Feature grouping and selection

Zheng, Ling; Chao, Fei; Parthaláin, Neil Mac; Zhang, Defu; Shen, Qiang

Published in:
Information Sciences

DOI:
10.1016/j.ins.2020.09.022

Publication date:
2021

Citation for published version (APA):
Zheng, L., Chao, F., Parthaláin, N. M., Zhang, D., & Shen, Q. (2021). Feature grouping and selection: A graph-based approach. *Information Sciences*, *546*, 1256-1272. https://doi.org/10.1016/j.ins.2020.09.022

Highlights

Feature Grouping and Selection: A Graph-based Approach

Ling Zheng, Fei Chao, Neil Mac Parthaláin, Defu Zhang, Qiang Shen

- A novel graph-based feature grouping framework with different types of feature relationships.
- An undirected graph representing features as vertices with edges.
- Both straightforward and metaheuristic search methods for graph optimisation.

Feature Grouping and Selection: A Graph-based Approach

Ling Zheng[a], Fei Chao[a,b], Neil Mac Parthaláin[b], Defu Zhang[a], Qiang Shen[b]

[a]*School of Informatics, Xiamen University, Xiamen, 361005, China.*
[b]*Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK.*

Abstract

Most current feature selection techniques are focused on the incremental inclusion or exclusion of single individual features with respect to the candidate feature subset(s). The use of such approaches, where only the individual inclusion/exclusion of features is considered, means that information such as the collaborative contribution or correlation between features may be lost. The result is that the final selected feature subset may contain high levels of inter-feature redundancy, assuming that the key information embedded in the original feature set can still be retained. To address this problem, a general framework based on graph processing and three-way mutual information metrics is proposed in this paper that works by clustering similar features into groups, from which representative features are then drawn. Two different feature selection techniques based on this framework are presented: one by straightforward selection of representative features from the resulting feature groups and the other via a music-inspired metaheuristic search. Comparative experimental evaluation against traditional feature selection techniques over a diverse range of 20 benchmark datasets demonstrates the efficacy of the proposed approach. With these implementations, significant performance gains can be made in terms of classification accuracy in general and dimensionality reduction in particular while retaining feature semantics and considerably lessening the redundancy in the returned feature subsets.

Keywords:
Feature selection, feature grouping, graph processing, minimum spanning tree, harmony search

1. Introduction

Feature selection (FS) [13] is becoming an increasingly necessary step as the problem of dataset dimensionality grows ever more pervasive for real-world problems. Traditionally used in areas such as data mining, pattern recognition, and machine learning, FS is now being widely applied in many other domains [25]. This is because complex problems often contain large numbers of features, which may result in considerable computational overhead for data-driven knowledge discovery and decision-making tasks [17]. In particular, certain features may be either irrelevant or redundant and therefore offer no contribution when building robust predictive models. These features may involve a significant amount of noise or even be misleading and hence adversely affect the accuracy of a given model [7]. FS works by selecting a subset of features relevant to the problem at hand from the full set of available features and therefore preserves the original underlying semantics of the data. It can be used to remove irrelevant or redundant features, including noisy ones. FS techniques not only indirectly alleviate the computational overhead for subsequent learning mechanisms and use learned models but also offer more compact representations and reductions in data acquisition and storage requirements [29].

FS is considered to be an NP-hard problem [4]. Given a data set with n dimensions, FS techniques attempt to search for an "optimal" feature subset from 2^n candidate subsets. An exhaustive search can be used to guarantee a global optimum, but this also leads to an exponential increase in the computational time complexity. This means that exhaustive methods are often computationally intractable for feature subset searches where the datasets are large. One of the most common approaches to addressing this drawback is the greedy hill climbing strategies, where single features that result in the greatest increase in the subset score are greedily added to the candidate subset (i.e., the emerging subset of selected features). However, many of these approaches can easily become trapped in local optima. Alternatives that employ metaheuristics may facilitate escape from local regions and achieve, or at least approach the achievement of, the global optimum. Such work includes genetic algorithm(s) (GAs) [15], memetic algorithms (MAs) [12], particle swarm optimisation (PSO) [23], harmony search (HS) [26], and other nature-inspired techniques [25].

Although the existing work in the area of FS has resulted in many powerful techniques, for most approaches, important information regarding the

level of feature correlation is often ignored during the selection process (of course, not intentionally). Techniques that only iteratively include/exclude single individual features from a candidate subset of features are typical examples where this is the case. This loss of important information may result in the appearance of redundant features in the final selected subset [17]. FS algorithms based upon a high-level clustering framework can address this issue by grouping redundant features together and then selecting the representative features from each group to form the final subset of features. Initial clustering-related FS methods have been reported in the literature [9, 18, 27], where features are partitioned according to their similarity. Features with a high degree of similarity tend to be members of the same group. However, these intra-group feature similarities are calculated in isolation of their correlation to the decision, and the resultant group is more likely to contain features that involve redundant contributions with regard to the decision.

This paper proposes a novel graph-based feature grouping framework by considering different types of feature relationships in the context of decision-making (particularly for classification problems). This general framework can be implemented in a number of different ways. In this work, two particular instantiations are described, one based on the ranking of generated feature groups and the other based on a music-inspired metaheuristic (harmony search [26]). The framework itself involves a series of three primary steps:

1. First, an undirected graph is constructed by representing the features as vertices, where the edges are created by computing feature redundancy or collaboration with respect to the decision.
2. Second, an algorithm is devised to derive minimum spanning trees (MSTs) [24] from the undirected graph, where an MST is a graph representation of all the given features inter-connected such that the sum of the weights on the edges ensures a global minimum.
3. Finally, candidate feature groupings are obtained by manipulating the resulting MST through an iterative process of identifying and then removing a certain single edge from the MST.

The remainder of this paper is structured as follows. Section 2 reviews the relevant work on FS algorithms that are based on feature grouping or clustering. Section 3 presents the novel feature grouping approach by exploiting MSTs. Section 4 describes two FS methods based on feature grouping. Section 5 reports on an experimental evaluation and discusses the results,

demonstrating the efficacy of the proposed framework. Finally, the paper is concluded with further suggestions related to the current work in Section 6.

2. Background

Broadly speaking, FS approaches can be divided into three different categories (or variants thereof): wrapper, filter, and hybrid methods [29]. Filter-based FS techniques often make use of information-based metrics such as mutual information [4], correlation coefficients [28], message passing [21], and fuzzy-rough set dependency [2] to determine the relevance between the features themselves and the decision classes. In [8], the correlation coefficient is used to assist in identifying pair-wise redundancy between features and the relatedness between these features and the decision classes. The conventional K-means method is adopted for grouping features. Through the selection of a representative feature from each group, a feature subset is then formed. Whether a feature is chosen as the representative from a group depends on its correlation level to the decision classes. The features most correlated to the decision classes are selected to represent the full group of features. However, this approach requires the specification of the number of feature groups and hence the dimensionality of the selected feature subset in advance.

In [18], symmetric uncertainty, defined using the Shannon entropy, is employed as an estimation of the inter-feature correlation and correlations between the conditional features and the decision classes. This algorithm is proposed within the framework of graph processing methods by introducing and exploiting the following four concepts: *T-Relevance*, which indicates the symmetric uncertainty correlation between the features and the decision classes, *F-Correlation*, which implies the pair-wise correlation between any pair of features, *F-Redundancy*, which refers to the features that may be identified as redundant in a particular cluster, and *R-Feature*, which stands for a representative feature for a particular cluster. In this work, features that are less correlated with each other but are most highly correlated with the decision classes are selected to form the feature subset. Additionally, this ensures that the features with the most redundant information are clustered together. However, the *T-Relevance* of the features in the same resulting subset may have a large bias.

In [9], a greedy hill climbing approach for FS based on feature grouping is proposed, where an evaluation metric based on fuzzy-rough set dependency is utilised to determine the internal ranking of the features in each group as well

as the overall subset quality. Correlation coefficients are utilised to calculate the degree of redundancy between any pair of features. If the correlation between a given pair of features is greater than a predefined threshold, then one of the features is considered to be redundant, and both are assigned to the same group. Each individual group is initialised with a single distinct feature prior to recruiting other group members. As a result, an individual feature can be included or assigned to more than one group. Features are then internally ranked within the groups according to the fuzzy-rough set metric prior to returning the final subset. This algorithm is generally efficient, but it requires the determination of the value of the externally introduced threshold. Different configurations of this parameter may have significantly different impacts on the subset selection outcome.

In addition to the aforementioned representations of feature grouping-based FS methods, more recently, embedded FS techniques have been developed to identify homogeneous subgroups of a set of features. In [27], for instance, a feature grouping method is embedded within the process of sparse modelling. First, the popular OSCAR algorithm is used to generate the so-called co-efficiency matrix among features. The features that have identical coefficients are then grouped together, and those with coefficients of zero are immediately discarded. The new features formed by merging the features in the same group are subsequently used to train a sparse regression model. Testing against selected real-world datasets (e.g., breast cancer [3]), the regression models generated by this algorithm may be more robust than those obtained by conventional methods, though this is not always the case. Nevertheless, when the problem dimensionality increases, this algorithm tends to be more efficient than others.

While the techniques for FS outlined above are all interesting and potentially useful for conducting semantics-preserving data dimensionality reduction, they each have various shortcomings, as discussed. Inspired by this observation, the work below presents an alternative FS framework, following a rigorous theoretical approach based on graph processing [1, 10, 20].

3. Framework for feature grouping

Information regarding redundancy between relevant features (aka., inter-feature relevance) is the main focus of this work. Therefore, the identification of homogeneous feature groups may help to remove redundancy from the final returned feature subset. The framework described in this section is based

on the concepts of graph processing, clustering redundant features together. This is achieved using the idea of three-way mutual information [19]. In this framework, denoted graph-based feature grouping (GBFG) hereafter, one feature that is the most collaborative with the other selected features regarding the decision classes is selected as a representative from each group. The evaluation result of these representatives is used as the quality metric for comparing the existing and the currently emerging candidate feature grouping. The final grouping is automatically determined according to whether the quality of the previous (stored) groupings is improved compared with that of the current candidate groupings. Therefore, the grouping can be viewed as an iterative refinement process. The new GBFG framework is illustrated in Fig. 1. The technical details of GBFG are presented below and in the next section.

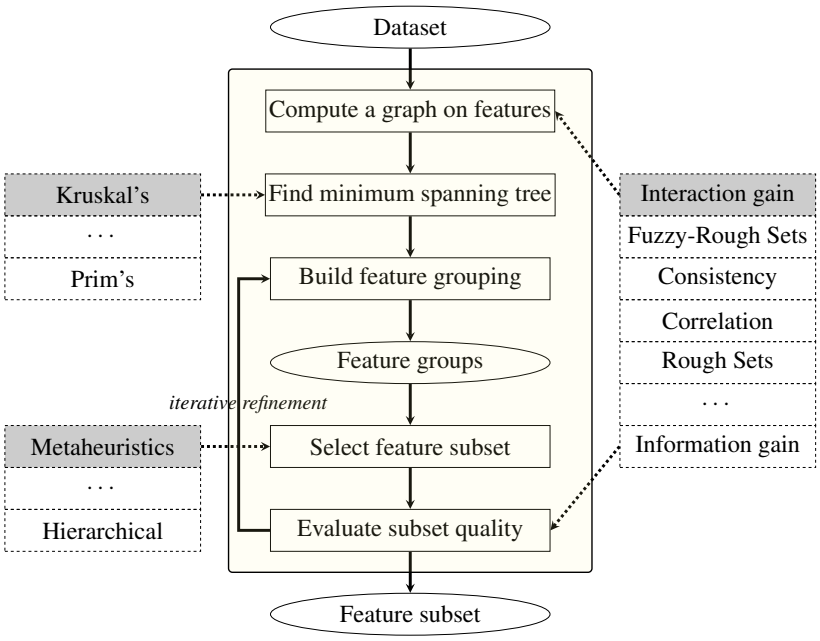


Figure 1: Framework for feature selection using GBFG

3.1. Relationship metrics and interaction gain

This subsection addresses the issue of what features may be grouped together. Generally, redundant features are those whose information content is already present in other features, and irrelevant features are those that

provide no information with respect to the decision classes, while naturally, relevant features are highly correlated to the decision classes. In this work, the relevant features are further divided into two classes according to the quantity of information that they carry, namely, strong relevance and weak relevance. According to the above relations between the features or those between the features and the decision classes, three possible methods may be considered for developing feature groups: simply clustering highly redundant features, clustering features that are equally relevant to the decision classes, and clustering features that are not only highly redundant but also equally relevant to the decision classes.

Many of the quality metrics developed in the literature (e.g., those reported in [9, 18, 27]) are capable of distinguishing between different types of features by ranking them using the calculated degree of correlation or dependency. High values of correlation not only indicate redundancy between features but also suggest that the relationship between the conditional features and the decision classes is strongly relevant. A moderate value typically implies a weak relevance, while a value level close to zero signifies irrelevance. Note, however, that most known metrics can only be used to assess the pairwise relationship between two features.

In this research, the measure of three-way mutual information is used to build a graph on the original features. This measure is also known as interaction gain [19], which is a metric that attempts to identify the relationships between the domain features, including collaboration and redundancy with respect to the decision classes. No problem-specific assumptions are required when three-way mutual information to feature selection is applied, regardless of whether a maximum-relevance or minimum redundancy-based approach is taken [28]. This is because the measure can be applied to help automatically identify the relationships between *subsets* of features that are similar with respect to the decision classes [19].

In an information system, data are depicted as a tuple $\langle X, Y \rangle$, where X is a non-empty set of finite objects, also referred to as the universe of discourse; and Y is a non-empty, finite set of features. For decision-making systems, $Y = \{A \cup C\}$, where $A = \{a_1, a_2, \cdots, a_{|A|}\}$ is the set of conditional features, $|A|$ denotes the cardinality of A, and C is a set of decision classes. Each feature $a_i \in A$ may be either discrete- or continuous-valued and has a value domain of V_{a_i}. The three-way mutual information of any given two (conditional) features $a_i, a_j \in A$ is computed as follows:

$$I(a_i, a_j, C) = \sum_{a_i} \sum_{a_j} \sum_{C} p(a_i, a_j, C) \log \frac{p(a_i, a_j, C)p(a_i)p(a_j)p(C)}{p(a_i, a_j)p(a_i, C)p(a_j, C)} \quad (1)$$

where $p(\dots)$ is the probability distribution function and the values of three-way mutual information are bounded by the inequality:

$$-[H(a_i) + H(a_j)] \leq I(a_i, a_j, C) \leq [H(a_i) + H(a_j)] \quad (2)$$

where $H(a_i)$ and $H(a_j)$ are the entropies of a_i and a_j, respectively. In practical use, such an interaction gain is often normalised to the interval $[-1, 1]$ by the term $[(H(a_i) + H(a_j)]$; thus, Eqn 2 can be rewritten as:

$$I_{ij} = \frac{I(a_i, a_j, C)}{H(a_i) + H(a_j)} \quad (3)$$

Similar to conventional two-way mutual information, the interaction gain satisfies the symmetry property, which means that it is not influenced by the ordering of the features involved. Unlike two-way mutual information, three-way mutual information can be positive, negative, or zero. A positive interaction gain value implies collaboration between two features. Such inter-feature collaboration indicates that the two features provide more information about the decision classes together than they do individually. The higher the positive value is, the stronger the collaboration. A negative interaction value implies that two features are redundant. In other words, the two features provide common information about the decision classes. A low negative value that tends towards -1 demonstrates high redundancy. A value of zero indicates that the inclusion of feature a_i (or a_j) has no impact on the relationship between a_j (or a_i) and C. That is, a_i and a_j provide information about C independently of one another.

3.2. Feature graph construction

In the initial stages of GBFG, a graph is constructed according to the distribution of the features and their relatedness. Each of the conditional features is represented by a node in the graph, and the relationships between the features are represented by the graph edges. Let $A = \{a_1, a_2, \cdots, a_{|A|}\}$ be the set of features in a dataset and C be the set of decision classes of the dataset. For any pair of features $a_i, a_j \in A$, the concept of a feature

matrix that represents the relatedness can be introduced by using the normalised interaction gain I_{ij} as shown in Eqn. (3) with $i, j \in \{1, 2, \cdots, |A|\}$ and $i \neq j$, which is computed from $I(a_i, a_j, C)$ between the features a_i and a_j per Eqn. (1). This establishes a link between a pair of features if a non-zero normalised interaction gain is calculated between them, and the link is weighted by the value of the resulting gain.

Table 1: Example of a feature relatedness matrix

Feature	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1	0	**0.25**	0	0.4	0	0.5	0.6	0
a_2		0	**0.15**	0	0	**0.1**	0.35	0
a_3			0	0.45	0.3	-0.6	0	0
a_4				0	**0.2**	**0.3**	-0.5	0
a_5					0	0.4	**0.15**	0
a_6						0	0	0
a_7							0	0
a_8								0

Table 1 shows an example of a graph matrix. Features $a_1 - a_7$ have weighted connections with the others, where any link that is associated with the gain value of '0' is omitted, as no interaction exists between such features. Note that feature a_8 does not interact with the others at all. In terms of three-way mutual information, a_8 independently provides information about the decision classes. The addition of such features neither improves the performance of the inter-feature collaboration nor demonstrates that they are redundant features with respect to the others. At this point, these fully isolated features are therefore disregarded at the feature clustering stage. Although the independent features may contain a certain amount of information regarding the decision classes, their intra-feature uncertainty always remains unresolved with the others. A graph constructed with this matrix is shown in Fig. 2. An increasing feature space could result in this kind of graph with more internal complexity. Thus, the direct use of such a graph for grouping tasks becomes intractable. The simplification of the graph into a kNN graph could be a solution. This may generate a loop, which in turn may lead to more complex feature grouping. An MST avoiding looping is thus used for grouping within the proposed approaches.

An MST is a sub-graph of a given graph that contains all the nodes

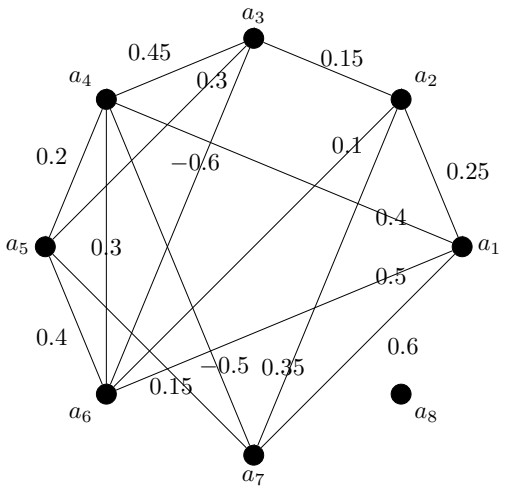


Figure 2: Graph constructed with the link weights given in Table 1

from the original super-graph such that the nodes are connected with the minimal total weighting on all links. To help reduce the otherwise massive computational effort necessary, a well-known adjacency list where an MST is built is introduced to represent the original graph. From the adjacency list obtained in the graph of Fig. 2, a possible MST can then be constructed as illustrated in Fig. 3. The MST is used in the proposed methods to produce feature groups, each of which is expected to cluster the most similar features together. Thus, only one feature is necessarily nominated from each feature group.

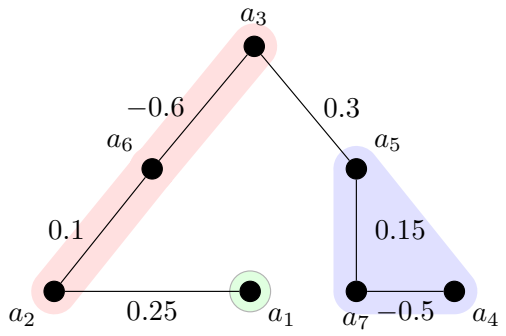


Figure 3: A possible MST derived from the adjacency list generated from the graph of Fig. 2

3.3. Grouping of features

The use of three-way mutual information reinforces the concepts of collaboration and redundancy. Features that are collaborative can provide more information about the decision than they can individually, and features that are redundant provide common information about the decision classes. The larger the positive value of three-way mutual information is, the stronger the collaborative contribution of the features. Similarly, the smaller the negative value of three-way mutual information is, the higher the level of redundancy between the features.

Reflecting the above observation, the feature grouping process proposed here consists of two steps:

- Generate a minimum spanning tree of the features involved in the given problem using Kruskal's algorithm [11].
- Cluster the features by iteratively removing edges from the resulting trees such that highly redundant (or weakly collaborative) features appear in the same groups.

In particular, the negative relationships of three-way mutual information can be used to build a feature MST to minimise redundancy. That is, to cluster highly redundant features together, edges with larger negative values in the resulting MST are removed iteratively, and features that are connected by edges with smaller negative values are formed into feature groups. If considered from the perspective of inter-feature collaboration, in building an MST, features that are deemed weakly collaborative are clustered into the same groups. This design intention reflects the underlying task of the present research, which is to perform feature subset selection where the choice of one feature from each of the resulting groups can readily construct and serve as the required feature subset.

More formally, let $A' \subseteq A$ be a set of nodes $\{a_1, a_2, \cdots, a_{|A'|}\}$ and E be the set of possible graph edges $\{< a_i, a_j > | a_i, a_j \in A', i \leq j\}$. A graph or an adjacency list can be generated for the features using the three-way mutual information measures. Then, an MST, T, that has $|A'|$ nodes and $|A'| - 1$ edges can be built from the adjacency list using Kruskal's algorithm. Note that such a resultant T may not be unique. However, the strategy of a first-come, first-served basis is used to help guarantee Kruskal's algorithm to generate a single T. The implementation of such a strategy for Kruskal's algorithm simply ranks the edges in ascending order in terms of their weights.

In an MST T, the removal of a certain number of edges will result in a forest F that contains the same number of subtrees as that of the removed edges plus one. The proposed approach to feature grouping works by iteratively removing a single edge from the graph such that the existing tree is divided into two new MST subtrees by deleting the link that has the maximum weight in the original tree. Every subtree then forms a feature group, and the resulting forest F forms a grouping of features. To judge the quality of this grouping, a selection of representative features are evaluated regarding their relevance to the decision classes. In particular, the popular probabilistic consistency [13] is herein adopted as the quality metric.

The probabilistic consistency measure calculates the discriminability of a given feature subset $S \subseteq A$ with respect to the given decision classes. For each feature $a_i \in S$ $(i = 1, 2, \cdots, |S|)$, suppose that a_i has $|V_{a_i}|$ distinctive values. A combination of the values, one for each different feature, forms a feature pattern, which is a part of a data instance without the decision class. The total number of patterns for S is the product of the value amount of all features in S, $\Pi_{i=1}^{|S|}|V_{a_i}|$. In practice, not all of the possible patterns have to be contained in a real-world dataset, and the consistency measure only needs to consider the patterns already existing in the dataset. For all the emerged patterns $\{N_S^j : j = 1, 2, \cdots, k\}$ of S, such a probabilistic consistency measure between S and a given set of decision classes C is mathematically defined by

$$f(S, C) = 1 - \sum_{j=1}^{k} \big(\sum_{c \in C} p(N_S^j|c)p(c) - \sup_{c \in C}(p(N_S^j, c)) \big) \tag{4}$$

where regarding all the instances in a given dataset, $\sum_{c \in C} p(N_S^j|c)p(c)$ is the marginal probability of the pattern N_S^j over all the decision classes c in C, calculated by the frequency of the instances containing N_S^j, and $p(N_S^j, c)$ is the joint probability between the pattern N_s^j and a given class c, calculated by the frequency of the instances that contain N_S^j and c at the same time. Instances containing the same pattern and the same decision class are considered to be consistent, and instances containing the same pattern but different decision classes are treated as inconsistent with each other. The term $\sup_{c \in C}(p(N_S^j, c))$ determines the most consistent instances for a specified pattern after taking into account all the given decision classes, and with this term, the remaining instances are determined to be inconsistent.

Recall the MST of Fig. 3, which is discovered from the graph given in Fig. 2 (with a_8 treated as an independent feature and hence ignored). From this, the feature grouping process can be illustrated as follows: Initially, forest F contains only a single tree $T = \{A', E\}$, where $A' = \{a_1, a_2, \cdots, a_7\}$ and $E = \{< a_2, a_6 >, < a_2, a_3 >, < a_5, a_7 >, < a_4, a_5 >, < a_1, a_2 >, < a_4, a_6 >\}$, with its elements weighted by {0.1, 0.15, 0.15, 0.2, 0.25, 0.3}, respectively. The proposed approach attempts to remove a single edge that iteratively takes the maximum weight in F. For this example, the edge $< a_4, a_6 >= 0.3$ is removed in the first instance, and the current tree is then divided into two subtrees, $T^1_{new} = \{\{a_4, a_5, a_7\}, \{< a_4, a_5 >, < a_5, a_7 >\}\}$ and $T^2_{new} = \{\{a_1, a_2, a_3, a_6\}, \{< a_2, a_3 >, < a_1, a_2 >, < a_2, a_6 >\}\}$. Thus, F now contains two members $\{T^1_{new}, T^2_{new}\}$ rather than the original single tree $\{T\}$. As such, a grouping of two feature groups $\{\{a_4, a_5, a_7\}, \{a_1, a_2, a_3, a_6\}\}$ is formed in place of the previous fully connected tree. Suppose that following the ranking of the features using the probabilistic consistency measure, two features a_4 and a_6 are selected as representatives of these two groups (one from each). This leads to a selected feature subset $\{a_4, a_6\}$. A better subsequent grouping will result in the continuation of the algorithm and otherwise, the algorithm is terminated. For illustration, suppose that the current grouping is better than the previous, then the edge $< a_1, a_2 >$ with the weight 0.25 is excluded from F. Three feature groups are therefore obtained at this stage: $\{a_1\}$, $\{a_2, a_3, a_6\}$, and $\{a_4, a_5, a_7\}$. The process of edge removal will cease and the algorithm will terminate if there is no improvement in the quality of the resulting grouping. Supposing that $|A|$ is the number of features for an input dataset, the complexity of the feature grouping process can be approximated to $O(|A|^2)$.

4. Feature Selection with GBFG

In this section, two feature selection techniques based on GBFG are proposed. The first method performs feature selection by a straightforward selection of representative features from the generated feature groups, where the feature subsets are assessed and selected following the choice of one representative feature from each group that is most collaborative with the other representative features chosen from the other groups. The second approach performs feature selection from the feature groups by employing a harmony search in an effort to discover multiple quality feature subsets in one pass.

4.1. GBFG-based feature selection with greedy hill climbing

In the proposed framework for GBFG, the evaluation of the selected representatives from feature groups is used to decide whether the current candidate feature grouping is of a better quality than the previous one. This iterative refinement step helps to automatically decide when to stop pruning edges from the MST. In this method, the process of selecting representatives out of a feature grouping can be directly viewed as a form of implementing FS. This selection process includes the iteration of the two steps listed as follows:

1. Select a pair of features from two groups (one from each) returned by GBFG that are the most collaborative with each other and include these features in the feature representative subset R.

2. For the remaining feature groups that do not contain features in R, search for one feature from each group that is the most collaborative with R regarding the decision classes and add this feature to R until all the groups have nominated a representative feature. The collaboration of a feature a_i of a given group and R is defined as follows:

$$Col(a_i, R) = \Sigma_{a_j \in R \wedge 0 \leq I(a_i, a_j, C)} I(a_i, a_j, C) \tag{5}$$

Algorithm 1 formalises the above procedure in which a desirable feature grouping is returned, and subsequently, one representative feature is selected from each group and is jointly returned with the other group representatives as the elements of the ultimately selected feature subset. Supposing that k groups result in the process of feature grouping, the worst searching complexity occurs when all k groups have the same number of members $|A|/k$. The complexity of this search strategy is then $O(|A|)$. The complexity of the entire GBFG-based FS algorithm is $O(|A|^3)$.

In particular, the loop in Alg. 1 is first used to weight the edges between any paired features in a constructed graph, where all conditional features are mapped as nodes. Edges with a large weight value indicate high collaboration between the features according to three-way mutual information. In line 6, Kruskal's algorithm is utilised to build an MST. Through the removal of the edges with the largest weight value (or one of those with the largest weight if there are more than one), feature groupings are generated in line 10. Feature groups in a grouping are consistent with the subtrees in the resulting

forest. As such, large-weighted edges are iteratively eliminated, the weakly collaborative features are then grouped in the same subtree. To evaluate the quality of the feature groupings, lines 11-13 in the second while loop in conjunction with lines 7-8 introduce the process of finding representatives, and a possible combination of features that are most collaborative with each other is produced for evaluation (instead of the evaluation of the feature groupings themselves) in line 14. In this implementation, the resulting representative features are evaluated using the probabilistic consistency measure $f(S, C)$, as defined in Section 3.3. For lines 15-17, if the representative features selected from the current grouping are better than those selected from the previous grouping, then the loop continues. Otherwise, the algorithm returns the currently best feature grouping, and the selected representatives are also returned as the FS outcome.

To continue the example used in Section 3.3, the grouping of two feature groups $\{\{a_4, a_5, a_7\}, \{a_1, a_2, a_3, a_6\}\}$ is obtained after the first iteration. Based on the relatedness matrix of Table 1, a pair of features a_7 and a_1 are deemed most collaborative with each other, so they are included in the set of (emerging) representative features R (which started as an empty set). Since a_7 and a_1 are in $\{a_4, a_5, a_7\}$ and $\{a_1, a_2, a_3, a_6\}$, respectively, $R = \{a_1, a_7\}$ is then used as the selected feature subset for the current grouping. If this feature subset is better than the previous subset, which was obtained in the same manner, the feature grouping process continues. Otherwise, the algorithm returns the selected feature subset as well as the current grouping.

In continuing the illustrative example, the evaluation of the selected subset is (correctly) assumed to be better than the previous (which was empty). Thus, the next grouping results with three groups: $\{a_1\}$, $\{a_2, a_3, a_6\}$, and $\{a_4, a_5, a_7\}$. As with the last iteration, a_1 and a_7 are included in R. For the group $\{a_2, a_3, a_6\}$ that has not yet nominated a feature, we compute the collaboration of each feature of $\{a_2, a_3, a_6\}$ and R based on Eqn. (5). The value of the collaboration of a_2 and R is $0.25 + 0.35 = 0.6$, the value of that of a_3 and R is $0 + 0 = 0$, and the value of that of a_6 and R is $0.5 + 0 = 0.5$. As a_2 has the largest collaboration value, a_2 is then included in R. Thus, the feature subset $\{a_1, a_2, a_7\}$ is selected. If the quality of this subset is no better than that of the current best, then $R = \{a_1, a_7\}$ is returned as the final feature subset.

4.2. GBFG-based feature selection with harmony search

Although the previous approach has an advantage in runtime cost, better feature combinations may exist between feature groups. This is where a metaheuristic approach may help strengthen the work. Selecting an 'optimal' feature subset from groups of features is a combinatorially difficult problem. An exhaustive search can guarantee that the best feature subsets are discovered, but this is often computationally impractical for real-world applications. A harmony search (HS) [26] may be useful for the task of selecting features from a particular feature grouping and can potentially be used as a meta-heuristic approach to set up the initial solution pool. More significantly, an HS can help identify multiple quality feature subsets by completing a single search process. This observation has inspired the following development.

An HS is proposed as a meta-heuristic search algorithm that mimics the improvisation process of instrument players, primarily for discrete variables. Each musician represents a system variable (not a feature from FS perspective) that characterises the objective function, playing a note (or taking a value) to construct a harmony (solution) together with the rest to optimise this function. Newly generated harmonies iteratively progress based on the musicians' experience (a pool of existing harmonies) and are used to update historical solutions with respect to harmony quality. The HS algorithm consists of the following key steps:

The original harmony search-based FS approach, as represented in [26], can be extended to take advantage of the GBFG framework. In particular, musicians are mapped onto feature selectors, the number of which is equal to that of the features in a given dataset. Suppose that a dataset has $|A|$ features, and each selector then has $|A|$ choices. This means that the space for the harmony search is $|A|^{|A|}$ in theory. Here, the harmony search is employed to effectively reduce the size of the search space and thus the computational effort. In contrast to the original work of [26], where feature selectors are assigned for each feature, here, a single feature selector is assigned to each grouping of features. For example, if k groups are derived from $|A|$ features by establishing homogeneous feature groups, the search space is then reduced to $|A|^k$ (where typically $k \ll |A|$), while a single selector still has $|A|$ choices. Additionally, the original approach, as reported in [26], treats the FS as a bi-objective optimisation problem, while the new algorithm turns the FS into a single objective optimisation problem. This is because the feature subset size is a direct derivation of GBFG, and there is no longer a need to consider the

evaluation of this search parameter. The procedure of the GBFG-based HS algorithm is summarised in Algorithm 2, and the steps are listed as follows:

Step 1: Initialise parameters: There are six key parameters in the HS algorithms: HMS, HMCR, PAR, λ, BW and M. HMS is the number of (potential) solutions stored in the harmony memory (HM), which is a two-dimensional matrix where each row indicates a solution and each column is dedicated to a single musician, storing the musician's experience. HMCR and PAR control the global and local searches of the HS algorithm, respectively. In a typical implementation, both take values ranging from 0 to 1. The threshold λ acts as the stopping criterion, with the search process terminating when the number of iterations reaches λ. Note, however, that BW is no longer employed since in FS, each feature is an independent granule [14]. The neighbouring features cannot be computed simply using the standard arithmetic operators and random numbers. Instead, the feature similarity measure proposed in [26] is used here to identify any neighbouring features given another. As the number of musicians is assigned to the number of feature groups obtained using the GBFG approach (see Section 3.3), M is set to the number of groups k rather than that of features $|A|$, which is the natural setting of the original HS (fortunately, $k \ll |A|$).

Step 2: Initialise HM: HM is now a two-dimensional matrix with a size of HMS $\times\, k$. Each row stores a feature subset, while each column stores HMS historical features, taking the form below:

$$\mathrm{HM} = \left[\begin{array}{cccc|c} a_1^1 & a_2^1 & \cdots & a_k^1 & f(S^1, C) \\ a_1^2 & a_2^2 & \cdots & a_k^2 & f(S^2, C) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1^{\mathrm{HMS}} & a_2^{\mathrm{HMS}} & \cdots & a_k^{\mathrm{HMS}} & f(S^{\mathrm{HMS}}, C) \end{array}\right]$$

where a_i^j is a feature selected by the i^{th} feature selector in the j^{th} harmony and $f(S^j, C)$ is the evaluation function used to calculate the relevance of a given feature subset S^j to C, which denotes a set of decision classes. As there may exist duplicated features in a harmony, the size of a feature subset may be smaller than that of the corresponding harmony.

Step 3: Improvise new feature subsets: A new harmony $H' = (a'_1, a'_2, \cdots, a'_k)$ is generated based on the same two key factors as those of the original HS: the HMCR and a random value r $(0 < r < 1)$. A new feature a'_i is selected

by the i^{th} feature selector with respect to the following rules:

$$a_i' = \begin{cases} a_i' \in f_i & \text{if HMCR} \leq r \\ a_i' \in \{a_i^1, a_i^2, \cdots, a_i^{\text{HMS}}\} & \text{if HMCR} > r \end{cases} \tag{6}$$

where f_i is a homogeneous group of the features that are only utilised by the i^{th} feature selector. When the condition HMCR $\leq r$ is satisfied, the feature selector randomly selects a feature from its own homogeneous group. Otherwise, the feature selector randomly selects a feature from the historical feature pool. As shown in the graph of Fig. 2, constructed using three-way mutual information, the neighbours of a feature are those that are topologically connected. The link of the smallest weight determines the closest neighbour of a feature in terms of all possible links regardless of whether the link is negatively weighted or positively weighted.

Step 4: Update the HM: The quality of each newly produced harmony is computed using the probabilistic consistency measure $f(S, C)$ after the conversion of the harmony to a feature subset. Regarding the evaluated quality, the update of HM is then denied if no existing feature subsets worse than the new feature subset are found. Otherwise, the HM is updated on the basis of the following rule:

$$\mathbf{H}' \in \text{HM} \wedge \mathbf{H}^{worst} \notin \text{HM} \tag{7}$$

Step 5: Check the stopping criterion: If the number of improvisations reaches λ, the algorithm stops. Otherwise, repeat step 3 and step 4. The complexity of this HS-based FS algorithm is $O(\lambda * |A|^3)$ for the worst case.

Algorithm 1: GBFG-FS: Direct GBFG-based FS

input : A: set of features and C: set of decision classes
output: $S \leftarrow \emptyset$: feature subset and $F \leftarrow \emptyset$: feature grouping
// Step 1: construct a graph using features by calculating the three-way mutual information amongst them
$G \leftarrow \{V, E\}$: undirected graph, where $V \leftarrow A$ and $E \leftarrow \emptyset$
while $a_i, a_j \in V$ **do**
    if $< a_i, a_j > \notin E$ **then**
        $Weight(< a_i, a_j >) \leftarrow \dfrac{I(a_i, a_j, C)}{H(A_i) + H(A_j)}$
        $E \leftarrow E \cup \{< a_i, a_j >\}$
// Step 2: build an MST, T over G using Kruskal's algorithm
$T \leftarrow Kruskal(G)$, which has nodes V' and edges E'
// Step 3: iteratively generate feature groupings and select representative features
while *True* **do**
    Remove edge(s) with the maximum weight in E', and break T into a forest $\{V', E'\}$, indicating a feature grouping
    $F \leftarrow \{V', E'\}$
    $R \leftarrow \emptyset$: set of representative features
    Select one largest edge from E, and include nodes connected by the selected edge in R
    for *Each group* $f' \in F$ *&&* $f' \cap R == \emptyset$ **do**
        $S \leftarrow R$
        foreach $a' \in f'$ **do**
            if $Col(a', R)$ *is the largest* **then**
                $Temp \leftarrow R \cup \{a'\}$
    Evaluate the selected representatives $Temp$ as the quality of the grouping F
    if $f(Temp, C) > f(R, C)$ **then**
        $R \leftarrow Temp$
    else
        $S \leftarrow R$
        return S *and* F

Algorithm 2: GBFG-HS: FS with harmony search via GBFG

F: set of feature groups from graph-based grouping
C: set of decision classes
S: returned (conditional) feature subset
H: harmony memory (2-dimensional matrix)
$H_{i,}$, $H_{,j}$, and $H_{i,j}$: row, column, and cell of H
$H_{new,}$, $H_{w,}$, and $H_{b,}$: newly improvised harmony, worst harmony and best harmony in H, respectively
HMS: number of rows in H
k: number of columns in H and number of groups

$F \leftarrow \{f_1, f_2, \cdots, f_k\}$
$S \leftarrow \emptyset$
// Initialisation Phase
for $i \leftarrow 1$ **to** *HMS* **do**
    for $j \leftarrow 1$ **to** k **do**
        $H_{i,j} \leftarrow$ Random(f_j)
    end
end
// Iteration Phase
while Λ *is not satisfied* **do**
    for $j \leftarrow 1$ **to** k **do**
        if Random(*[0,1]*) $<$ *HMCR* **then**
            $t \leftarrow$ Random(f_j)
        else
            $t \leftarrow$ Random($H_{,j}$)
        end
        $H_{new,j} \leftarrow t$
    end
    if $f(H_{new,}, C) > f(H_{w,}, C)$ **then**
        $H_{w,} \leftarrow H_{new,}$
    end
end
$S \leftarrow H_{b,}$
return S

5. Experimental Evaluation

In this section, a series of experiments are discussed using 20 different UCI-MLR benchmark datasets [3]. The experimental studies include 1) a comparison with popular FS approaches that are based on an existing feature grouping method (FRFG) [9] or on metaheuristic searches or greedy hill climbing, including the genetic algorithm (GA-FS) [6], particle swarm optimisation (PSO-FS) [5], and greedy hill climbing (GHC-FS) [28]; and 2) a comparison with the harmony search-based FS approach (HSFS) [26].

5.1. Experimental Setup

Ten stratified randomisations of 10-fold cross-validation [22] are employed in the generation of the experimental results. FS is performed as part of the cross-validation, and each fold results in a new selection of features. Three different aspects of performance are examined in the evaluation: the classification accuracy, final selected subset size, and average runtime per cross-validation fold. A paired t-test ($p = 0.05$) is used to examine the statistical significance of the generated results of both the classification accuracy and subset size.

The datasets used for the experimental evaluation range in size from 120-5000 instances and from 10-2557 features. Most of the data have 2-7 decision classes, but a number of them have 10 to 19. A summary of the datasets is shown in Table 2, where these datasets are arranged in ascending order by the number of decision classes. As stated previously, all the datasets are drawn from [3]; they are selected to facilitate comparative studies since they have been used by the other algorithms that are compared against here.

The parametric settings in the compared methods are those typically used by the original approaches in the literature. In particular, the GA-based FS method has an initial population size of 20, a maximum number of generations of 5000, crossover probability of 0.6 and mutation probability of 0.033. The number of generations for the PSO search is set to 5000, while the number of particles is set to 20, with acceleration constants c1 = 2 and c2 = 2. For the harmony search approaches (both HSFS [26] and GBFG-HS), the maximum number of iterations is set to 5000, harmony-memory consideration rate to 0.7, pitch-adjustment rate to 0.8, and harmony memory size to 20. These parameters may not be ideal for all of the datasets employed here, and an optimisation phase may well result in an improvement in performance. However, such a parameter optimisation would need to be

performed on a dataset-by-dataset basis, which would involve a significant investment of computational effort (and which may involve unfair settings for the comparisons) and therefore is not adopted here.

For consistency and fair comparison within all the experiments concerning FS, the probabilistic consistency measure [13] is used to evaluate the feature subsets. Other subset-based evaluation functions may also be applicable, such as correlation [28], correlation coefficient [17], message passing [21], and fuzzy rough dependency [2], but this is beyond the scope of this paper. Note that the numbers in brackets indicate the standard deviation (SD) in all the tables hereafter. The time unit is milliseconds (MS) for gauging the runtime of the algorithms. A figure in bold signifies a statistically better result in the comparison of the experimental results of the top two methods.

Table 2: Summary of datasets

Dataset	Features	Instances	Classes
breastcancer	10	286	2
heart	14	270	2
vote	17	435	2
ionosphere	35	230	2
credit-g	21	1000	2
water2	39	390	2
sonar	61	208	2
ozone	73	2534	2
secom	591	1567	2
water3	39	390	3
waveform	41	5000	3
olitos	26	120	4
cleveland	14	297	5
web	2557	149	5
glass	10	214	6
segment	20	1500	7
multifeat	650	2000	10
libras	91	360	15
arrhythmia	280	452	16
soybean	36	683	19

5.2. Comparison with popular FS methods

The two implementations (GBFG-FS and GBFG-HS) developed in this work are evaluated in this section by comparison with several popular existing FS methods that are readily available. FRFG [9] is an existing feature

grouping-based FS technique. GA-FS, PSO-FS and HSFS use metaheuristic strategies, while GHC-FS employs a greedy search. In FRFG, the features are grouped by clustering redundant features, which is defined as the degree of correlation between features exceeding a predefined threshold β that can take values from 0 to 1. The best results tend to be obtained when β is set to a value between 0.8 and 0.9; therefore, two sets of results are presented as one by taking their average. The classification accuracies achieved by the two algorithms proposed in this work and the existing FS methods are presented in Tables 3-6, the sizes of the reduced feature subsets (which reflects the number of feature groups) for these algorithms are reported in Table 7, and the runtimes produced by these FS methods are shown in Table 8.

The results presented in Tables 3-6 are the classification accuracies that are attained by four classic learning classifiers. They are JRip, a rule-based classifier; J48, a decision-tree learner; IBk, a nearest-neighbour classifier (with $k = 3$), and their ensemble with the voting strategy [16]. For the different classifiers used in this experiment, although all three classifiers return slightly different results for the same dataset, there are no statistically significant differences among them. When compared with the other approaches regarding the JRip classifier, the two proposed algorithms together have a win ratio of 40%. The remainder are all statistically comparable to the best of other methods, and their difference values fall in the range $[0, 2)$. Regarding classification with J48, overall, the proposed GBFG-based algorithms outperform the others for 55% of the datasets. Especially for the datasets *ionosphere* and *web*, GBFG-HS is the only algorithm that achieves statistically better performance, consistently beating all the rest. Regarding the use of the IBk classifier, similar overall results can be seen across the use of feature subsets returned by different FS methods. However, when the ensemble classifier is used, the proposed algorithms have the best achievement for the win ratio of 60%. Obviously, whichever classifier is used, the proposed methods are the overall winners regarding the average of the classification accuracy on 20 datasets.

Table 7 presents the results in terms of the selected feature subset size. The statistically smallest size was achieved by GBFG-HS for 90% of the datasets except for *multifeat* and *waveform*, on which GBFG-HS remains comparable to GHS-FS and GA-FS, respectively, regarding the feature reduction capability. GBFG-FS also performs better at reducing the features on most datasets than GA-FS, PSO-FS, and FRFG-based FS. These results indicate that the graph-based approach is effective at identifying compact

representatives.

Compared with the GBFG-FS algorithm, GBFG-HS not only achieves better classification accuracy but also further reduces the feature subset size on over 30% of the datasets for any classifiers used. GBFG-FS, however, offers a significant reduction in the runtime, which is clear from Table 8. It is also the most efficient among all the compared FS methods and across all the datasets, with the exception of *web*. For the dataset *web*, GA-FS offers a very good runtime, but its corresponding subset size is far larger than those of the other FS methods (more than 140 times larger than that achieved using GBFG-HS, for instance). Although the runtime of GBFG-HS is not as good as that of GBFG-FS, it is more efficient than the other FS methods except for GHC-FS when dealing with large datasets such as *arrhythmia*, *secom*, and *multifeat*. Overall, GBFG-HS is a clear winner.

Table 3: Comparison with other FS methods: average classification accuracies (%(SD)) for the JRip classifier, where bold indicates the statistically best value

	Unred.	GBFG-HS	GBFG-FS	FRFG	GA-FS	PSO-FS	GHC-FS	HSFS
breastcancer	71.37 ± 6.62	**71.37 ± 6.63**	70.93 ± 5.90	70.00 ± 6.59	70.96 ± 6.96	71.24 ± 6.58	69.90 ± 6.53	69.02 ± 6.88
heart	78.63 ± 7.37	78.89 ± 7.21	76.63 ± 8.61	77.73 ± 7.04	79.37 ± 6.56	**79.41 ± 6.58**	77.23 ± 6.89	77.61 ± 6.86
vote	95.61 ± 2.79	**95.64 ± 3.94**	95.42 ± 3.08	93.70 ± 2.76	95.47 ± 2.68	95.40 ± 2.77	93.70 ± 2.73	93.61 ± 2.87
ionosphere	86.78 ± 7.43	**86.09 ± 6.74**	84.52 ± 7.75	84.92 ± 7.63	85.04 ± 6.58	83.74 ± 7.76	85.30 ± 7.59	83.21 ± 7.48
credit-g	71.92 ± 3.65	70.60 ± 5.95	72.25 ± 3.89	69.95 ± 4.19	71.74 ± 3.83	**72.41 ± 3.99**	71.04 ± 4.40	70.32 ± 4.10
water3	82.26 ± 6.80	82.56 ± 6.26	**84.15 ± 5.65**	81.44 ± 5.16	82.97 ± 5.95	82.56 ± 5.80	81.71 ± 5.41	82.05 ± 6.10
sonar	75.06 ± 8.64	71.64 ± 9.92	72.83 ± 9.97	72.78 ± 9.80	71.51 ± 9.40	**74.93 ± 9.82**	73.43 ± 9.82	73.22 ± 10.48
ozone	93.13 ± 1.33	**93.33 ± 1.12**	93.13 ± 1.27	91.35 ± 1.14	93.19 ± 1.02	92.83 ± 1.18	91.42 ± 1.21	91.45 ± 1.13
secom	92.52 ± 1.03	**93.36 ± 0.32**	93.06 ± 0.76	89.94 ± 1.30	92.69 ± 1.00	92.51 ± 1.21	91.34 ± 0.57	90.63 ± 1.11
water4	82.10 ± 4.81	82.31 ± 6.78	80.95 ± 6.06	80.83 ± 5.93	**82.74 ± 6.8**	81.54 ± 6.13	80.91 ± 6.69	80.51 ± 5.68
waveform	79.14 ± 1.70	76.64 ± 2.08	**78.37 ± 2.10**	76.70 ± 1.98	76.88 ± 2.07	75.66 ± 2.20	76.53 ± 1.92	75.80 ± 2.21
olitos	68.25 ± 11.29	65.00 ± 12.91	65.50 ± 12.54	65.38 ± 13.12	64.67 ± 12.09	64.42 ± 14.26	**68.03 ± 12.20**	64.36 ± 11.80
cleveland	54.08 ± 3.36	53.18 ± 2.68	53.78 ± 3.65	53.16 ± 3.31	55.23 ± 3.31	**55.53 ± 3.81**	54.03 ± 3.21	53.69 ± 3.07
web	55.57 ± 13.23	**57.64 ± 11.16**	56.20 ± 11.24	53.28 ± 12.80	55.16 ± 11.17	51.39 ± 12.27	55.05 ± 12.19	53.19 ± 10.59
glass	68.19 ± 10.23	66.41 ± 7.89	66.43 ± 9.26	65.84 ± 9.12	**66.85 ± 9.17**	65.51 ± 9.03	65.55 ± 9.03	64.66 ± 9.79
segment	93.23 ± 2.08	92.47 ± 1.30	92.16 ± 3.47	91.96 ± 1.71	93.15 ± 2.45	**93.59 ± 2.01**	92.06 ± 1.91	91.29 ± 2.42
multifeat	92.17 ± 1.84	86.90 ± 2.88	72.46 ± 6.18	85.55 ± 4.01	82.32 ± 2.84	81.53 ± 2.94	86.28 ± 2.57	**89.56 ± 2.09**
libras	54.61 ± 9.80	51.67 ± 12.51	49.67 ± 9.01	52.55 ± 8.40	54.14 ± 8.69	**54.08 ± 7.99**	51.88 ± 8.05	52.98 ± 8.67
arrhythmia	70.55 ± 5.42	69.69 ± 6.86	65.77 ± 6.77	68.93 ± 6.11	69.47 ± 5.72	70.38 ± 5.97	**69.71 ± 5.41**	68.07 ± 6.25
soybean	91.88 ± 3.03	79.13 ± 5.10	74.49 ± 5.17	**82.59 ± 6.10**	69.57 ± 5.97	81.05 ± 5.82	78.40 ± 3.49	78.50 ± 6.06
Avg.	77.85 ± 5.62	**76.23 ± 6.01**	74.94 ± 6.12	75.43 ± 5.91	75.66 ± 5.72	75.99 ± 5.91	75.68 ± 5.59	75.19 ± 5.78
Win Ratio		6/20	2/20	1/20	2/20	6/20	2/20	1/20

To further illustrate the potential of GBFG-HS, it is important to demonstrate that it is an improvement over the improved counterpart of the original harmony search-based FS (HSFS) method (which uses the same search strategy) [26]. This section presents such a comparative analysis. In terms of the classification accuracy presented in Tables 3-6, GBFG-HS offers results comparable to those of HSFS. However, when the results of the respective approaches are considered in terms of the subset size with similar classification accuracies, the advantage of employing GBFG-HS becomes clear. Of the 20 datasets, GBFG-HS offers reductions that are impressive and statistically

Table 4: Comparison with other FS methods: average classification accuracies(%(SD)) for the J48 classifier, where bold indicates the statistically best value

	Unred.	GBFG-HS	GBFG-FS	FRFG	GA-FS	PSO-FS	GHC-FS	HSFS
breastcancer	74.28 ± 6.02	**73.40 ± 7.72**	72.96 ± 6.01	71.64 ± 6.20	70.40 ± 6.26	70.12 ± 6.29	71.03 ± 6.32	68.85 ± 6.34
heart	78.15 ± 7.38	**82.22 ± 5.47**	78.04 ± 8.52	77.29 ± 7.31	79.19 ± 7.35	79.22 ± 7.33	77.23 ± 7.54	77.53 ± 7.36
vote	96.57 ± 2.55	95.87 ± 3.54	96.27 ± 2.94	94.55 ± 2.57	96.55 ± 2.62	**96.60 ± 2.58**	94.50 ± 2.60	94.50 ± 2.70
ionosphere	86.13 ± 6.17	**88.26 ± 5.81**	85.04 ± 7.78	85.90 ± 7.74	86.13 ± 6.47	84.74 ± 7.19	87.18 ± 6.46	83.43 ± 7.10
credit-g	71.25 ± 3.15	**72.80 ± 4.29**	72.20 ± 3.43	70.24 ± 3.29	72.03 ± 3.26	72.47 ± 3.56	70.98 ± 3.36	70.48 ± 3.35
water3	81.59 ± 6.48	**84.62 ± 6.51**	83.03 ± 4.75	82.18 ± 5.32	83.79 ± 5.40	82.82 ± 5.07	82.00 ± 4.98	81.64 ± 5.45
sonar	73.61 ± 9.30	69.17 ± 11.70	74.20 ± 10.13	72.72 ± 9.60	73.86 ± 9.90	**75.68 ± 9.82**	74.17 ± 9.82	72.84 ± 10.19
ozone	92.48 ± 1.34	93.21 ± 0.67	**93.27 ± 1.06**	91.22 ± 1.11	93.17 ± 1.17	92.59 ± 1.24	91.24 ± 1.08	91.23 ± 1.14
secom	89.49 ± 1.97	**93.30 ± 0.33**	93.06 ± 0.87	90.04 ± 1.20	91.10 ± 1.72	90.01 ± 2.03	91.30 ± 0.84	88.19 ± 1.76
water4	83.18 ± 5.47	79.49 ± 3.63	80.23 ± 6.59	80.52 ± 6.20	81.20 ± 6.16	**81.44 ± 6.18**	79.51 ± 5.70	79.33 ± 6.70
waveform	75.25 ± 1.89	75.00 ± 1.83	**76.16 ± 1.87**	74.79 ± 2.00	74.95 ± 2.28	73.12 ± 2.41	74.57 ± 2.03	73.57 ± 2.05
olitos	65.75 ± 12.07	62.50 ± 13.75	**65.25 ± 11.85**	63.09 ± 12.73	62.17 ± 13.37	63.08 ± 13.41	63.87 ± 11.81	62.89 ± 12.11
cleveland	53.39 ± 7.28	53.23 ± 8.17	**55.40 ± 6.61**	52.81 ± 7.29	55.13 ± 6.75	55.11 ± 6.93	53.80 ± 6.49	53.97 ± 7.23
web	57.63 ± 11.25	**56.38±11.89**	55.98 ± 12.11	52.55 ± 12.62	55.40 ± 12.87	53.21 ± 14.45	54.59 ± 11.78	54.35 ± 11.00
glass	68.08 ± 9.24	68.14 ± 7.99	69.78 ± 8.70	68.35 ± 9.17	**69.84 ± 9.26**	68.50 ± 9.15	68.26 ± 9.23	68.62 ± 9.37
segment	95.71 ± 1.84	95.20 ± 1.63	94.06 ± 3.23	94.05 ± 1.78	**95.49 ± 1.84**	95.38 ± 1.88	93.92 ± 1.76	93.59 ± 1.89
multifeat	94.62 ± 1.68	86.75 ± 3.59	75.30 ± 5.52	87.69 ± 5.00	84.72 ± 2.24	83.88 ± 2.54	**87.91 ± 2.17**	91.75 ± 1.74
libras	69.36 ± 8.34	65.83 ± 7.75	62.28 ± 8.71	**66.89 ± 8.26**	66.33 ± 8.44	66.67 ± 6.96	62.72 ± 8.01	65.49 ± 7.07
arrhythmia	65.78 ± 5.75	64.39 ± 5.43	61.58 ± 7.28	66.09 ± 6.54	**66.59 ± 5.67**	66.46 ± 5.83	65.23 ± 5.83	64.33 ± 6.41
soybean	91.78 ± 3.17	80.10 ± 4.87	78.51 ± 4.96	**83.73 ± 5.24**	69.46 ± 6.52	82.45 ± 5.43	81.52 ± 3.95	81.46 ± 5.05
Avg.	78.20 ± 5.62	**76.99 ± 5.83**	76.13 ± 6.15	76.32 ± 6.06	76.38 ± 5.98	76.68 ± 6.01	76.28 ± 5.59	75.90 ± 5.80
Win Ratio		7/20	4/20	2/20	3/20	3/20	1/20	0

better than those of HSFS for all datasets. For example, GBFG-HS offers reductions of 99.6% and 94% for the *web* and *arrhythmia* datasets over the results returned by HSFS, respectively. This illustrates that significant improvements in performance are achieved by GBFG-HS compared with HSFS, as GBFG-HS is capable of discovering compact and robust feature subsets. When using the runtime for the algorithm estimations, as seen from Table 8, with the exception of the *web* and *waveform* datasets, GBFG-HS offers considerably faster performance. For the two exceptions noticed, the relatively higher runtimes may be related to the high dimensionality of the datasets, which necessitate a long time for feature grouping. Therefore, the more efficient algorithm of feature grouping remains desirable, but GBFG-HS performs better than the HSFS algorithm [26] in terms of the size of the selected feature subsets. More interestingly, *web* has 2557 dimensions but only takes 149 samples. The distribution of the small size of samples in such high dimensionality means this could be considered a sparse dataset, which may be the main cause of the singularity of GBFG-HS.

5.3. Comparison between GBFG-HS and HSFS regarding iterations

To gain more useful insight into the behaviour of both GBFG-HS and HSFS, as the search for feature subsets progresses, an investigation is further conducted on two of the relatively more complex datasets: *arrhythmia* and *multifeat*. In Fig. 4, four plots are shown in each of the two sub-figures: the

Table 5: Comparison with other FS methods: average classification accuracies (%(SD)) for the IBk (k=3) classifier, where bold indicates the statistically best value

	Unred.	GBFG-HS	GBFG-FS	FRFG	GA-FS	PSO-FS	GHC-FS	HSFS
breastcancer	73.13 ± 5.51	70.62 ± 5.87	**72.25 ± 5.94**	71.36 ± 5.52	71.27 ± 5.86	71.20 ± 5.85	70.81 ± 5.59	70.08 ± 5.57
heart	79.11 ± 6.74	**81.85 ± 6.64**	77.33 ± 8.16	78.00 ± 7.30	79.48 ± 7.76	79.48 ± 7.76	77.17 ± 8.03	78.00 ± 7.82
vote	93.08 ± 3.68	**95.64 ± 3.13**	94.51 ± 3.67	91.62 ± 3.50	94.28 ± 3.22	94.48 ± 3.70	92.17 ± 3.57	92.26 ± 3.35
ionosphere	82.74 ± 5.72	**86.52 ± 4.32**	85.70 ± 7.50	83.01 ± 7.40	83.04 ± 6.58	82.87 ± 7.74	83.34 ± 6.61	79.85 ± 6.70
credit-g	72.21 ± 3.24	71.90 ± 4.25	72.17 ± 3.47	69.85 ± 3.32	71.74 ± 3.64	**72.38 ± 3.16**	70.56 ± 3.75	70.56 ± 3.50
water3	82.28 ± 4.47	83.08 ± 4.05	84.46 ± 5.14	85.08 ± 5.36	**86.95 ± 5.19**	85.26 ± 4.56	85.56 ± 5.04	84.65 ± 4.73
sonar	83.76 ± 8.46	81.31 ± 7.82	81.13 ± 9.52	80.22 ± 8.50	80.98 ± 9.09	**82.59 ± 8.42**	80.94 ± 8.42	82.42 ± 7.44
ozone	93.71 ± 0.92	93.25 ± 1.01	93.45 ± 1.01	91.83 ± 0.96	**93.50 ± 1.09**	93.63 ± 0.88	91.69 ± 1.01	91.97 ± 0.98
secom	92.72 ± 0.74	92.47 ± 1.07	92.29 ± 1.63	91.41 ± 1.27	92.25 ± 1.09	**92.72 ± 0.82**	90.88 ± 0.88	90.78 ± 1.10
water4	84.82 ± 4.48	81.79 ± 3.72	80.23 ± 5.63	83.20 ± 4.84	**84.87 ± 4.94**	82.77 ± 4.51	83.07 ± 4.65	81.74 ± 5.08
waveform	77.67 ± 1.78	75.98 ± 2.72	**79.49 ± 2.40**	77.21 ± 2.09	76.71 ± 2.41	74.56 ± 2.56	76.48 ± 2.14	75.29 ± 2.54
olitos	81.25 ± 9.08	70.00 ± 17.66	76.42 ± 11.11	75.42 ± 10.62	74.00 ± 10.68	**76.83 ± 9.52**	74.40 ± 11.90	75.95 ± 10.29
cleveland	55.70 ± 6.35	53.21 ± 7.21	53.92 ± 6.65	53.94 ± 6.47	**54.18 ± 6.20**	54.12 ± 6.96	52.70 ± 5.88	53.26 ± 6.34
web	37.97 ± 4.31	**58.12 ± 8.26**	57.28 ± 8.46	42.67 ± 10.41	38.43 ± 7.06	39.71 ± 8.93	55.71 ± 13.00	38.21 ± 7.08
glass	69.84 ± 8.57	**75.69 ± 10.21**	71.60 ± 9.01	72.11 ± 9.87	72.92 ± 9.86	70.88 ± 9.83	71.46 ± 9.98	71.55 ± 10.01
segment	94.95 ± 1.67	94.67 ± 1.54	94.49 ± 3.62	93.67 ± 1.78	94.80 ± 1.80	**94.95 ± 1.87**	93.04 ± 1.63	93.04 ± 1.90
multifeat	97.97 ± 0.94	91.40 ± 2.94	82.41 ± 4.95	92.78 ± 2.29	92.40 ± 1.70	88.66 ± 2.35	91.49 ± 1.86	**95.76 ± 1.02**
libras	80.67 ± 5.62	76.39 ± 6.31	71.58 ± 7.48	75.73 ± 6.62	75.86 ± 6.28	**79.11 ± 5.65**	74.78 ± 6.26	77.28 ± 5.44
arrhythmia	58.37 ± 3.75	**63.51 ± 5.32**	62.37 ± 6.00	60.02 ± 4.74	60.70 ± 5.18	59.60 ± 4.34	60.16 ± 4.49	59.04 ± 4.45
soybean	91.20 ± 3.16	73.93 ± 4.68	76.61 ± 4.86	**80.74 ± 6.04**	66.97 ± 5.38	79.87 ± 5.09	76.85 ± 4.01	77.77 ± 5.33
Avg.	79.16 ± 4.46	78.57 ± 5.44 (*)	77.98 ± 5.81	77.49 ± 5.44	77.27 ± 5.25	77.78 ± 5.23	77.66 ± 5.44	76.97 ± 5.03
Win Ratio		6/20	2/20	1/20	4/20	6/20	0	1/20

dashed lines represent the subset size, the dotted lines depict the subset evaluation results that are computed using the probabilistic consistency measure (described in Section 3.3), and the solid lines indicate the classification accuracy. All of these are plotted with respect to the total number of iterations used for the evaluation (5,000) at an interval of 500.

The observed trend shows a logarithmic increase in the evaluation score coupled with a stable and very low subset size for GBFG-HS from the outset. For *arrhythmia*, the trend is even more pronounced. Statistically, there are no significant differences in the resulting classification accuracies, as the outcomes are essentially statistically comparable. However, there is a large difference between HSFS and GBFG-HS in terms of the trend of subset size (which is obvious from the outset), indicating that whilst GBFG-HS may sometimes not score with respect to absolute classification accuracy, it always results in very compact and representative feature subsets.

Table 6: Comparison with other FS methods: average classification accuracies (%(SD)) for the ensemble classifier, where bold indicates the statistically best value

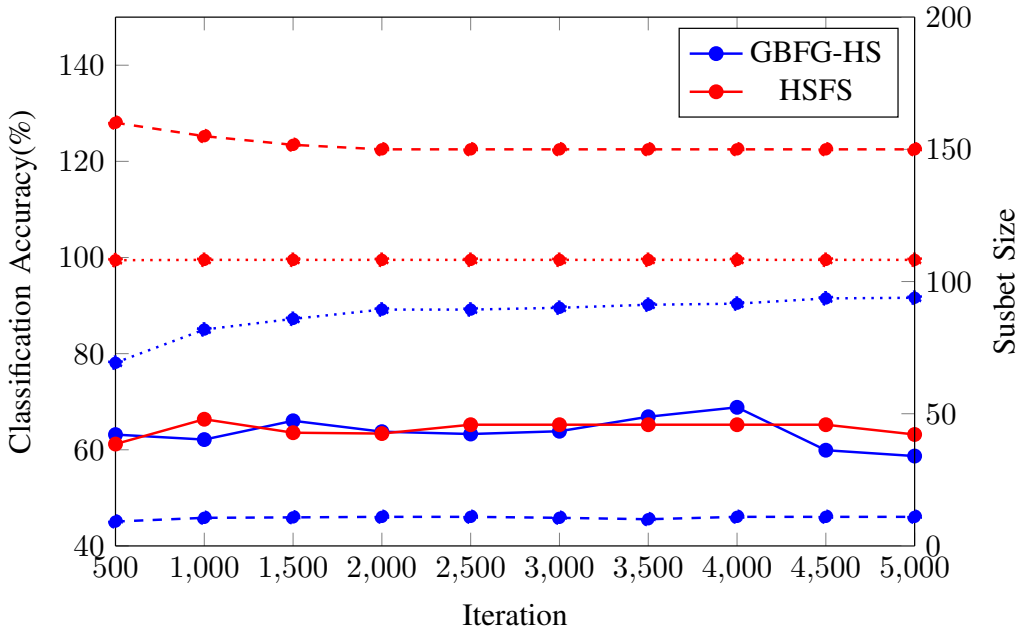
	Unred.	GBFG-HS	GBFG-FS	FRFG	GA-FS	PSO-FS	GHC-FS	HSFS
breastcancer	72.93 ± 6.05	71.80 ± 6.74	**72.05 ± 5.95**	71.00 ± 6.10	70.88 ± 6.36	70.85 ± 6.24	70.58 ± 6.15	69.32 ± 6.26
heart	78.63 ± 7.16	**80.99 ± 6.44**	77.33 ± 8.43	77.67 ± 7.21	79.35 ± 7.22	79.37 ± 7.22	77.21 ± 7.49	77.71 ± 7.35
vote	95.09 ± 3.01	**95.72 ± 3.54**	95.40 ± 3.23	93.29 ± 2.94	95.43 ± 2.84	95.49 ± 3.02	93.46 ± 2.97	93.46 ± 2.97
ionosphere	85.22 ± 6.44	**86.96 ± 5.62**	85.09 ± 7.68	84.61 ± 7.59	84.74 ± 6.54	83.78 ± 7.56	85.27 ± 6.89	82.16 ± 7.09
credit-g	71.79 ± 3.35	71.77 ± 4.83	**72.21 ± 3.60**	70.01 ± 3.60	71.84 ± 3.58	72.42 ± 3.57	70.86 ± 3.84	70.46 ± 3.65
water3	82.04 ± 5.92	83.42 ± 5.61	**83.88 ± 5.18**	82.90 ± 5.28	84.57 ± 5.51	83.55 ± 5.14	83.09 ± 5.14	82.78 ± 5.43
sonar	77.48 ± 8.80	74.04 ± 9.81	76.05 ± 9.87	75.24 ± 9.30	75.45 ± 9.46	**77.73 ± 9.35**	76.18 ± 9.35	76.16 ± 9.37
ozone	93.11 ± 1.20	93.26 ± 0.93	93.28 ± 1.11	91.47 ± 1.07	**93.29 ± 1.09**	93.02 ± 1.10	91.45 ± 1.10	91.55 ± 1.08
secom	91.58 ± 1.25	**93.04 ± 0.57**	92.80 ± 1.09	90.46 ± 1.25	92.01 ± 1.27	91.75 ± 1.35	91.17 ± 0.76	89.87 ± 1.32
water4	83.37 ± 4.92	81.20 ± 4.71	80.47 ± 6.09	81.51 ± 5.66	82.94 ± 5.99	**81.92 ± 5.61**	81.16 ± 5.68	80.53 ± 5.82
waveform	77.35 ± 1.79	75.87 ± 2.21	**78.01 ± 2.12**	76.24 ± 2.02	76.18 ± 2.25	74.45 ± 2.39	75.86 ± 2.03	74.89 ± 2.27
olitos	71.75 ± 10.81	65.83 ± 14.77	**69.06 ± 11.83**	67.96 ± 12.16	66.95 ± 12.05	68.11 ± 12.40	68.77 ± 11.97	67.73 ± 11.40
cleveland	54.39 ± 5.66	53.21 ± 6.02	54.37 ± 5.64	53.30 ± 5.69	54.85 ± 5.42	**54.92 ± 5.90**	53.51 ± 5.19	53.64 ± 5.55
web	50.39 ± 9.60	**57.38 ± 10.44**	56.49 ± 10.60	49.50 ± 11.94	49.66 ± 10.37	48.10 ± 11.88	55.12 ± 12.32	48.59 ± 9.56
glass	68.70 ± 9.35	**70.08 ± 8.70**	69.27 ± 8.99	68.76 ± 9.38	69.87 ± 9.43	68.30 ± 9.34	68.42 ± 9.41	68.28 ± 9.72
segment	94.63 ± 1.86	94.11 ± 1.49	93.57 ± 3.44	93.23 ± 1.76	94.48 ± 2.03	**94.64 ± 1.92**	93.01 ± 1.77	92.64 ± 2.07
multifeat	94.92 ± 1.49	88.35 ± 3.14	76.72 ± 5.55	88.67 ± 3.77	86.48 ± 2.26	84.69 ± 2.61	88.56 ± 2.20	**92.36 ± 1.62**
libras	68.21 ± 7.92	64.63 ± 8.86	61.18 ± 8.40	65.06 ± 7.76	65.44 ± 7.80	**66.62 ± 6.87**	63.13 ± 7.44	65.25 ± 7.06
arrhythmia	64.90 ± 4.97	**65.86 ± 5.87**	63.24 ± 6.68	65.01 ± 5.80	65.59 ± 5.52	65.48 ± 5.38	65.03 ± 5.24	63.81 ± 5.70
soybean	91.62 ± 3.12	77.72 ± 4.88	76.54 ± 5.00	**82.35 ± 5.79**	68.67 ± 5.96	81.12 ± 5.45	78.92 ± 3.82	79.24 ± 5.48
Avg.	78.40 ± 5.23	**77.26 ± 5.76**	76.35 ± 6.02	76.41 ± 5.80	76.43 ± 5.65	76.82 ± 5.72	76.54 ± 5.54	76.02 ± 5.54
Win Ratio		7/20	5/20	1/20	1/20	5/20	0	1/20

Table 7: Comparison with other FS methods: average feature subset size, where the bold indicates the statistically smallest size

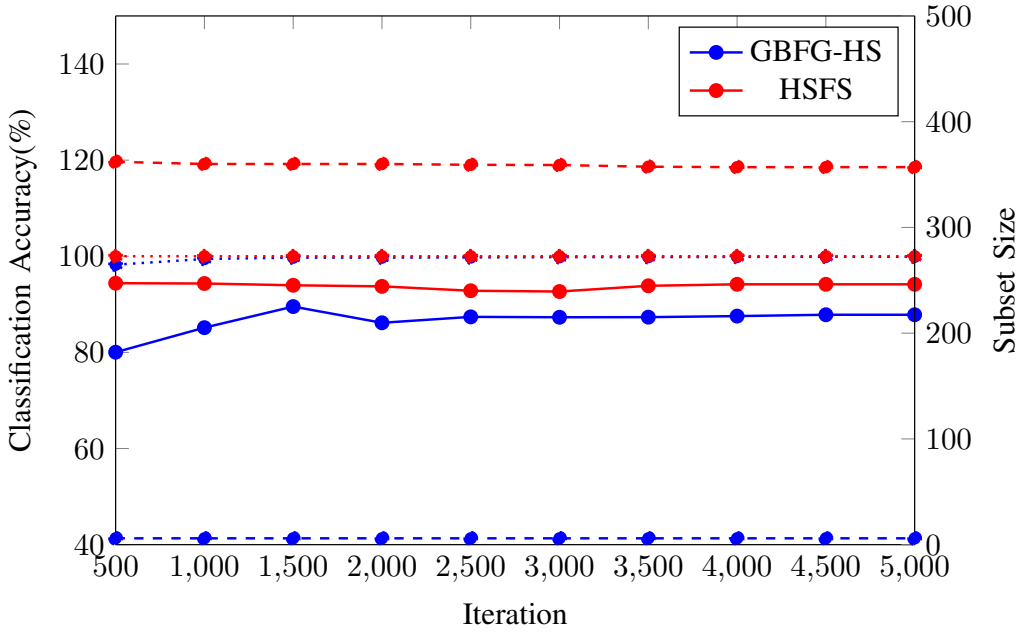
	Unred.	GBFG-HS	GBFG-FS	FRFG	GA-FS	PSO-FS	GHC-FS	HSFS
breastcancer	10.00 ± 0.00	**7.00 ± 0.47**	8.31 ± 0.93	8.04 ± 0.47	7.08 ± 0.27	7.08 ± 0.27	7.99 ± 0.48	7.08 ± 0.27
heart	14.00 ± 0.00	**6.60 ± 1.43**	7.18 ± 1.63	10.85 ± 1.39	9.60 ± 0.49	9.61 ± 0.49	9.08 ± 2.11	9.68 ± 0.53
vote	17.00 ± 0.00	**5.30 ± 0.82**	8.38 ± 2.36	12.99 ± 2.34	8.53 ± 0.58	9.86 ± 1.10	8.66 ± 1.57	9.13 ± 0.68
ionosphere	35.00 ± 0.00	**6.40 ± 1.07**	8.53 ± 1.85	9.09 ± 4.42	9.14 ± 1.63	11.24 ± 1.82	7.64 ± 1.26	16.13 ± 1.18
credit-g	21.00 ± 0.00	**10.60± 1.51**	14.32± 1.51	13.47 ± 1.84	12.18± 0.58	13.65 ± 0.94	12.62 ± 0.76	12.29 ± 0.59
water3	39.00 ± 0.00	**4.50 ± 1.58**	5.09 ± 3.36	18.07 ± 4.39	12.03± 1.04	23.79 ± 2.05	12.56 ± 1.09	20.39 ± 1.11
sonar	61.00 ± 0.00	**9.20 ± 1.40**	13.46± 1.85	22.26 ± 8.60	11.89± 1.17	12.63 ± 1.32	12.63 ± 1.32	31.53 ± 1.77
ozone	73.00 ± 0.00	**12.40± 0.97**	21.31± 8.22	34.24 ± 9.96	19.38± 1.43	39.04 ± 5.09	19.76 ± 2.86	35.84 ± 1.08
secom	591.00± 0.00	**2.10 ± 1.10**	8.09 ± 2.71	108.25± 74.32	94.40± 10.06	401.61± 42.25	3.57 ± 7.03	326.61± 3.42
water4	39.00 ± 0.00	**3.20 ± 0.42**	5.51 ± 2.66	15.12 ± 4.95	10.23± 1.32	21.98 ± 2.96	10.57 ± 1.05	19.54 ± 1.31
waveform	41.00 ± 0.00	11.70± 1.57	13.42± 1.16	13.34 ± 1.89	**11.14± 0.90**	17.58 ± 1.87	11.68 ± 0.68	18.46 ± 0.86
olitos	26.00 ± 0.00	**7.80 ± 1.62**	11.47± 2.04	11.83 ± 3.05	9.30 ± 0.67	12.92 ± 1.52	10.03 ± 1.14	13.84 ± 1.14
cleveland	14.00 ± 0.00	**6.70 ± 1.16**	6.84 ± 1.41	11.21 ± 1.63	8.03 ± 0.67	8.85 ± 0.48	7.27 ± 2.38	8.37 ± 0.69
web	2557.0± 0.00	**6.98 ± 0.74**	18.34± 1.56	411.29±489.09	987.96±159.06	940.99±162.14	17.68 ± 1.41	1459.65± 5.51
glass	10.00 ± 0.00	**5.50 ± 0.85**	6.35 ± 0.98	6.79 ± 0.53	6.73 ± 0.53	7.60 ± 0.64	6.73 ± 0.53	6.73 ± 0.53
segment	20.00 ± 0.00	**7.00 ± 1.15**	11.94± 2.53	9.11 ± 2.07	7.22 ± 0.63	8.82 ± 1.35	7.85 ± 0.83	8.32 ± 0.66
multifeat	650.00± 0.00	6.90 ± 0.88	13.00± 1.46	7.85 ± 0.24	43.00± 0.00	28.00 ± 0.00	**6.46 ± 0.5**0	353.72± 2.87
libras	91.00 ± 0.00	**8.60 ± 0.52**	15.19± 2.96	48.65 ± 18.05	17.50± 3.37	45.00 ± 6.85	16.87 ± 1.73	46.61 ± 1.99
arrhythmia	280.00± 0.00	**8.90 ± 1.60**	14.49± 3.29	54.24 ± 58.85	30.12± 2.64	160.66± 18.38	22.14 ± 1.99	150.58± 2.38
soybean	36.00 ± 0.00	**10.80± 1.99**	13.99± 2.24	19.00 ± 4.73	10.49± 0.63	18.93 ± 2.81	12.48 ± 0.72	16.88 ± 1.06
Avg.	231.25± 0.00	**7.41 ± 1.14**	11.26± 2.34	42.28 ± 34.64	66.30± 9.38	89.99 ± 12.72	11.21 ± 1.57	128.57± 1.48

Table 8: Comparison with other FS methods: average runtime of 10×10 cross-validation folds (MS) in terms of generating FS outcomes, where bold indicates the statistically most efficient

	GBFG-HS	GBFG-FS	FRFG	GA-FS	PSO-FS	GHC-FS	HSFS
breastcancer	426 ± 62	**6 ± 1**	780 ± 312.5	26 ± 14	81 ± 252	31 ± 165	572 ± 115
heart	230 ± 49	**4 ± 1**	486± 7	100 ± 50	84 ± 167	8 ± 19	333 ± 50
vote	391 ± 86	**7 ± 1**	4264 ± 861	364 ± 175	172 ± 296	59 ± 213	864 ± 162
ionosphere	278 ± 54	**4 ± 1**	9083 ± 1603	120 ± 289	187 ± 211	77 ± 452	454 ± 81
credit-g	1856 ± 277	**126 ± 2**	39605 ± 2319	4356 ± 272	255 ± 95	197 ± 106	2776 ± 108
water2	295 ± 69	**3 ± 1**	30455 ± 9875.5	5086 ± 1566	206 ± 116	137 ± 94	837 ± 55
sonar	287 ± 51	**22 ± 1**	20385± 3586	3332 ± 1339	78 ± 60	78 ± 60	577 ± 55
ozone	5208 ± 476	**634 ± 25**	1745352 ± 276629	83577 ± 9500	1791 ± 390	2904 ± 570	8164 ± 432
secom	1334 ± 146	**839 ± 951**	44678361 ± 5555810	205022 ± 16832	11220 ± 5392	1352 ± 3076	30692 ± 1818
water3	329 ± 36	**3 ± 1**	24318 ± 5586	4127 ± 2204	182 ± 66	70 ± 46	855 ± 87
waveform	33977 ± 3805	**1282 ± 53**	2264626 ± 400507.5	166172 ± 90326	3238 ± 1667	3619 ± 789	29292 ± 2371
olitos	153 ± 45	**7 ± 1**	426 ± 86	417 ± 286	93 ± 46	17 ± 34	195 ± 20
cleveland	238 ± 44	**8 ± 4**	674 ± 116	36 ± 21	94 ± 149	29 ± 81	330 ± 40
web	1414253 ± 318	**1412567 ± 421**	5460160 ± 1140662	675 ± 714	15267 ± 6607	4602 ± 1228	22264 ± 116
glass	181 ± 36	**2 ± 0**	166 ± 5	10 ± 8	126 ± 732	39 ± 304	262 ± 73
segment	2176 ± 266	**113 ± 4**	69656 ± 32476	1719 ± 1461	313 ± 211	123 ± 61	3359 ± 442
multifeat	8189 ± 664	828 ± 433	85723105 ± 2277627	144 ± 6	**4005 ± 390**	10229 ± 3030	54541 ± 8564
libras	510 ± 91	**49 ± 4**	104658 ± 38064	8096 ± 2333	392 ± 177	510 ± 189	1203 ± 40
arrhythmia	959 ± 171	**53 ± 7**	966082 ± 146275	23529 ± 432	2905 ± 2534	856 ± 123	7702 ± 2225
soybean	1061 ± 89	**48 ± 1**	41559 ± 7108	9534 ± 3565	287 ± 308	215 ± 490	1511 ± 40
Avg.	73616 ± 341	70830 ± 95	7059210± 494976	25822 ± 6569	2048 ± 993	1257 ± 556	8339 ± 844



(a) arrhythmia



(b) multifeat

Figure 4: Analysis of GBFG-HS and HSFS in terms of classification accuracy, subset size, and evaluation score for the arrhythmia and multifeat datasets.

6. Conclusion

This paper presents a novel framework for feature grouping, upon which two instantiations for the task of feature selection are proposed. The first offers a simple group-then-rank approach based on the selection of representative features from the feature grouping generated. The second employs a metaheuristic approach to strengthen the search since the simple inclusion of representatives selected from feature groups may not consider the information on inter-feature collaboration. In particular, the harmony search has been used for the purpose of selecting the final subsets. The multiple subsets ultimately selected by the harmony search provide more flexibility in returning the optimal subsets instead of producing just a single subset at a time. Interestingly, other search mechanisms, such as particle swarm optimisation, ant colony optimisation, and genetic algorithms, are equally applicable to this particular instantiation of the framework for the task of FS. However, the proposed harmony search-based method outperforms such FS methods, including FRFG, GA-FS, PSO-FS and GHC-FS, especially with respect to the resultant subset size across all twenty datasets investigated. The proposed FS methods also offer great potential for improving the performance of SVMs and other learning classifiers. Since conducting experimental evaluations similar to ours by using the other classifiers over the 20 datasets would require substantial computational efforts, we feel this investigation is best treated as an important piece of future work.

While both implementations of the proposed approach are promising, as indicated previously, more efficient strategies for generating groupings are highly desirable. At the moment, a rather simple approach of iteratively removing a single edge from the MST is employed. This could potentially be improved with two alternatives below. One particular strategy that could be employed here is the simultaneous removal of a number of edges in each iteration, where the edges are of equal weight. Another strategy might be to adopt a fuzzy approach where all edge weights are considered linguistically. It would also be interesting to further investigate the method used for the assessment of the quality of the feature subsets. In the current approach, they are assessed by evaluating the representative features drawn from each of the feature groups. Since the size of the selected subset is controlled by the number of groups, wrapper or hybrid methods could be considered for the grouping phase, further reducing the computational overhead. The deep features learned by deep neural networks often have high dimensionality. This

would be a potential application of the proposed method on deep features.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments, which significantly helped to improve the presentation of this paper. This work was partially funded by grants from the National Natural Science Foundation of China (91746103, 61672439, and 61673326) and partially by a Sêr Cymru II COFUND Fellowship, UK, No. 663830.

References

[1] Cavallari, S., Cambria, E., Cai, H., Chang, K.C.C., Zheng, V.W., 2019. Embedding both finite and infinite communities on graphs [application notes]. IEEE Computational Intelligence Magazine 14, 39–50.

[2] Chen, J., Mi, J., Lin, Y., 2020. A graph approach for fuzzy-rough feature selection. Fuzzy Sets and Systems 391, 96–116.

[3] Dua, D., Graff, C., 2017. UCI machine learning repository. URL: `http://archive.ics.uci.edu/ml`.

[4] Gonzalez-Lopez, J., Ventura, S., Cano, A., 2020. Distributed multi-label feature selection using individual mutual information measures. Knowledge-Based Systems 188, 105052.

[5] Gu, S., Cheng, R., Jin, Y., 2018. Feature selection for high-dimensional classification using a competitive swarm optimizer. Soft Computing 22, 811–822.

[6] Guha, R., Ghosh, M., Kapri, S., Shaw, S., Mutsuddi, S., Bhateja, V., Sarkar, R., 2019. Deluge based genetic algorithm for feature selection. Evolutionary intelligence , 1–11.

[7] Harandi, M., Salzmann, M., Hartley, R., 2017. Dimensionality reduction on spd manifolds: The emergence of geometry-aware methods. IEEE transactions on pattern analysis and machine intelligence 40, 48–62.

[8] Hsu, H.H., Hsieh, C.W., 2010. Feature selection via correlation coefficient clustering. Journal of Software 5, 1371–1377.

[9] Jensen, R., Mac Parthalain, N., Cornells, C., 2014. Feature grouping-based fuzzy-rough feature selection, in: Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on, pp. 1488–1495.

[10] Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S., 2020. A survey on knowledge graphs: Representation, acquisition and applications. arXiv preprint arXiv:2002.00388 .

[11] Kumar, P.M., Devi G, U., Manogaran, G., Sundarasekar, R., Chilamkurti, N., Varatharajan, R., 2018. Ant colony optimization algorithm with internet of vehicles for intelligent traffic control system. Computer Networks 144, 154 – 162.

[12] Lee, J., Yu, I., Park, J., Kim, D.W., 2019. Memetic feature selection for multilabel text categorization using label frequency difference. Information Sciences 485, 263–280.

[13] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H., 2017. Feature selection: A data perspective. ACM Computing Surveys (CSUR) 50, 94.

[14] Mahdavi, M., Fesanghary, M., Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. Applied mathematics and computation 188, 1567–1579.

[15] Moslehi, F., Haeri, A., 2020. A novel hybrid wrapper–filter approach based on genetic algorithm, particle swarm optimization for feature subset selection. Journal of Ambient Intelligence and Humanized Computing 11, 1105–1127.

[16] Sagi, O., Rokach, L., 2018. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, e1249.

[17] Shang, R., Xu, K., Shang, F., Jiao, L., 2020. Sparse and low-redundant subspace learning-based dual-graph regularized robust feature selection. Knowledge-Based Systems 187, 104830.

[18] Song, Q., Ni, J., Wang, G., 2011. A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE transactions on knowledge and data engineering 25, 1–14.

[19] Tang, X., Dai, Y., Sun, P., Meng, S., 2018. Interaction-based feature selection using factorial design. Neurocomputing 281, 47–54.

[20] Tran, H.N., Cambria, E., 2018. A survey of graph processing on graphics processing units. The Journal of Supercomputing 74, 2086–2115.

[21] Wang, H., Zhang, Y., Zhang, J., Li, T., Peng, L., 2019. A factor graph model for unsupervised feature selection. Information Sciences 480, 144–159.

[22] Wong, T.T., Yeh, P.Y., 2019. Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering .

[23] Xue, Y., Xue, B., Zhang, M., 2019. Self-adaptive particle swarm optimization for large-scale feature selection in classification. ACM Transactions on Knowledge Discovery from Data (TKDD) 13, 1–27.

[24] Yang, J., Hao, M., Liu, X., Wan, Z., Zhong, N., Peng, H., 2019. Fmst: an automatic neuron tracing method based on fast marching and minimum spanning tree. Neuroinformatics 17, 185–196.

[25] Zhang, Y., Gong, D.w., Gao, X.z., Tian, T., Sun, X.y., 2020. Binary differential evolution with self-learning for multi-objective feature selection. Information Sciences 507, 67–85.

[26] Zheng, L., Diao, R., Shen, Q., 2015. Self-adjusting harmony search-based feature selection. Soft Computing 19, 1567–1579.

[27] Zhong, L.W., Kwok, J.T., 2012. Efficient sparse modeling with automatic feature grouping. IEEE transactions on neural networks and learning systems 23, 1436–1447.

[28] Zhou, H., Zhang, Y., Zhang, Y., Liu, H., 2019. Feature selection based on conditional mutual information: minimum conditional relevance and minimum conditional redundancy. Applied Intelligence 49, 883–896.

[29] Zhu, X., Li, X., Zhang, S., Ju, C., Wu, X., 2016. Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE transactions on neural networks and learning systems 28, 1263–1275.