 | 76.76 | 140.82 | 3.24 | 83.39 | 4.35 | 9.95 | 88.92 | 90.17 |
| Lung Discrete | 2.18 | 2.79 | 4.35 | 1.56 | 2.86 | 1.60 | 5.41 | 3.17 | 3.40 |
| Madelon | 129.96 | 275.44 | 406.35 | 175.20 | 322.9 | 176.22 | 195.04 | 298.66 | 310.64 |
| Micromass | 11.95 | 88.97 | 141.71 | 3.76 | 88.62 | 10.82 | 27.57 | 100.47 | 96.11 |
| Nci9 | 1.60 | 73.96 | 115.36 | 1.09 | 63.37 | 2.30 | 4.91 | 74.90 | 70.08 |
| Orlraws10P | 2.56 | 124.62 | 151.10 | 1.41 | 75.64 | 1.54 | 6.74 | 105.82 | 109.97 |
| Yale | 3.10 | 9.44 | 34.81 | 1.39 | 9.02 | 2.95 | 8.33 | 9.59 | 10.13 |
| Average | 27.36 | 212.35 | 338.37 | 26.76 | 217.73 | 35.13 | 42.76 | 227.06 | 212.03 |



(a) Friedman Ranks

(b) Holm test

**Fig. 7** Statistical analysis for runtime

4. The proposed strategies perform well in all three analysis approaches, but no strategy emerges as a superior.

Therefore, to determine the best method, we use multi-criteria metric, EARR for further analysis.

### 4.3.4 Analysis of results using multi-criteria metric

The outcomes of EARR is calculated by setting both $\beta$ and $\gamma$ to 0.1%, 1% and 10% as shown in Fig. 9. The results of EARR show that the proposed strategies rank highest in all three scenarios, thereby confirming that orderly combination of feature selection and feature weighting results in better performance than existing work in the field.

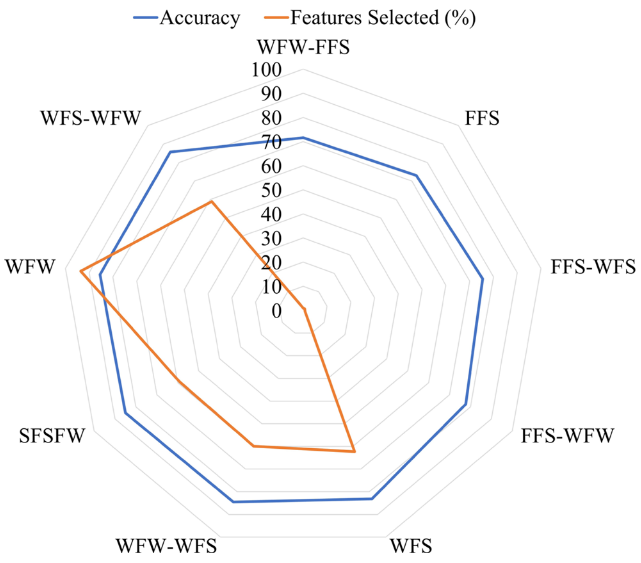
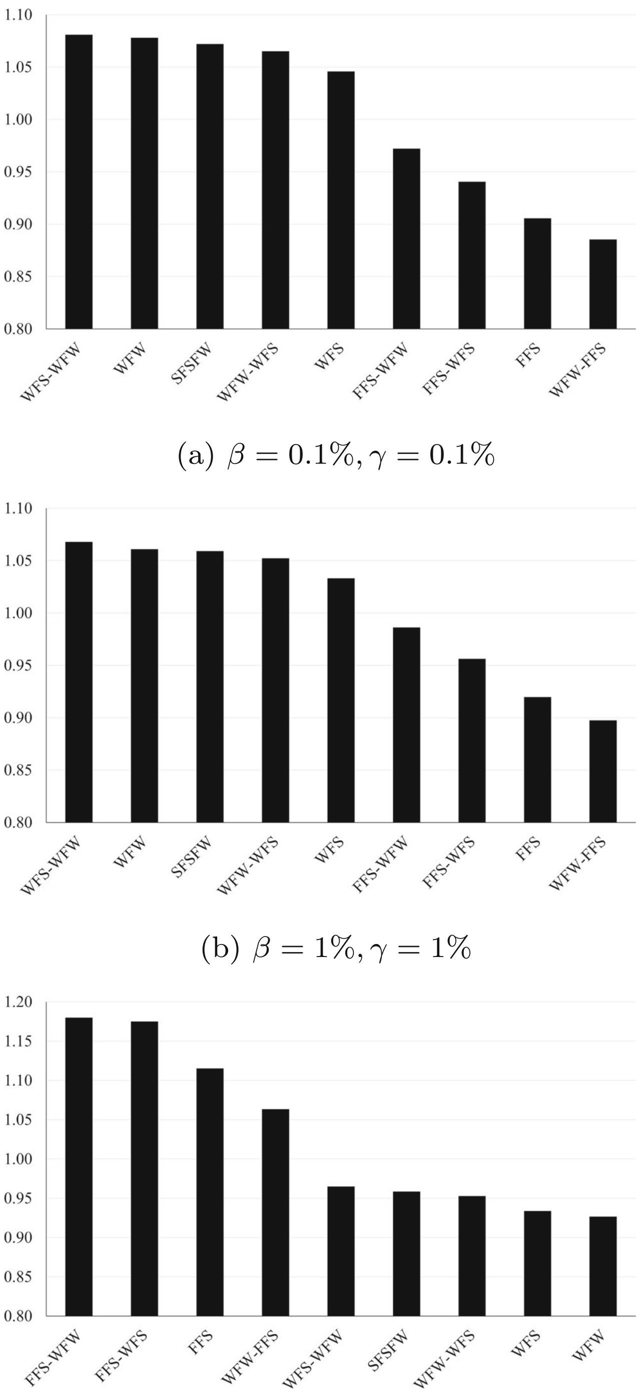


**Fig. 8** Comparison between accuracy and percentage of the feature selected

Furthermore, WFS-WFW emerges as the best choice in 0.1% and 1% scenarios. The first scenario depicted the favor to classification accuracy as compared to feature selection and runtime whereas the latter scenario depicted the balance between classification accuracy, feature selection, and runtime. In 10% scenario, FFS-WFW obtained the highest rank depicting favor to feature selection as well as runtime over classification accuracy. Therefore, based on the observations of the EARR, we conclude that the wrapper based FS followed by FW is the best choice when classification accuracy is the primary concern in high-dimensional datasets. In case, the runtime is the main concern, filter based FS followed by wrapper based FW is the optimum choice.

## 4.4 Comparison with other works

We have also made a comparison with the existing works based on classification accuracy as shown in Table 6. The different cross-validation methods used for evaluating the model of these works are also presented where 10-fold cross-validation is not used. The outcomes of the highest accurate strategy (i.e., WFS-WFW) are compared with the other methods to determine the superiority of the proposed work. The well-known methods used for the comparison consists of the filter (FRFS [51], KM-IG [52], SRFS [53], OSFSMI [54] and DCFS [55]), wrapper (CSO-$k$NN [56]), hybrid (EFR-ESO [57], UFSMB-PSO [58] and IGIS [5]) feature selection and wrapper based simultaneous feature selection and weighting (DE-CHCGA [6]) method. Other included methods are MV-NNMF [59] and KNN-$m_{0.5}$ [60] where MN-NNMF method used majority voting and non-negative matrix factorization for dealing with high dimensional data while the latter method improves the performance of NN classifier with data-dependent dissimilarity measure. To provide the fair and unbiased



(a) $\beta = 0.1\%, \gamma = 0.1\%$

(b) $\beta = 1\%, \gamma = 1\%$

(c) $\beta = 10\%, \gamma = 10\%$

**Fig. 9** EARR calculated for three different scenarios

comparison with the existing work, the accuracies of WFS-WFW are compared with those methods that have used 10-fold cross-validation on the NN classifier. The WFS-WFW strategy outperforms the other methods in 5 out of 8 datasets excluding a tie case as highlighted in the Table. The main reason for the success of the proposed strategy is the orderly combination of feature selection and feature weighting. The prior outcomes (Tables 3 and 4) indicate that feature selection yields less accurate models with fewer features whereas feature weighting yields the higher accurate models but lacks in dimensionality reduction. The WFS-WFW strategy that combines both approaches removes many unwanted features in the initial stage which helps in effective search of feature weights from the rest of the data dimensions. It helps to achieve higher accuracy with fewer features as compared to other methods. Hence, combining the strengths of feature selection and feature weighting ensures the success of the proposed strategies. These outcomes confirm the validity of the work presented in the paper.

### 4.5 Limitations and future work

Feature weighting plays an important role in improving the classification performance but the concept of weighting the features is not generalized for all machine learning algorithms. The linear weight assignment method utilized in this study is applied to various machine learning algorithms which includes Naive Bayes (NB), Support Vector Machines (SVM) and Artificial Neural Networks (ANN). However, while working with tree structured based learning algorithms, the weighting method should be selected carefully because some of its classifiers (ID3, CART and Random Forests) are insensitive to all monotone transformations on the features [61, 62]. Therefore, linear feature weighting will not be applied and only those features that have zero weight values will affect the classification performance. So, feature weighting is employed in another manner for these classifiers such as weighting the merit function [63] or testing data only [64].

The work presented in the paper provides plenty of opportunities for developing new hybrid methods based on feature selection and feature weighting approaches. Numerous filter and wrapper methods are available in the literature that will help the researchers to come up with new combinations of such hybrid methods. Furthermore, the choice of optimization algorithms is also a key aspect in wrapper methods for obtaining the best learning models. In the growing literature of optimization, several algorithms [65–68] are available which could enhance the performance even further. Moreover, the performance of these hybrid methods on other machine learning algorithms that are sensitive to feature weighting can be assessed.

Additionally, it would be interesting to investigate the performance of the proposed hybrid method on the problem of data streams. Data streams is a recent topic in machine learning that emphasize the real-world problems where data arrive continuously causing ever-growing dataset [69]. The new instances will arrive continuously one by one

**Table 6** Comparison of average accuracy with other methods

| Methods | Year | Datasets | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Arcene | Arrhythmia | Basehock | Isolet | Lung | Lung_discrete | Madelon | Orlraws10P |
| FRFS | 2013 | – | 67.39 | 88.05 | 84.71 | – | – | – | **99** |
| DE-CHCGA | 2015 | – | 66 | 86.45 | – | – | – | 54.21 | – |
| MV-NNMF | 2016 | 87.5 | – | – | – | – | – | – | – |
| KM-IG | 2016 | – | – | – | – | 89.29 | 86.19 | – | – |
| SRFS | 2017 | – | 67.27 | 85.87 | – | – | – | – | 92.67 |
| OSFSMI (66-34 train-test split) | 2017 | 65.06 | 47.45 | – | – | 71.85 | – | 56.4 | 61 |
| KNN-$m_{0.5}$ | 2017 | 84 | **71.9** | – | – | – | – | 59.23 | – |
| UFSMB-PSO | 2017 | – | 70.42 | 81.47 | – | – | – | – | 89.5 |
| EFR-ESO (70-30 train-test split) | 2017 | – | 70.09 | – | 86.14 | – | – | 87.47 | |
| IGIS (5-CV) | 2018 | – | – | **94.94** | – | – | – | – | 91.6 |
| CSO-$k$NN (70-30 train-test split) | 2018 | – | 67.6 | – | 85.09 | – | – | 84.28 | – |
| DCFS | 2018 | – | 57.87 | 86.56 | 60.17 | 89.11 | 69.75 | – | – |
| Proposed work (WFS-WFW) | – | **92.34** | 67.8 | 91.37 | **92.39** | **97.97** | **93.86** | **82.14** | **99** |

The outcomes in bold signifies the maximum accuracy for a dataset

or in batches which make the problem more complex as compared to the static data set. In this area, the balance between accuracy and computation cost is a prime concern because time is bounded by the incoming speed of instances and learning algorithms need to be updated constantly with new data. Therefore, higher computation requirements make the wrapper methods not feasible for the problem and Ramírez-Gallego et al. [70] found out that no wrapper method was proposed for this online problem. He further suggested that the combination of filter and wrapper methods of feature selection can be used to achieve better accuracy with lower computations. But, the filter feature selection-wrapper feature weighting strategy is better than the hybrid feature selection method and therefore, it is more suitable for the problem of data streams.

## 5 Conclusion

Feature selection has been explored more extensively as compared to feature weighting, but feature weighting is more successful. However, feature weighting is the more computationally complex problem because weights of the features are searched from defined search space whereas feature selection is a binarized version of the feature weighting. In fact, feature selection is a subset of feature weighting and therefore theoretically, it will result in lower (average scenario) or equal (best scenario) performance when compared with the feature weighting.

The paper presents the hybrid strategies using feature selection and feature weighting approaches for high dimensional data on the NN classifier. Considering FAST filter method and Ant Lion optimization based wrapper method, four hybrid strategies are presented. FAST method is used for feature selection only whereas ALO is used for both feature selection and feature weighting. Moreover, we have also used the ALO for individual feature selection, individual feature weighting, hybrid feature selection, and simultaneous feature selection and feature weighting to realize the concept of recent wrapper based methodologies. The datasets selected for experiments represent the classification problems from different areas of research for the effective analysis and comparison with the state of the art methodologies. We have used extended adjusted ratio of ratios metric to recognize the best performing method. The results are also analysed by utilizing statistical tests. Experiments using eight methods show that the proposed hybrid strategies have gained better performance. The proposed strategies have obtained the highest ranks in three different scenarios of multi-criteria metric. The work is further strengthened by the outcomes of the statistical tests towards hybrid methods as well as accuracy based comparison with the existing works. The hybrid feature selection- feature weighting wrapper method is best for the application where higher accuracy is the main concern whereas hybrid feature selection filter-feature weighting wrapper method is best for applications that need low response time but with less classification accuracy. In conclusion, feature weighting improves the learning models obtained from the feature selection either by filter or wrapper method. Hence, the hybridization of feature selection and feature weighting is fruitful for achieving better classification accuracy with minimum features.

**Acknowledgements** The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

## Compliance with Ethical Standards

**Conflict of interests** There is no conflict of interest.

## References

1. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37
2. Hughes G (1968) On the mean accuracy of statistical pattern recognizers. IEEE Trans Inform Theory 14(1):55–63
3. Koller D, Sahami M (1996) Toward optimal feature selection. Technical report, Stanford InfoLab
4. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5(Oct):1205–1224
5. Nakariyakul S (2018) High-dimensional hybrid feature selection using interaction information-guided search. Knowl-Based Syst 145:59–66
6. Pérez-Rodríguez J, Arroyo-Peña AG, García-Pedrajas N (2015) Simultaneous instance and feature selection and weighting using evolutionary computation: proposal and study. Appl Soft Comput 37:416–443
7. Wu X, Yu K, Ding W, Wang H, Zhu X (2013) Online feature selection with streaming features. IEEE Trans Pattern Anal Mach Intell 35(5):1178–1192
8. Yu K, Ding W, Wu X (2016) Lofs: a library of online streaming feature selection. Knowl-Based Syst 113:1–3
9. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2001) Feature selection for SVMs. In: Advances in neural information processing systems, pp 668–674
10. Kelly JD Jr, Davis L (1991) A Hybrid Genetic Algorithm for Classification. In: IJCAI, vol 91, pp 645–650
11. Raymer ML, Punch WF, Goodman ED, Kuhn LA, Jain AK (2000) Dimensionality reduction using genetic algorithms. IEEE Trans Evol Comput 4(2):164–171
12. Wettschereck D, Aha DW, Mohri T (1997) A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artif Intell Rev 11(1-5):273–314
13. Kira K, Rendell L (1992) A Practical Approach to Feature Selection. In: Proceedings of ninth international workshop on machine learning, pp 249–256
14. Kononenko I (1994) Estimating attributes: analysis and extensions of RELIEF. In: European conference on machine learning. Springer, pp 171–182
15. Sun Y (2007) Iterative RELIEF for feature weighting: algorithms, theories, and applications. IEEE Trans Pattern Anal Mach Intell 29(6):1035–1051
16. Deng Z, Chung FL, Wang S (2010) Robust relief-feature weighting, margin maximization, and fuzzy optimization. IEEE Trans Fuzzy Syst 18(4):726–744
17. Gilad-Bachrach R, Navot A, Tishby N (2004) Margin based feature selection-theory and algorithms. In: Proceedings of the twenty-first international conference on machine learning. ACM, pp 43
18. Sun Y, Todorovic S, Goodison S (2010) Local-learning-based feature selection for high-dimensional data analysis. IEEE Trans Pattern Anal Mach Inteill 32(9):1610–1626
19. Hall MA (1999) Correlation-based feature selection for machine learning
20. Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 856–863
21. DeSarbo WS, Carroll JD, Clark LA, Green PE (1984) Synthesized clustering: a method for amalgamating alternative clustering bases with differential weighting of variables. Psychometrika 49(1):57–78
22. Huang JZ, Ng MK, Rong H, Li Z (2005) Automated variable weighting in k-means type clustering. IEEE Trans Pattern Anal Mach Intell 27(5):657–668
23. Domeniconi C, Gunopulos D, Ma S, Yan B, Al-Razgan M, Papadopoulos D (2007) Locally adaptive metrics for clustering high dimensional data. Data Min Knowl Disc 14(1):63–97
24. Jing L, Ng MK, Huang JZ (2007) An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Transactions on knowledge and data engineering 19(8):1026–1041
25. Chen X, Ye Y, Xu X, Huang JZ (2012) A feature group weighting method for subspace clustering of high-dimensional data. Pattern Recogn 45(1):434–446
26. Song Q, Ni J, Wang G (2013) A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Trans Knowl Data Eng 25(1):1–14
27. Revanasiddappa MB, Harish BS (2018) A New Feature Selection Method based on Intuitionistic Fuzzy Entropy to Categorize Text Documents. International Journal of Interactive Multimedia and Artificial Intelligence (In Press), pp 1–12
28. Liu Y, Wang G, Chen H, Dong H, Zhu X, Wang S (2011) An improved particle swarm optimization for feature selection. J Bionic Eng 8(2):191–200
29. Ghamisi P, Benediktsson JA (2015) Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci Remote Sens Lett 12(2):309–313
30. Hancer E, Xue B, Karaboga D, Zhang M (2015) A binary ABC algorithm based on advanced similarity scheme for feature selection. Appl Soft Comput 36:334–348
31. Hafez AI, Zawbaa HM, Emary E, Hassanien AE (2016) Sine cosine optimization algorithm for feature selection. In: International symposium on innovations in intelligent systems and applications (INISTA). IEEE, pp 1–5
32. Paredes R, Vidal E (2000) A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. Pattern Recogn Lett 21(12):1027–1036
33. Tahir MA, Bouridane A, Kurugollu F (2007) Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier. Pattern Recogn Lett 28(4):438–446
34. Barros AC, Cavalcanti GD (2008) Combining global optimization algorithms with a simple adaptive distance for feature selection and weighting. In: Proceedings of IEEE international joint conference on neural networks, pp 3518–3523
35. Derrac J, Triguero I, García S, Herrera F (2012) Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms. IEEE TRrans Syst Man Cybern Part B (Cybern) 42(5):1383–1397
36. Chuang LY, Yang CH, Wu KC, Yang CH (2011) A hybrid feature selection method for DNA microarray data. Comput Biol Med 41(4):228–237
37. Derrac J, Cornelis C, García S, Herrera F (2012) Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. Inf Sci 186(1):73–92
38. Apolloni J, Leguizamón G, Alba E (2016) Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. Appl Soft Comput 38:922–932
39. Duch W (2006) Filter methods. In: Feature extraction. Springer, pp 89–117
40. den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evol Comput 8(3):225–239

41. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
42. Gupta E, Saxena A (2016) Performance evaluation of antlion optimizer based regulator in automatic generation control of interconnected power system. Journal of Engineering 2016
43. Yao P, Wang H (2017) Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle. Soft Comput 21(18):5475–5488
44. Tharwat A, Hassanien AE (2018) Chaotic antlion algorithm for parameter optimization of support vector machine. Appl Intell 48(3):670–686
45. Eshelman LJ, Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. Found Genet Algorithm 2:187–202
46. Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H (2017) Feature selection: a data perspective. ACM Comput Surv (CSUR) 50(6):94
47. Asuncion A, Newman D (2007) UCI machine learning repository. available: http://archive.ics.uci.edu/ml/, 2018-04-23
48. Wang G, Song Q, Sun H, Zhang X, Xu B, Zhou Y (2013) A feature subset selection algorithm automatic recommendation method. J Artif Intell Res 47:1–34
49. Mateos-García D, García-Gutiérrez J, Riquelme-Santos JC (2016) An evolutionary voting for k-nearest neighbours. Expert Syst Appl 43:9–14
50. Sindhu R, Ngadiran R, Yacob YM, Zahri NAH, Hariharan M (2017) Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Comput Appl 28(10):2947–2958
51. Wang G, Song Q, Xu B, Zhou Y (2013) Selecting feature subset for high dimensional data via the propositional FOIL rules. Pattern Recogn 46(1):199–214
52. Dubey VK, Saxena AK, Shrivas MM (2016) A cluster-filter feature selection approach. In: International conference on ICT in business industry & government (ICTBIG). IEEE, pp 1–5
53. Wang Y, Wang J, Liao H, Chen H (2017) An efficient semi-supervised representatives feature selection algorithm based on information theory. Pattern Recogn 61:511–523
54. Rahmaninia M, Moradi P (2017) OSFSMI: online stream feature selection method based on mutual information. Applied Soft Computing
55. Gao W, Hu L, Zhang P (2018) Class-specific mutual information variation for feature selection. Pattern Recogn 79:328–339
56. Gu S, Cheng R, Jin Y (2018) Feature selection for high-dimensional classification using a competitive swarm optimizer. Soft Comput 22(3):811–822
57. Dowlatshahi MB, Derhami V, Nezamabadi-pour H (2017) Ensemble of filter-based rankers to guide an epsilon-greedy swarm optimizer for high-dimensional feature subset selection. Information 8(4):152
58. Wang Y, Wang J, Liao H, Chen H (2017) Unsupervised feature selection based on Markov blanket and particle swarm optimization. J Syst Eng Electron 28(1):151–161
59. Seetha H, Murty MN, Saravanan R (2016) Classification by majority voting in feature partitions. Int J Inf Decis Sci 8(2):109–124
60. Aryal S, Ting KM, Washio T, Haffari G (2017) Data-dependent dissimilarity measure: an effective alternative to geometric distance measures. Knowl Inf Syst 53(2):479–506
61. Breiman L (2017) Classification and regression trees. Routledge, Evanston
62. Friedman JH (2006) Recent advances in predictive (machine) learning. J Classif 23(2):175–197
63. Maudes J, Rodríguez JJ, García-Osorio C, García-Pedrajas N (2012) Random feature weights for decision tree ensemble construction. Inf Fusion 13(1):20–30
64. Galili T, Meilijson I (2016) Splitting matters: how monotone transformation of predictor variables may improve the predictions of decision tree models. arXiv:161104561
65. Arora S, Singh S (2017) An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization. Int J Interact Multimed Artif Intell 4(4):14–21
66. Meza J, Espitia H, Montenegro C, Giménez E, González-Crespo R (2017) Movpso: Vortex multi-objective particle swarm optimization. Appl Soft Comput 52:1042–1057
67. Aydilek IB (2018) A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. Appl Soft Comput 66:232–249
68. Han X, Liu Q, Wang H, Wang L (2018) Novel fruit fly optimization algorithm with trend search and co-evolution. Knowl-Based Syst 141:1–17
69. Gaber MM (2012) Advances in data stream mining. Wiley Interdiscip Rev Data Min Knowl Discov 2(1):79–85
70. Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F (2017) A survey on data preprocessing for data stream mining: current status and future directions. Neurocomputing 239:39–57

**Dalwinder Singh** received the Master of Technology degree in Computer Science from Punjabi University, India. Now, he is pursing PhD in computer science and engineering from Sant Longowal Institute of Engineering and Technology, India. He has published several papers in conference proceedings. His research interests include machine learning, image processing and computer vision.

**Birmohan Singh** is working as associate professor in the Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal. He had received the Master in Engineering degree in Computer Science and Engineering from Thapar Institute of Engineering and Technology and completed his PhD from SLIET, Longowal. His research interests include signal processing, image processing, machine learning and metaheuristic algorithms.