Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 5554873, 18 pages
https://doi.org/10.1155/2021/5554873

Research Article

A Constrained Feature Selection Approach Based on Feature Clustering and Hypothesis Margin Maximization

Samah Hijazi and Vinh Truong Hoang

Faculty of Information Technology, Ho Chi Minh City Open University, Ho Chi Minh City 700000, Vietnam

Correspondence should be addressed to Vinh Truong Hoang; vinh.th@ou.edu.vn

Received 2 February 2021; Revised 25 June 2021; Accepted 13 July 2021; Published 27 July 2021

Academic Editor: Cesar F. Caiafa

In this paper, we propose a semisupervised feature selection approach that is based on feature clustering and hypothesis margin maximization. The aim is to improve the classification accuracy by choosing the right feature subset and to allow building more interpretable models. Our approach handles the two core aspects of feature selection, i.e., relevance and redundancy, and is divided into three steps. First, the similarity weights between features are represented by a sparse graph where each feature can be reconstructed from the sparse linear combination of the others. Second, features are then hierarchically clustered identifying groups of the most similar ones. Finally, a semisupervised margin-based objective function is optimized to select the most data discriminative feature from within each cluster, hence maximizing relevance while minimizing redundancy among features. Eventually, we empirically validate our proposed approach on multiple well-known UCI benchmark datasets in terms of classification accuracy and representation entropy, where it proved to outperform four other semisupervised and unsupervised methods and competed with two widely used supervised ones.

1. Introduction

In many machine learning and pattern recognition applications, the data used is provided with a very large feature space describing it [1]. This space can be composed of the following four groups of features [2, 3]: (a) completely irrelevant, (b) redundant and weakly relevant, (c) nonredundant but weakly relevant, and (d) strongly relevant features. The first two groups can significantly degrade the performance of learning algorithms (classification, regression, and clustering), decrease their computational efficiency, increase their likelihood of overfitting, and undermine their generalization capability [4–6]. Thus, a good dimensionality reduction technique is usually applied with the aim of identifying features from groups (c) and (d).

Feature selection and feature extraction are two well-known dimensionality reduction tools [2]. In this work, we were particularly interested in feature selection due to its straightforwardness [7]. Unlike feature extraction, it simply obtains a subset of the original features that can best describe a dataset according to some objective function. This is done without applying any changes or transformation to the original feature space [8–10]. Feature selection can be categorized into different groups according to the availability of supervision information. It is unsupervised when label information is not available [5, 11–13], supervised when data is fully labeled [14–17], and semisupervised when few labeled data points are labeled and many are unlabeled [18–21].

In fact, many real-world applications pose the latter case where both supervised and unsupervised feature selection algorithms cannot fully take advantage of all data points in this scenario. Thus, we were interested in semisupervised methods to feat both labeled and unlabeled points. In fact, compared to class labels, pairwise constraints are another type of supervision information that can be acquired more easily [22]. These constraints simply specify whether a pair of data points belongs to the same class (must-link constraint) or to different classes (cannot-link constraint) without specifying the classes themselves [23, 24]. Some constraint scores use these two notions to rank features; however, they

neglect the information provided by unconstrained and unlabeled data [25].

On the other hand, although existing work in the domain of feature selection has resulted in many powerful techniques, most methods unintentionally ignore important information regarding the level of feature correlation during their selection process [26]. Typical examples are iterative methods by which a single individual feature is included/excluded once at a time from a candidate subset of features. This loss of important information might lead to having redundant features in the final selected subset. To be precise, multiple research works applied feature clustering with the aim of selecting a nonredundant and relevant feature subset. For that, they tended to build the similarity matrices between features using information theoretic measures such as mutual information [27], conditional mutual information [15], and maximal information coefficient [28]. It was also considered easy to transfer traditional data clustering methods to work for feature clustering; however, the challenge was in finding a suitable and meaningful definition of similarity notion between features [29].

Therefore, in this paper, we propose a semisupervised feature selection method that combines feature clustering with hypothesis margin maximization to obtain a nonredundant feature subset where features are ranked in the order of their relevance to a target concept. This method consists of (1) a constrained margin-based feature selection algorithm (Relief-Sc) that utilizes pairwise cannot-link constraints and benefits from both the local unlabeled neighborhood of the data points as well as the provided constraints and (2) a feature clustering method that combines sparse graph representation of the feature space with margin maximization.

We first benefit from the characteristics of nonparametrized sparse representation where it is possible to reconstruct each feature by the sparse linear combination of others [30]. This is done through solving an L_1-minimization problem. In fact, we treat these sparse coefficients as similarity weights to build our feature similarity matrix on top of which we apply clustering. For instance, we adopt an agglomerative single-link hierarchical clustering method. Upon obtaining the clustering solution, the features within each cluster (or group) play important roles in reconstructing each other and are thus assumed to be redundant in our scenario. Thus, we select a representative feature from each cluster [28, 31, 32]. For that, we utilize Relief-Sc to find the feature that best maximizes a pairwise constraint-relevance margin-based objective function. This maximization is quantized by assigning bigger weights to features that best contribute to enlarging a semisupervised distance metric called constrained hypothesis margin. In fact, as cannot-link constraints are considered more important than must-link constraints from the margin's point of view [33], our constrained hypothesis margin particularly utilizes them.

Finally, the core contribution of our work lies in the overall approach called FCRSC (Feature Clustering Relief-Sc) that aims at maximizing relevance while minimizing redundancy, and to the best of our knowledge, no work has previously done that by combining feature clustering upon sparse representation with the constrained hypothesis margin. To be precise, there exists another work [23] which is also based on the constrained margin concept but deals differently with redundancy, which is included in our experimental comparisons.

The rest of this paper is organized as follows. In Section 2, we present the proposed approach and detail its main building blocks. In Section 3, we present the experimental results to validate the proposed approach. Experiments are achieved on multiple well-known UCI machine learning datasets. Finally, the discussion and conclusion are provided in Section 4.

2. Feature Selection by Hierarchical Clustering and Hypothesis Margin Maximization

In this section, we detail each step of the proposed approach. For instance, in Section 2.1, we explain how to represent the relationships between features to be used in clustering. In Section 2.2, we briefly explain hierarchical clustering in our context, and in Section 2.3, we give a detailed explanation of hypothesis margin and its concepts. Finally, in Section 2.4, we present the overall semisupervised feature selection approach that combines both feature clustering and hypothesis margin maximization.

2.1. Feature Space Sparse Graph Construction. Sparse graph representation has received a great deal of attention in recent years [1, 34, 35]; this is due to its ability to find the most compact representation of the original data and to preserve its underlying discriminative information [36]. In fact, the sparse representation model generally aims at representing a data point using as few as possible other data points within the same dataset (overcomplete dictionary). Conventionally, some recent work utilized sparse theory to build the similarity matrix between data points (instances) by assuming that each point can be reconstructed by the sparse linear combination of other points [30, 37, 38]. On the contrary, in this paper, the similarity graph adjacency structure and the corresponding graph weights are built simultaneously among features instead of data points [39, 40]. While computing the sparse linear coefficients by solving an L_1-norm regularized least squares loss problem, the most similar features as well as their estimated similarity weights to the reconstructed feature are identified. Hence, we obtain the feature-wise sparse similarity matrix that will be used in grouping features.

It is important to note that the main advantages of using the L_1-graph are the following:

(i) It can lead to a sparse representation which can enhance the efficiency and the robustness to noise in learning algorithms [36]

(ii) While many clustering algorithms [41, 42] are very sensitive to some parameters when building their similarity graphs (like the performance of traditional spectral clustering that is heavily related to the choice of sigma in Heat Kernel), our graph construction is parameter-free

(iii) It obtains both the graph adjacency structure and the corresponding similarity weights by L_1-optimization, while L_2-graphs usually separate them into two steps

To mathematically formalize the problem, we consider a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times F}$ including all the features in its columns. To be clear, we also consider $\mathbf{X}$ from the features point of view; thus, it can be expressed as $\mathbf{X} = [\mathbf{A}_1, \ldots, \mathbf{A}_i, \ldots, \mathbf{A}_F] \in \mathbb{R}^{N \times F}$. As our aim is to find the reconstruction sparse linear coefficients of each feature, we use the second representation of $\mathbf{X}$ in this section. Therefore, to reconstruct each feature (attribute) $\mathbf{A}_i$ using as few entries of $\mathbf{X}$ as possible, we solve an L_0-norm optimization problem as follows:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_0, \text{ s.t.}, \quad \mathbf{A}_i = \mathbf{X}\mathbf{s}_i, \tag{1}$$

where $\|.\|_0$ denotes the L_0-norm, which is equal to the number of nonzero components in $\mathbf{s}_i$, and $\mathbf{s}_i = [s_{i1}, \ldots, s_{i,i-1}, 0, s_{i,i+1}, \ldots, s_{iF}]^T$ is an F-dimensional coefficient vector in which the i^{th} element is equal to zero (implying that $\mathbf{A}_i$ is removed from $\mathbf{X}$). The element s_{ij} $(i \neq j)$ denotes the contribution of any other feature $\mathbf{A}_j$ in reconstructing $\mathbf{A}_i$.

Note that the solution of (1) is NP-hard. Thus, a sparse vector $\mathbf{s}_i$ can be approximately estimated by the following L_1-minimization problem [38]:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1, \text{ s.t.}, \quad \mathbf{A}_i = \mathbf{X}\mathbf{s}_i, 1 = \mathbf{1}^T\mathbf{s}_i, \tag{2}$$

where $\|.\|_1$ denotes the L_1-norm and $\mathbf{1} \in \mathbb{R}^F$ is a vector of all ones values.

In fact, due to the presence of noise, the constraint $\mathbf{A}_i = \mathbf{X}\mathbf{s}_i$ in (2) does not always hold. Thus, in [1], Liu and Zhang mentioned a modified robust extension (invariant to translation and rotation) to mitigate this problem. It can be defined as follows:

$$\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_1, \text{ s.t.}, \quad \|\mathbf{A}_i - \mathbf{X}\mathbf{s}_i\|_2 < \xi, 1 = \mathbf{1}^T\mathbf{s}_i, \tag{3}$$

where ξ represents a given error tolerance. The sparse vector $\mathbf{s}_i$ is computed for each feature $\mathbf{A}_i$. The optimal solution of (3) for each sample $\mathbf{A}_i$ is a sparse vector $\widehat{s}_i$; this vector allows building the sparse reconstructive similarity matrix $\mathbf{S} = (\widehat{s}_{i,j})_{F \times F}$, defined by

$$\mathbf{S} = \left[\widehat{s}_1, \ldots, \widehat{s}_i, \ldots, \widehat{s}_F\right]^T. \tag{4}$$

The L_1-minimization problem can be solved in polynomial time by standard linear programming methods [30] using publicly available packages such as SLEP package [43]. As the vector $\widehat{\mathbf{s}}_i$ is sparse (a lot of its components can be zero and few have nonzero values), the features in the dataset which are far from each other will have very small (zero or near zero) coefficients. This solution can reflect the intrinsic geometric properties of feature space. Algorithm 1 summarizes the graph construction.

2.2. Agglomerative Hierarchical Feature Clustering. As we mentioned before, a good feature selection algorithm is expected to find features that are most relevant in terms of discriminating data points between different classes while being least correlated to each other. The latter is similar to the general assumption of clustering where the data is partitioned such that points within the same cluster are as similar as possible to each other and as different as possible from points in other clusters [27]. Focusing on the idea of finding the least redundant (most diverse) features brought up clustering the features themselves instead of clustering data points [15, 44]. For instance, this minimized-redundancy among features can be obtained by dividing them into different groups according to a similarity criterion followed by choosing one or more features to represent each group.Among the four primary clustering categories that are hierarchical, density-based, statistical, and centroid-based [45], we were interested in hierarchical clustering. It is useful when the structure of the dataset can hold nested clusters and does not require a predefined number of clusters, since hierarchical clustering algorithm outputs a tree diagram called dendrogram. The dendrogram that records the sequence of mergers of clusters (features) into larger clusters presents a multilevel grouping of these features [31].

Hence, depending on the cutoff level of the dendrogram, the number of obtained clusters can vary between 1 and F. Intuitively, at lower levels of the dendrogram, we have the clusters of most redundant features that are first to be grouped. Thus, cutting at low levels results in a higher number of clusters and therefore more cluster representatives. This yields a bigger output feature subset. However, cutting the dendrogram at higher levels results in a smaller number of clusters and thus fewer cluster representatives, i.e., a smaller output feature subset. Hence, although choosing a high level of clustering ensures eliminating more redundancy, it could still cause more information loss.

As a result, a good quality of clustering is closely related to the problem-adequate choice of the cutoff level. Therefore, we decided cutoff when merging distances become large enough to create a second-level hierarchy, which means when clusters of features start being merged together instead of merging individual features. This is explained by our goal of reducing redundancy among features to a certain level without excessive compression that might lead to some information loss.

In summary, hierarchical clustering can have multiple methods for computing the distance between clusters (ward, complete, median, centroid, single, and others); however, as we are working with feature clustering and as we wish to merge the clusters based on the most similar features (not their average or the farthest two features within a cluster, i.e., complete linkage), the agglomerative single-linkage hierarchical clustering was applied in this paper. It takes as input the $F \times F$ feature-wise similarity matrix denoted by $\mathbf{S}$ obtained from Section 2.1. This algorithm initially assigns each feature to its own cluster and then finds the largest element s_{ij} in $\mathbf{S}$; after that, the corresponding two most similar

Input: Training data $\mathbf{X}$
Output: Feature similarity matrix $\mathbf{S}$
(1) For each feature $\mathbf{A}_i$
(i) Solve equation (3) to obtain $\widehat{s}_i$
(ii) Iff the jth entry of $\widehat{s}_i$ denoted $\widehat{s}_{i,j} \neq 0$ set the weight $\mathbf{S}_{ij} = |\widehat{s}_{i,j}|$, $1 \leq i, j \leq F$
(2) Force symmetry by $\mathbf{S} = (\mathbf{S}^T + \mathbf{S})/2$

Algorithm 1: Sparse graph construction.

clusters (or features) are merged based on s_{ij}. After each merging step, the similarity matrix is updated by replacing the two grouped clusters (or features) by the newly formed cluster in $\mathbf{S}$.This update can be expressed as $s_{e,ij} = \max\{s_{ei}, s_{ej}\}$, where $s_{e,ij}$ represents the similarity between the new obtained cluster (by merging two clusters $\mathbf{C}_i$ and $\mathbf{C}_j$ or features $\mathbf{A}_i$ and $\mathbf{A}_j$) and any other existing cluster $\mathbf{C}_e$ (or existing feature $\mathbf{A}_e$). s_{ei} and s_{ej} are the respective similarities between cluster (or feature) pairs.

2.3. Hypothesis Margin in the Semisupervised Constrained Context. In this section, we present a concise summary of our margin-based feature selection algorithm called Relief-Sc [33]. The power of Relief-Sc lies in its ability to solve a simple convex problem in a closed form (obtaining a unique solution) through utilizing a highly nonlinear classifier to evaluate its margin-based objective function. The latter unique solution is a set of features ranked in the order of their relevance with respect to a specified problem (e.g., the classification of a newly arriving data point). It tends to rank features in terms of their ability to maximize this hypothesis margin-based objective function using a given set of cannot-link constraints. As a reminder, these constraints are a cheaper kind of supervision information specifying only that two data points should belong to different groups without specifying the groups themselves [23, 46]. However, we noticed that a drawback of Relief-Sc and its basic supervised precursor Relief algorithm [47] is that they lack the ability of dealing with redundancy among features. Nevertheless, it is well known that eliminating redundant features is a very important aspect of feature selection.

By definition, the hypothesis margin is the largest distance a data point can travel in its feature space without affecting the labeling structure of the dataset and thus without altering the label prediction of a new arriving point. Therefore, having a large margin provides high classifier-confidence when it is undergoing prediction. The hypothesis margin is closely built up on two important notions that were initially suggested in a supervised context [47, 48]. These notions, namely, the near-hit and the near-miss of a data point, were defined as its nearest point within the same class and its nearest point from a different class, respectively. Thus, consider $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times F}$ where N is the number of data points, F is the number of features, and $\mathbf{x}_n = (x_{n1}, \ldots, x_{ni}, \ldots, x_{nF})^T$ is the nth data point characterized by F features.

Original definition: In a supervised context, the near-hit of a data point $\mathbf{x}_n$ is its nearest point within the same class denoted by $\mathbf{H}(\mathbf{x}_n)$ and the near-miss of a data point $\mathbf{x}_n$ is its nearest point from a different class denoted by $\mathbf{M}(\mathbf{x}_n)$.

However, in this paper, we work in a semisupervised context where the only supervision information available is in the form of pairwise cannot-link constraints. Hence, a modification of the near-hit and near-miss notions is to be applied.

Definition 1. In our considered semisupervised context, let $(\mathbf{x}_n, \mathbf{x}_m)$ be one cannot-link constraint in the set of constraints $\mathscr{C} = \{(\mathbf{x}_n, \mathbf{x}_m)\}$. Then, the nearest point to $\mathbf{x}_m$ or its near-hit, denoted by $\mathbf{H}(\mathbf{x}_m)$, will now represent the near-miss of $\mathbf{x}_n$. On the other hand, $\mathbf{H}(\mathbf{x}_n)$ represents the near-hit of $\mathbf{x}_n$. The number of constraints $|\mathscr{C}|$ is a user predefined value. An example is illustrated in Figure 1.

As a side note, we would like to explicitly mention how Relief-Sc is considered semisupervised compared to other constrained scores that utilize only the constraint itself in ranking features [25]. In fact, Relief-Sc does not only depend on the cannot-link constraint $\mathscr{C} = \{(\mathbf{x}_n, \mathbf{x}_m)\}$ but also on its unlabeled local neighborhood as can be seen in Figure 1. These unlabeled data points presented as black dots and denoted by $\mathbf{H}(\mathbf{x}_n)$ and $\mathbf{H}(\mathbf{x}_m)$ are considered in the margin calculation making the context semisupervised.

Definition 2. The constrained hypothesis margin is calculated as the difference between the distance from the data point $\mathbf{x}_n$ to the near-hit of the data point $\mathbf{x}_m$, i.e., $\mathbf{H}(\mathbf{x}_m)$, and the distance from the data point $\mathbf{x}_n$ to its own near-hit, i.e., $\mathbf{H}(\mathbf{x}_n)$.

$$\rho(\mathbf{x}_n, \mathbf{x}_m) = \sum_{i=1}^{F} [\Delta(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}(\mathbf{x}_m)) - \Delta(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}(\mathbf{x}_n))]. \tag{5}$$

The $\Delta(\mathbf{A}_i, \mathbf{p}_1, \mathbf{p}_2)$ function, defined as the L_1-norm (Manhattan distance) in our work, calculates the distance between any two data points $\mathbf{p}_1$ and $\mathbf{p}_2$ on a particular feature $\mathbf{A}_i$. Note that, for quantitative features, Δ is calculated as follows:

$$\Delta(\mathbf{A}_i, \mathbf{p}_1, \mathbf{p}_2) = \frac{|\text{value}(\mathbf{A}_i, \mathbf{p}_1) - \text{value}(\mathbf{A}_i, \mathbf{p}_2)|}{\max(\mathbf{A}_i) - \min(\mathbf{A}_i)}, \tag{6}$$

and for qualitative features, Δ is calculated as follows:

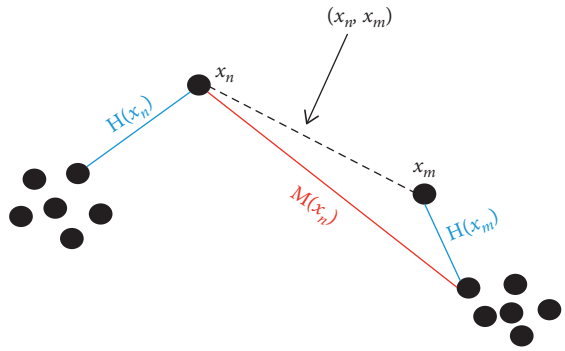

FIGURE 1: The modified notions of near-miss and near-hit used in the semisupervised constrained margin calculation. The black dots represent unlabeled data points, and the dashed line represents the cannot-link constraint between a data couple $(\mathbf{x}_n, \mathbf{x}_m)$. $\mathbf{H}(\mathbf{x}_m)$ represents the near-miss $\mathbf{M}(\mathbf{x}_n)$ of $\mathbf{x}_n$, and $\mathbf{H}(\mathbf{x}_n)$ represents the near-hit of $\mathbf{x}_n$.

$$\Delta(\mathbf{A}_i, \mathbf{p}_1, \mathbf{p}_2) = \begin{cases} 0, & \text{if value}(\mathbf{A}_i, \mathbf{p}_1) = \text{value}(\mathbf{A}_i, \mathbf{p}_2), \\ 1, & \text{otherwise}, \end{cases} \tag{7}$$

where $\text{value}(\mathbf{A}_i, \mathbf{p})$ is the value of the data point $\mathbf{p}$ over the i-th feature. The minimum and maximum values of a particular feature $\mathbf{A}_i$ denoted by $\min(\mathbf{A}_i)$ and $\max(\mathbf{A}_i)$ are evaluated over the whole set of data points. This normalization ensures that all weight-updates range between 0 and 1 for both quantitative and qualitative features.

As the ability of a feature to discriminate between data points can be evaluated by how much it contributes to the margin's maximization, we represent and quantify this contribution by a weight vector $\mathbf{w} = (w_1, \ldots, w_i, \ldots, w_F)^T$ spanning over the F features. Thus, a weight that is equal to the value of the hypothesis margin a feature can induce, will be assigned to each feature correspondingly.

Definition 3. The constrained weighted hypothesis margin for a particular constraint $(\mathbf{x}_n, \mathbf{x}_m)$ is presented as

$$\rho(\mathbf{w}, (\mathbf{x}_n, \mathbf{x}_m)) = \sum_{i=1}^{F} w_i [\Delta(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}(\mathbf{x}_m)) - \Delta(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}(\mathbf{x}_n))]. \tag{8}$$

Definition 4. The constrained weighted margin-based objective function to be optimized by Relief-Sc over the whole constraint set $\mathscr{C}$ can be written as follows:

$$\max_{\mathbf{w}} \sum_{(\mathbf{x}_n, \mathbf{x}_m) \in \mathscr{C}} \rho(\mathbf{w}, (\mathbf{x}_n, \mathbf{x}_m)), \text{ s.t.} \|\mathbf{w}\|_2^2 = 1, \text{ and } \mathbf{w} \geq \mathbf{0}, \tag{9}$$

where the weight vector $\mathbf{w} \geq \mathbf{0}$ holds positive values for relevant features since it is a distance metric, and $\|\mathbf{w}\|_2^2 = 1$ is to prevent the vector from being maximized with no bounds [49].

Definition 5. From (8) and (9), we denote by $\mathbf{z} = (z_1, \ldots, z_i, \ldots, z_F)^T$ the margin vector summed over all cannot-link constraints in $\mathscr{C}$, where each element z_i of $\mathbf{z}$ corresponds to the margin induced by a specific feature $\mathbf{A}_i$. z_i can be calculated as follows:

$$z_i = \sum_{(\mathbf{x}_n, \mathbf{x}_m) \in \mathscr{C}} (\Delta(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}(\mathbf{x}_m)) - \Delta(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}(\mathbf{x}_n))). \tag{10}$$

Accordingly, the optimization problem can be formulated as follows:

$$\max_{\mathbf{w}} \mathbf{w}^T \mathbf{z}, \text{ s.t.} \|\mathbf{w}\|_2^2 = 1, \quad \mathbf{w} \geq \mathbf{0}. \tag{11}$$

Equation (11) shows that the features that participate the most in the overall maximization of the margin will be assigned a higher weight and will be consequently selected.

Then, the optimum solution can be obtained in a closed form as follows:

$$\mathbf{w} = \frac{(\mathbf{z})^+}{\|(\mathbf{z})^+\|_2}, \quad \text{where,} \quad (\mathbf{z})^+ = [\max(z_1, 0), \ldots, \max(z_F, 0)]^T. \tag{12}$$

Just with the aim of making Relief-Sc more robust, the hypothesis margin can be calculated over a group of K-nearest neighbors (KNN) for each pair of cannot-link constraints. For instance, instead of calculating the margin over only the nearest hit and the nearest miss for the points $\mathbf{x}_n$ and $\mathbf{x}_m$, we can evaluate the margin over K-nearest hits and K-nearest misses represented by $K\mathbf{H}(\mathbf{x}_n)$: $\{\mathbf{H}_1(\mathbf{x}_n), \mathbf{H}_2(\mathbf{x}_n), \ldots, \mathbf{H}_K(\mathbf{x}_n)\}$ and $K\mathbf{H}(\mathbf{x}_m)$: $\{\mathbf{H}_1(\mathbf{x}_m), \mathbf{H}_2(\mathbf{x}_m), \ldots, \mathbf{H}_K(\mathbf{x}_m)\}$, respectively, where K is a user predefined parameter representing how many closest points to $\mathbf{x}_n$ and $\mathbf{x}_m$ are to be considered. In fact, this means that the margin will be averaged over a larger neighborhood and thus will be less vulnerable to noisy data. Consequently, the ith element (corresponding to the ith feature) of the margin vector $\mathbf{z}$ can be given by

$$z_i = \sum_{(\mathbf{x}_n, \mathbf{x}_m) \in \mathscr{C}} \frac{1}{K} \sum_{j=1}^{K} \left(\Delta\left(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}_j(\mathbf{x}_m)\right) - \Delta\left(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}_j(\mathbf{x}_n)\right)\right). \tag{13}$$

Finally, in the context of a robust constrained hypothesis margin over KNN, Relief-Sc utilizes a cannot-link constraint set to evaluate the averaged margin $\mathbf{z}$ in order to optimize $\mathbf{w}$ directly as shown in step 3 of Algorithm 2.

2.4. Proposed Feature Selection Approach. Our proposed approach that combines feature clustering with Relief-Sc, called FCRSC, is a filter-type feature selection method (it does not depend on the performance of any learning algorithm while obtaining its ranked feature subset). In addition, one very important advantage of our method is that it is nonparametric, which means its performance is not vulnerable to being closely related to any tuned parameter. Moreover, we do not specify the number of clusters to be obtained from hierarchical clustering, but instead we state a mechanism that choses the cutoff automatically such that no

Input:
(i) Training data $\mathbf{X}$
(ii) Set of cannot-link constraints $\mathscr{C}$
(iii) Number of nearest neighbors K
Output: Weight vector $\mathbf{w}$
(1) Calculate $K\mathbf{H}(\mathbf{x}_n)$ and $K\mathbf{H}(\mathbf{x}_m)$ for each cannot-link constraint in $\mathscr{C}$ with respect to
(2) $\mathbf{X}$ For $i = 1, \ldots, F$,

$$z_i = \sum_{(\mathbf{x}_n, \mathbf{x}_m) \in \mathbf{C}} 1/K \sum_{j=1}^{K} \left(\Delta\left(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}_j(\mathbf{x}_m)\right) - \Delta\left(\mathbf{A}_i, \mathbf{x}_n, \mathbf{H}_j(\mathbf{x}_n)\right)\right)$$

end For
(3) $\mathbf{w} = (\mathbf{z})^+ / \|(\mathbf{z})^+\|_2$, where, $(\mathbf{z})^+ = [\max(z_1, 0), \ldots, \max(z_F, 0)]^T$

Algorithm 2: Relief with side constraints {Relief-Sc}.

Input:
(i) Set of clusters C obtained by hierarchical clustering
(ii) Weight vector $\mathbf{w}$ obtained by Algorithm 2
Output: Ranked feature subset ranked $\mathbf{F_s}$
Initialize: Feature set $\mathbf{F_s} = \phi$
(1) **For** each cluster c_l in C

if $|c_l| = 1$

$\mathbf{F_s} \leftarrow \mathbf{F_s} \cup$ feature in c_l

end if
if $|c_l| > 1$

Sorted $\leftarrow$ sort (features $\in\in c_l$, descending $\mathbf{w}$)
Representative$\leftarrow$Sorted(first row)
$\mathbf{F_s} \leftarrow \mathbf{F_s} \cup$ Representative

end if

end For
(2) ranked $\mathbf{F_s} \leftarrow$ sort ($\mathbf{F_s}$, descending $\mathbf{w}$)

Algorithm 3: Proposed algorithm, FCRSC.

excessive compression or trivial solutions are obtained (explained in Section 2.2).

To sum it up, first, we build the feature similarity graph on top of which we apply agglomerative hierarchical clustering. This graph is obtained through sparse coding, where the assigned similarity weights between features are in fact sparse coefficients indicating how much each feature contributes to the reconstruction of the other.

It is very important to find a clustering solution C where features compression is not exaggerated (obtaining very few clusters) or underestimated (obtaining the trivial solution: each feature in its own cluster). Meanwhile, a weight is also assigned to each feature by Relief-Sc as it maximizes the semisupervised margin-based objective function.

Thus, the significance of this approach lies in the last but most important algorithm called FCRSC. It starts with the two available ingredients, i.e., the clustering solution C obtained by hierarchical clustering and the weight vector $\mathbf{w}$ obtained by Relief-Sc. Then, for each of the clusters c_l in C, the number of features within c_l is evaluated and denoted by $|c_l|$. When a cluster has one and only one feature, i.e., $|c_l| = 1$ (considered not redundant at all), it is directly added to the chosen feature subset $\mathbf{F_s}$. However, when more than one feature is assigned to c_l, the features within c_l are sorted in the descending order of their corresponding margin weights given in $\mathbf{w}$ and stored in a variable named *Sorte d*; then, the feature with the highest weight (most relevant obtained by getting the first ranked feature in Sorted) is added to the feature set $\mathbf{F_s}$, and the rest are eliminated as they are judged to be redundant. After getting the representative feature from each cluster, these are sorted again in the descending order of $\mathbf{w}$ (stored in ranked $\mathbf{F_s}$) leading to the optimization of a twofold objective problem, i.e., (1) minimizing redundancy between features in $\mathbf{F_s}$ and (2) maximizing relevance between the features and the cannot-link constraints in $\mathscr{C}$. Thus, we obtain the ranked feature subset *ranke d*$\mathbf{F_s}$. Note that the number of features in ranked $\mathbf{F_s}$ will be equal to

the number of obtained clusters denoted by $|C|$. We illustrate the FCRSC in Figure 2.

3. Experimental Results

In this section, we compare the performance of our proposed FCRSC approach with some of the well-known state-of-the-art feature selection methods. This comparison is applied in terms of classification accuracy, redundancy-removal ability, and execution time. Note that a machine with 2.60 GHz CPU and 16 GB of RAM was used to perform the experiments. The used datasets, feature selection methods, and classifiers are detailed in the following sections.

3.1. Datasets' Description. To evaluate the proposed approach, six well-known benchmark datasets representing a variety of problems were used. These datasets are Wine, WDBC, Ionosphere, Spambase, Sonar, and Arrhythmia from the UCI machine learning repository [50]. We summarize the main characteristics of each dataset in Table 1, where the first column identifies the name of the dataset, the second column shows its corresponding number of data points, the third column shows the data dimension, and finally the last column specifies the number of classes.

Usually, a dataset can have features lying within different ranges, which affects the performance of feature selection algorithms leading to unreliable outcomes. Thus, similarly to [5], we normalize the features of each dataset using max–min criterion to scale their values between zero and one. Furthermore, for Arrhythmia dataset where some of the feature values are missing, we replace them by the average of all available values of the same corresponding feature.

In addition, we also similarly partition each dataset into 2/3 for training and 1/3 for testing. This process is repeated independently 10 times, and only the averaged results are recorded. In each run, feature selection followed by classifiers learning is applied to the training subset to allow ranking the features according to their assigned scores by different algorithms and then training the classifier in these same ranked feature sets. The classification accuracy that can be obtained by each ranked set of features (each feature selection method) is then measured by applying the learned classifier on the testing subset defined by these same features.

3.2. Used Filter Feature Selection Methods for Comparison. In order to evaluate the performance of our proposed FCRSC method, we apply experimental comparisons with respect to six filter-type ranking feature selection methods, out of which two are supervised, two are unsupervised, and two are semisupervised. We briefly describe these methods as follows:

(i) Variance score [51]: it is generally known as the simplest unsupervised feature evaluation method. It uses the variance along a specific feature to reflect its representative power. Then, the features with the maximum variance are selected as it assumes that a feature with higher variance contains more information and is more relevant.

Variance score depends on the following equation to evaluate features:

$$v_i = \frac{1}{N}\sum_{n=1}^{N}\left(A_{in} - \mu_{\mathbf{A}_i}\right)^2, \tag{14}$$

where N is the number of data points, A_{in} is the value of feature $\mathbf{A}_i$ on a data point $\mathbf{x}_n$, and $\mu_{\mathbf{A}_i} = 1/N\sum_{n=1}^{N} A_{in}$.

(ii) Laplacian score [11]: it is a well-known unsupervised feature selection method which not only depends on selecting the features of larger variances and higher representative power but also considers their locality preserving ability. Its key assumption is that data points within the same class should be close to each other and far otherwise. Note that the smaller the Laplacian score, the better.

The Laplacian score is based on the following equation:

$$L_i = \frac{\sum_{\mathbf{x}_n,\mathbf{x}_m\in\mathbf{X}}\left(A_{in} - A_{im}\right)^2 S_{nm}}{\sum_{\mathbf{x}_n\in\mathbf{X}}\left(A_{in} - \mu_{\mathbf{A}_i}\right)^2 D_{nn}}, \tag{15}$$

where D is a diagonal matrix $D_{nn} = \sum_n S_{nm}$ and S_{nm} is the neighborhood matrix between data points.

(iii) Minimum-redundancy-maximum-relevance (mRMR) [14]: it is a supervised multivariate feature selection method that is said to output a feature subset having the most diverse features (as noncorrelated as possible) while still having a high correlation with the class label.

The maximal relevance criterion is

$$\max \text{Relevancy}\left(\mathbf{F_s}, \mathbf{Y}\right) = \frac{1}{\left|\mathbf{F_s}\right|}\sum_{\mathbf{A}_i\in\mathbf{F_s}} I\left(\mathbf{A}_i; \mathbf{Y}\right). \tag{16}$$

The minimal redundancy criterion is

$$\min \text{Redundancy}\left(\mathbf{F_s}\right) = \frac{1}{\left|\mathbf{F_s}\right|^2}\sum_{\mathbf{A}_i,\mathbf{A}_j\in\mathbf{F_s}} I\left(\mathbf{A}_i; \mathbf{A}_j\right), \tag{17}$$

where $\mathbf{F_s}$ is the selected subset of features, $|\mathbf{F_s}|$ is the number of features in $\mathbf{F_s}$, $\mathbf{Y}$ is the vector of class labels, and $I(\mathbf{x}; \mathbf{y})$ denotes the mutual information between elements $\mathbf{x}$ and $\mathbf{y}$.

(iv) ReliefF [52]: it is a supervised margin-based feature selection algorithm. It is a robust extension of Relief that chooses random data points and uses them to calculate the weights of the feature relevance based on a predefined number of nearest neighbors [47].

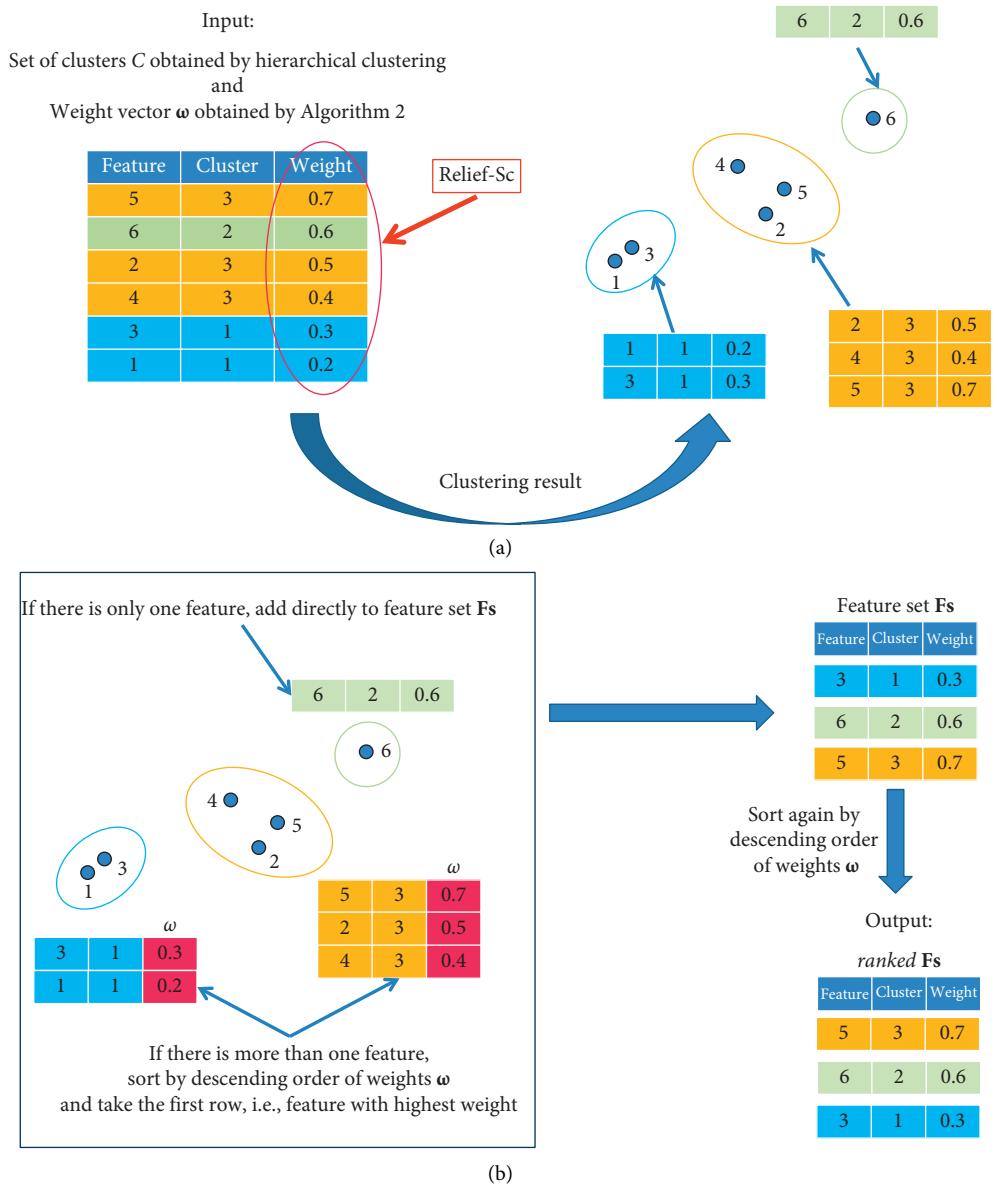

Figure 2: Illustration of the proposed FCRSC algorithm with a toy example.

Table 1: Datasets and their characteristics.

Dataset	# points	# features	# classes
Wine	178	13	3
WDBC	569	30	2
Ionosphere	351	34	2
Spambase	4601	57	2
Sonar	208	60	2
Arrhythmia	452	279	13

ReliefF depends on the following update rule upon a group of random data points:

$$w_i^{\text{new}} = w_i^{\text{old}} - \frac{1}{(m \cdot K)} \sum_{x \in X_m} \sum_{j=1}^{K} \Delta\left(\mathbf{A}_i, \mathbf{x}, \mathbf{H}_j\right) + \frac{1}{(m \cdot K)} \sum_{c \neq \text{class}(\mathbf{x})} \left[\frac{P(c)}{1 - P(\text{class}(\mathbf{x}))} \sum_{j=1}^{K} \Delta\left(\mathbf{A}_i, \mathbf{x}, \mathbf{M}_j(c)\right) \right], \tag{18}$$

where w_i is the weight assigned to the feature $\mathbf{A}_i$, $\mathbf{X}_m$ represents the set of random instances used to evaluate $\mathbf{w}$ with $|\mathbf{X}_m| = m$, K is the number of nearest hits and misses to be considered, c represents the class, and $\mathbf{H}$ and $\mathbf{M}(c)$ represent the near-hits and near-misses, respectively. Moreover, $P(c)$ represents the prior probability of class c (estimated from the training set) and $1 - P(\text{class}(\mathbf{x}))$ represents the sum of probabilities for the misses' classes.

(v) Simba with side constraints (Simba-Sc) [23]: it is a semisupervised margin-based algorithm that iteratively utilizes pairwise constraints, specifically cannot-link ones, to evaluate the ability of features to discriminate data points. This score uses a gradient ascent method to maximize its margin-based objective function. Thus, a higher score means a more relevant feature. Note that Simba-Sc has a mechanism to deal with redundancy; however, it may still choose correlated features only when this contributes positively to the overall performance.

The update rule used by Simba-Sc can be summarized as follows:

$$w_i^{\text{new}} = w_i^{\text{old}} + \sum_{(\mathbf{x}_n, \mathbf{x}_m) \in \mathbf{C}} \frac{1}{2}\left(\frac{\left(\Delta\left(\mathbf{A}_i, \mathbf{x}_n, H\left(\mathbf{x}_m\right)\right)\right)^2}{\left\|\Delta\left(\mathbf{A}_i, \mathbf{x}_n, H\left(\mathbf{x}_m\right)\right)\right\|_{\mathbf{w}}} - \frac{\left(\Delta\left(\mathbf{A}_i, \mathbf{x}_n, H\left(\mathbf{x}_n\right)\right)\right)^2}{\left\|\Delta\left(\mathbf{A}_i, \mathbf{x}_n, H\left(\mathbf{x}_n\right)\right)\right\|_{\mathbf{w}}} \right) w_i, \tag{19}$$

where $\|\mathbf{v}\|_{\mathbf{w}} = \|\mathbf{w} \circ \mathbf{v}\| = \sqrt{\sum_i w_i^2 v_i^2}$.

(vi) Relief-Sc [33]: the semisupervised margin-based algorithm is detailed in Algorithm 2.

In fact, we use the variance and Laplacian scores as they are widely used well-known unsupervised filter methods for comparison [5, 25]. We also choose to compare our FCRSC method with the supervised mRMR method since it aims at optimizing the same twofold objective problem as our method. ReliefF, Simba-Sc, and Relief-Sc, on the other hand, are all margin-based, similarly to the proposed FCRSC. Note that Relief-Sc is somehow a previous version of FCRSC that does not detect feature redundancy; that is why it is important to compare the performance of FCRSC with Relief-Sc to show the significance of our proposed method. Finally, since we aim to position the performance of our proposed method with respect to supervised, unsupervised, and semisupervised contexts, we chose two feature selection methods from each one.

As the three semisupervised constrained algorithms Simba-Sc, Relief-Sc, and the proposed FCRSC are dependent on cannot-link constraints, we generate them in each run (similarl to [23]) as follows. A pair of data points is chosen randomly from the training set, and the class of each point of the chosen pair is checked. If it appears that the two points belong to different classes, they are accordingly added to the cannot-link constraints set. This operation is repeated until we find the needed number of constraints for each dataset.

3.3. Parameter Setting. For the constrained algorithms that use cannot-link constraints, the number of constraints was considered relatively to the number of data points available in each dataset. Thus, it was set to 20 for Wine, Ionosphere, Sonar, and Arrhythmia; 40 for WDBC; and 100 for Spambase dataset. Moreover, the number of starting points for the nonlinear optimization method (gradient ascent) in Simba-Sc is set to its default value by the authors, i.e., 5. In addition, for fair comparisons, common parameters between different algorithms were set to the same values. For instance, Laplacian score, ReliefF, Relief-Sc, and FCRSC had their neighborhood size set to 10 in all experiments (similarly to [5]) except for Spambase, which was set to 60 due to its large sample size. We also set the number of features in the subset obtained by mRMR to be equal to the number of clusters obtained by FCRSC.

3.4. Used Classifiers. We apply four different widely used classification schemes to compare, with generality, the significance of rankings or subsets by each of the used feature selection methods.

(i) K-nearest neighbor (KNN) [53]: it is a simple nonparametric method that can achieve high performance when the number of data points is sufficiently big. It utilizes only the spatial distributions of empirical samples without any previous assumptions about their class distributions, where a new data point is classified by the class of the majority of its K-nearest points. In our experiments, we use $K = 1$.

(ii) Support vector machines (SVM) [54]: they are a set of well-known general learning methods that have become very popular in the last decade. SVM classifier maximizes a margin between data points called sample margin. In our experiments, similarly to [5], we apply multiclass SVM by one-against-one method with sequential minimal optimization (SMO) solver and polynomial kernel; however, we use *fitcecoc* and *predict* Matlab functions.

(iii) Naive Bayes (NB) [55]: it is a probabilistic classifier based on Bayes theorem. It applies classification with a naive (strong) independence assumption between features. In other words, it considers that, given the class labels, features are conditionally independent of each other. WEKA's implementation with default values was used [56].

(iv) Decision tree (C4.5) [57]: it is a well-known classifier that applies an entropy based criterion to a set of training data to build the decision tree. For

instance, the data points can be split into smaller subsets by using a feature as the decision rule. For this purpose, the algorithm measures the *information gain* at each split. WEKA's implementation with default values was used.

We use more than one classifier with different decision-making natures and learning processes in order to provide a fair evaluation of used filter feature selection methods independently of the applied classification rules. Moreover, these experiments can also be eye-opening as to which classifier can be best used with the proposed feature selection method.

3.5. Evaluation Metrics. We evaluate the performance of the different used supervised, unsupervised, and semisupervised feature selection methods in terms of two aspects. One is related to the data classification accuracy obtained by applying a classifier to the selected set of features, and the second is directly related to measuring the redundancy of a feature subset or ranking.

(i) Classification accuracy: it is a supervised metric defined as a percentage of correct predictions. Thus, it evaluates how many data points were correctly classified by classifiers in Section 3.4 using the selected features.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{20}$$

where TP, TN, FP, and FN represent the numbers of true positives, true negatives, false positives, and false negatives, respectively.

(ii) Representation entropy (RE) [5, 58]: it is an unsupervised metric used to compare the redundancy in obtained feature subsets. It obtains a maximum value when all the eigenvalues become equally important, which means the level of uncertainty becomes maximum. This indicates that information is evenly distributed along all the principal directions. On the contrary, it obtains a value of zero only when all the information is present along the single principal coordinate direction (all eigenvalues are equal to zero except one).

The RE of a d-sized feature subset, denoted by H_R, is calculated as follows:

$$H_R = -\sum_{j=1}^{d} \widetilde{\lambda}_j \log \widetilde{\lambda}_j, \tag{21}$$

where $\widetilde{\lambda}_j$, i.e., one eigenvalue of the $d \times d$ covariance matrix of the respective feature space, is calculated as follows:

$$\widetilde{\lambda}_j = \frac{\lambda_j}{\sum_{j=1}^{d} \lambda_j}. \tag{22}$$

3.6. Performance Evaluation in Terms of Classification Accuracy and Feature Set Redundancy for Constrained Algorithms. According to the previously detailed experimental setup, we first compare the performance of the proposed FCRSC method with that of the constrained feature selection methods (Relief-Sc and Simba-Sc) described in Section 3.2. We chose to closely compare our method with these algorithms first as they belong to the same supervision context of the proposed method and depend on an evaluation function of similar nature.

The comparison is applied in terms of two important aspects: the classification accuracy a ranked feature set can achieve and the amount of redundancy that this set possesses. In fact, we aim at showing how FCRSC can improve the performance of its precursor Relief-Sc and compare its performance with its competitor Simba-Sc in terms of both of these aspects. In addition, C4.5 classifier was used as it cannot detect feature interactions [59], a capability that the compared Relief-based algorithms have [60].

Hence, Figures 3 and 4 show the averaged accuracy rates obtained by the C4.5 classifier on the ranked feature sets obtained by the constrained filter methods (Simba-Sc, Relief-Sc, and the proposed FCRSC) on Wine, WDBC, Ionosphere, Spambase, Sonar, and Arrhythmia datasets over 10 independent runs. In fact, each row of the two figures corresponds to one of the datasets presenting the averaged classification accuracy in the left figures and the corresponding averaged representation entropy in the right figures.

For instance, from Figures 3 and 4, we can see that generally FCRSC was always able to obtain a feature subset that provides a better classification accuracy while being less redundant. This was true except for Spambase in Figures 4(a) and 4(b), where the three algorithms performed approximately the same in terms of accuracy and redundancy, and for Ionosphere in Figures 3(e) and 3(f), where FCRSC outperformed Relief-Sc and competed with Simba-Sc. On the other hand, FCRSC proved that it can significantly improve the classification performance of its constrained antecedent Relief-Sc (e.g., Ionosphere, Sonar, and Arrhythmia) through compromising between maximum relevance and minimum redundancy in order to compose a subset that holds either weakly relevant but nonredundant features or strongly relevant ones. The fact that FCRSC is able to maintain the same classification accuracy of Relief-Sc or slightly increase it while decreasing the number of ranked features, together with the fact that the representation entropy is higher, validates the hierarchical feature clustering upon sparse graph. This shows that our suggested grouping of features and our idea of representing the feature space were in harmony with margin maximization leading to the expected behavior.To be precise, from Figure 3(a) on Wine dataset, we can see that FCRSC outperformed both constrained algorithms. Although Figure 3(b) presenting RE on Wine shows that Simba-Sc had chosen less redundant features as the first three ones, still FCRSC outperformed it in terms of classification accuracy. In addition, the results on

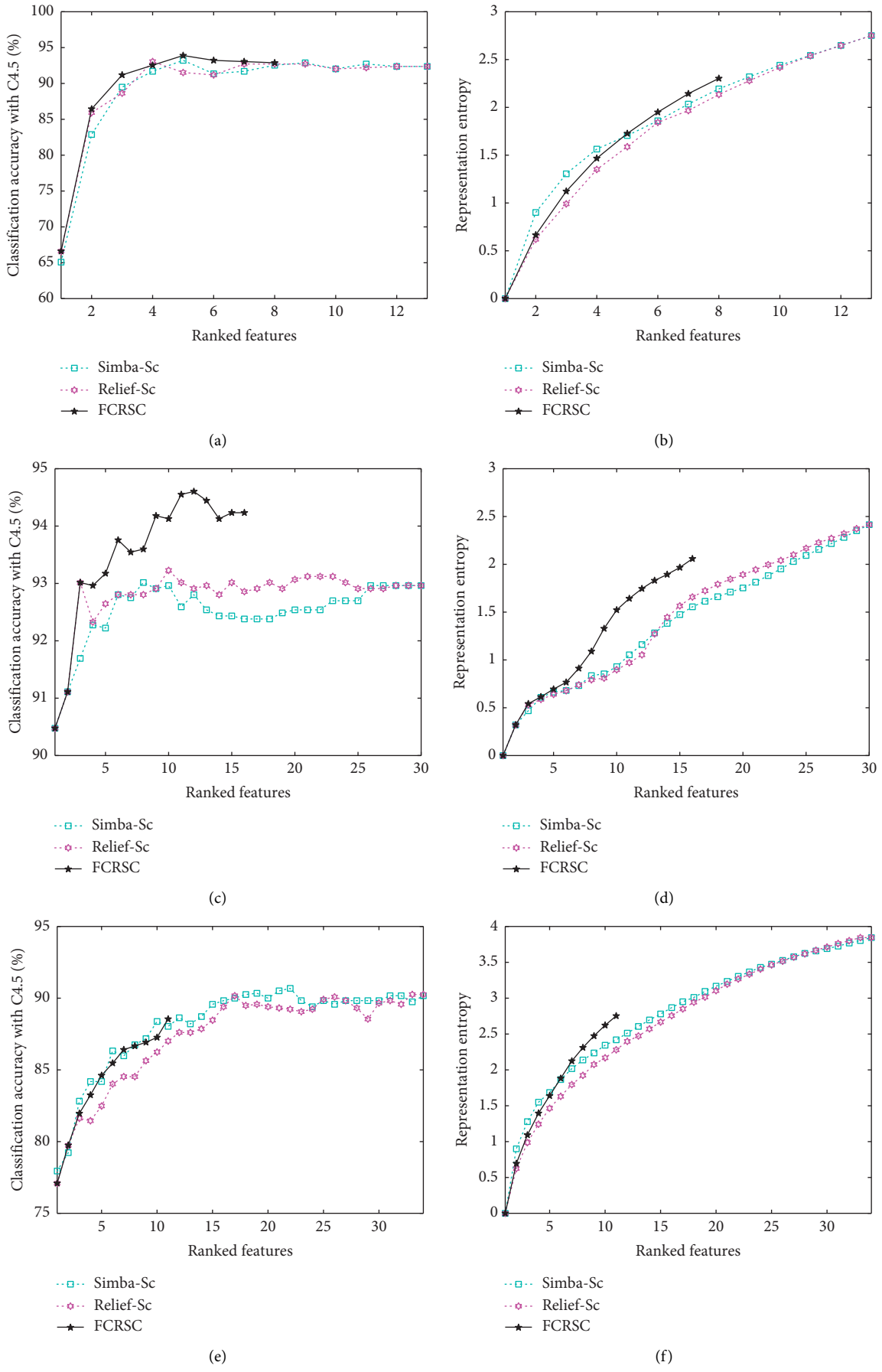

Figure 3: The averaged classification accuracy rates using C4.5 classifier vs. the number of ranked features obtained by the constrained algorithms: Simba-Sc, Relief-Sc, and the proposed FCRSC over 10 independent runs on (a) Wine, (c) WDBC, and (e) Ionosphere datasets. The averaged representation entropy (RE) of each dataset is also shown in (b), (d), and (f), respectively.

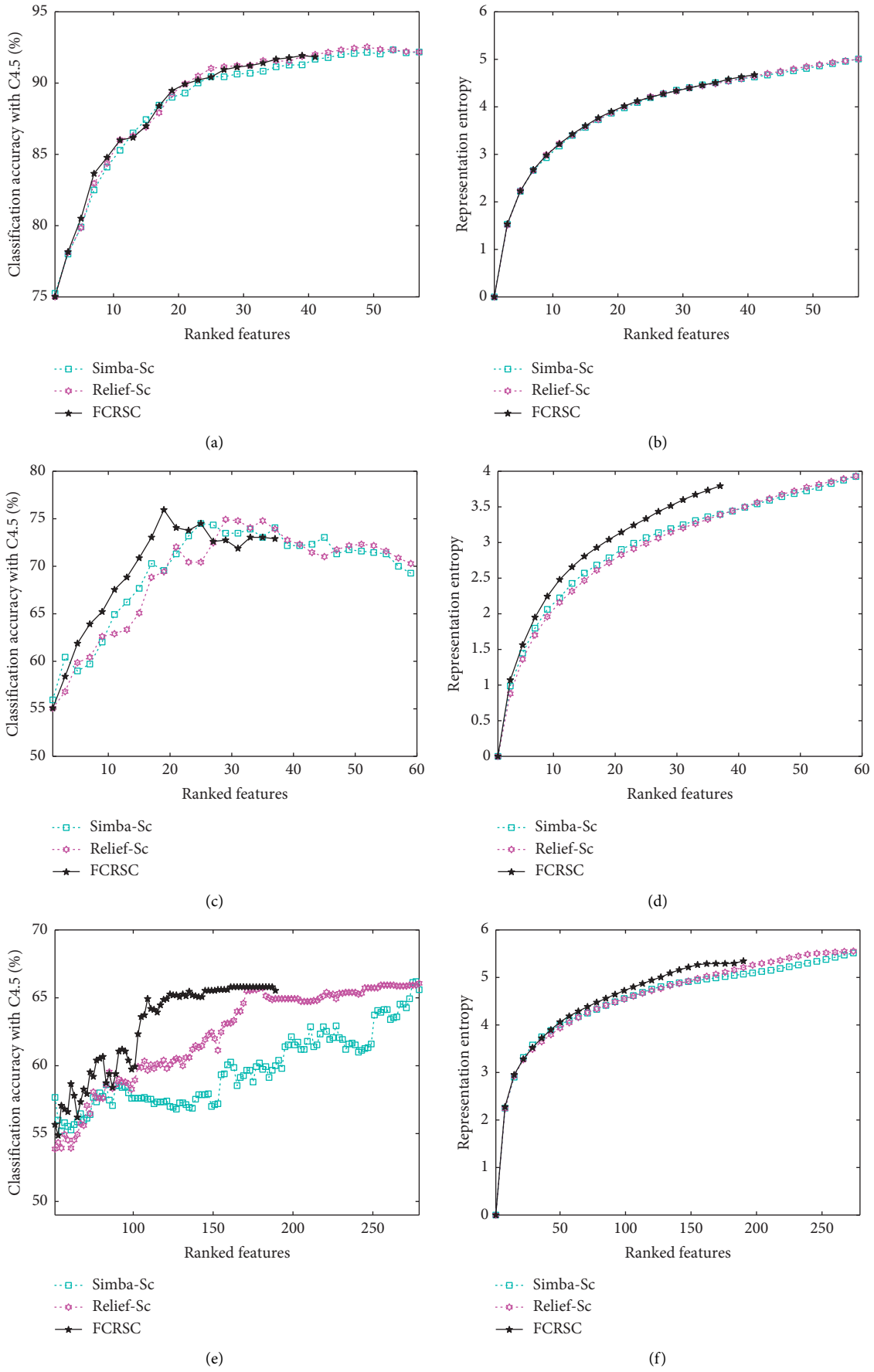

Figure 4: The averaged classification accuracy rates using C4.5 classifier vs. the number of ranked features obtained by the constrained algorithms: Simba-Sc, Relief-Sc, and the proposed FCRSC over 10 independent runs on (a) Spambase, (c) Sonar, and (e) Arrhythmia datasets. The averaged representation entropy (RE) of each dataset is also shown in (b), (d), and (f), respectively.

WDBC dataset were interesting, by which Figures 3(c) and 3(d) show that FCRSC was better than Relief-Sc and Simba-Sc in both classification accuracy and redundancy reduction. For instance, FCRSC allowed a maximum classification accuracy of 94.60% after only 12 features out of 30 in the original space. Moreover, FCRSC on Sonar dataset, as can be seen in Figures 4(c) and 4(d), outperformed Simba-Sc and Relief-Sc from the first few features until it reached its maximum of 75.94% on only 19 features out of 60 in the original space. However, later on, starting from the 27th chosen feature, Simba-Sc and Relief-Sc performed slightly better, although FCRSC maintained fewer redundant feature subsets throughout all of its ranked features. It is important to mention that FCRSC enhanced the performance of Relief-Sc over Simba-Sc approximately in all figures of Figures 3 and 4, which means that the drawback of Relief-Sc (see Section 2.3) was refrained by FCRSC. This also shows that the ability of Simba-Sc to detected redundant features was the reason for its ability to outperform Relief-Sc. This notice confirms the equal importance of redundancy removal and objective relevance when selecting features for classification. It also shows the superiority of FCRSC to its only similar-by-nature semisupervised competitors (Simba-Sc and Relief-Sc) noting the straightforwardness and easiness of reproducing the feature clustering part of FCRSC unlike Simba-Sc [23].Finally, as can be seen in Figures 4(e) and 4(f) on Arrhythmia, FCRSC clearly outperformed Simba-Sc which was not able to detect a feature subset that can at least provide a classification accuracy equivalent to the one obtained without feature selection. In fact, Relief-Sc was able to find such a subset, but its performance was lagging behind that of FCRSC. The latter was able to find a smaller feature subset with better classification accuracy and less redundancy among its features.

3.7. Performance Evaluation in Terms of Classification Accuracy Using Multiple Classifiers for the Unsupervised, Supervised, and FCRSC Algorithms. For the sake of generality, in this section, we compare the classification performance of the proposed constrained FCRSC method with that of the unsupervised and supervised methods described in Section 3.2 using three different classifiers. We use the well-known *K*NN, SVM, and NB classifiers, each depending on a different decision rule nature, to show the general positioning of FCRSC performance with respect to some of the well-known feature selection state-of-the-art algorithms. This also shows whether a performance degradation or improvement is classifier-dependent or is really imposed by the chosen feature subset.

For instance, each of Figures 5–10 corresponds to a dataset with the classification accuracies obtained by three different classifiers upon the feature subsets ranked by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC.From these figures, we can see that in general FCRSC was always able to obtain a higher accuracy curve compared to the unsupervised Variance and Laplacian scores except on WDBC where the five feature selection algorithms showed interfering and fluctuating accuracy curves from the first few ranked features as can be seen in Figure 6. FCRSC was also sometimes able to compete with supervised methods as can be seen in Figure 5 on Wine, in Figure 9 on Sonar, and in Figure 10 on Arrhythmia.

For instance, Figure 5 shows that using the three different classifiers on Wine dataset, FCRSC performed better than the unsupervised Variance score on all of them from the first few features, and it also outperformed the Laplacian score; however, the latter chose a better starting feature. This means that the locality preserving ability of the first feature was more significant for classification results than the constraint-relevance objective that is respected by FCRSC. This can be solved by improving the choice of constraints [22]. In addition, as mentioned before, FCRSC competed with the supervised ReliefF and mRMR as can be seen in Figures 5(a) and 5(b), where in fact FCRSC and mRMR performed approximately the same. This can be due to their similar behavior in compromising between maximizing relevance and minimizing redundancy.

Moreover, FCRSC on Ionosphere, as can be seen in Figure 7, clearly outperformed the unsupervised methods on the three classifiers in a very similar manner. Again, Figures 7(a) and 7(b) show close classification accuracy values recorded by the supervised mRMR and the constrained FCRSC.

On the other hand, on Spambase dataset, as can be seen in Figure 8, feature selection was generally not significant by all algorithms (the best accuracy was obtained on the original feature set) except for ReliefF and mRMR with NB classifier shown in Figure 8(c). This can be due to the fact that some datasets need all the available features to obtain best classification performance especially when they are not very high dimensional. However, the performance of FCRSC was generally stable with the three used classifiers. It lies between the supervised and unsupervised methods except for SVM classifier presented in Figure 8(b) where both the unsupervised methods together with FCRSC outperformed mRMR. In addition, the results on Sonar and Arrhythmia were very interesting. As can be seen in Figure 9(a) on Sonar dataset, both Variance and Laplacian scores reached their highest accuracy rates (82.03%) on the full feature space of Sonar dataset, i.e., 60 features, whereas FCRSC obtained an accuracy of 83.04% over only 37 features. This shows that, in this case, the unsupervised Variance and Laplacian scores could not obtain a smaller feature subset that can provide a similar or better classification compared to the original one. It is important here to note that the performance degradation of FCRSC that appeared between approximately the 27th and 37th features on Sonar with C4.5 classifier (analyzed in the previous section and presented in Figure 4(c)) was classifier-related since a good performance was obtained for the same features using *K*NN, SVM, and NB classifiers.

On the other hand, Figure 10 on Arrhythmia also shows that FCRSC outperforms the unsupervised algorithms and competes with the supervised ones. In fact, it was also able to obtain either a similar or a higher classification accuracy compared to the one obtained by the full feature space with approximately only half the number of features. For example, an accuracy of 55.3% on 117 feature was recorded

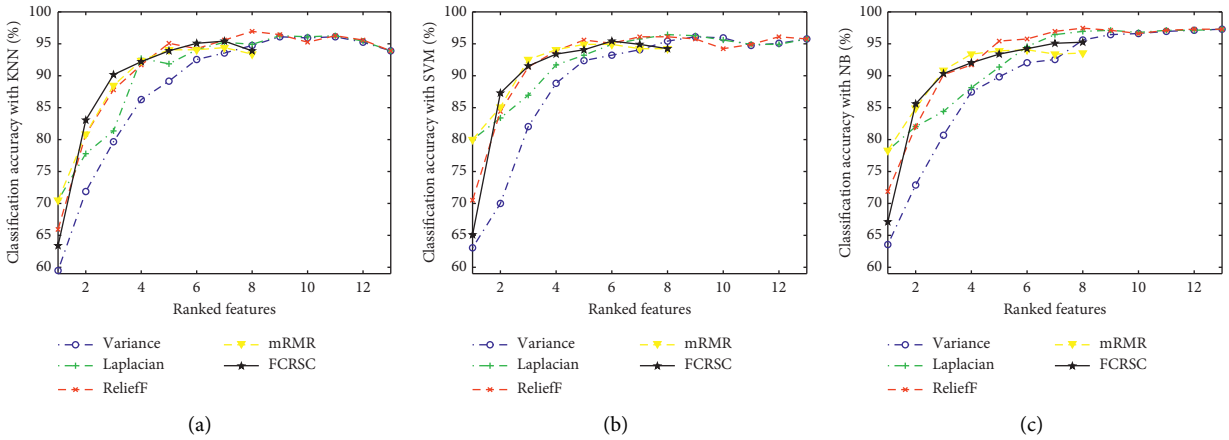

Figure 5: The averaged classification accuracy rates using (a) KNN, (b) SVM, and (c) NB classifiers vs. the number of ranked features obtained by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC over 10 independent runs on Wine dataset.

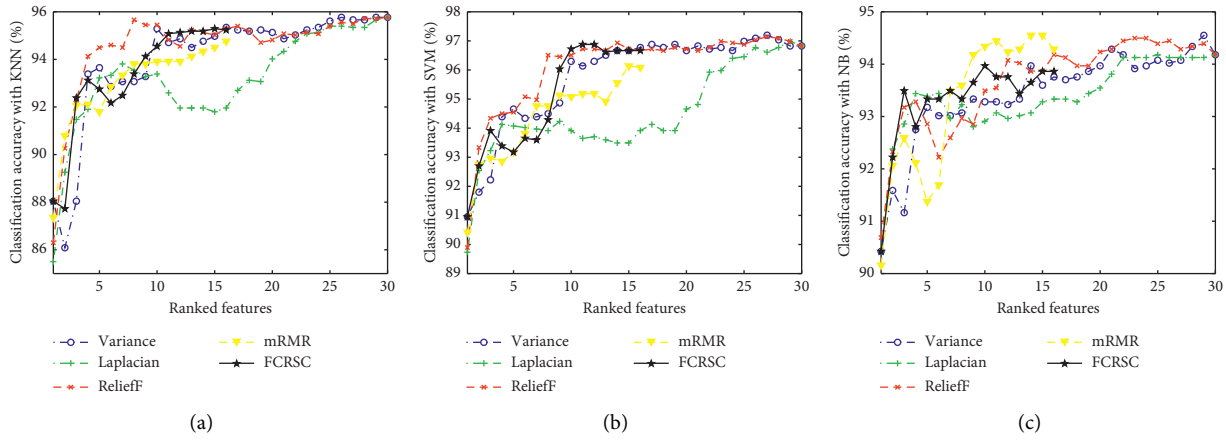

Figure 6: The averaged classification accuracy rates using (a) KNN, (b) SVM, and (c) NB classifiers vs. the number of ranked features obtained by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC over 10 independent runs on WDBC dataset.

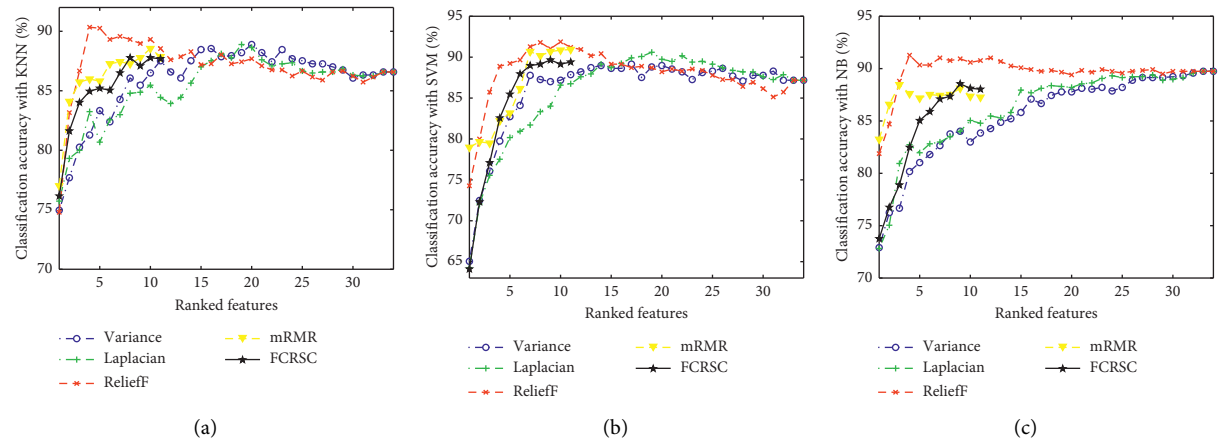

Figure 7: The averaged classification accuracy rates using (a) KNN, (b) SVM, and (c) NB classifiers vs. the number of ranked features obtained by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC over 10 independent runs on Ionosphere dataset.

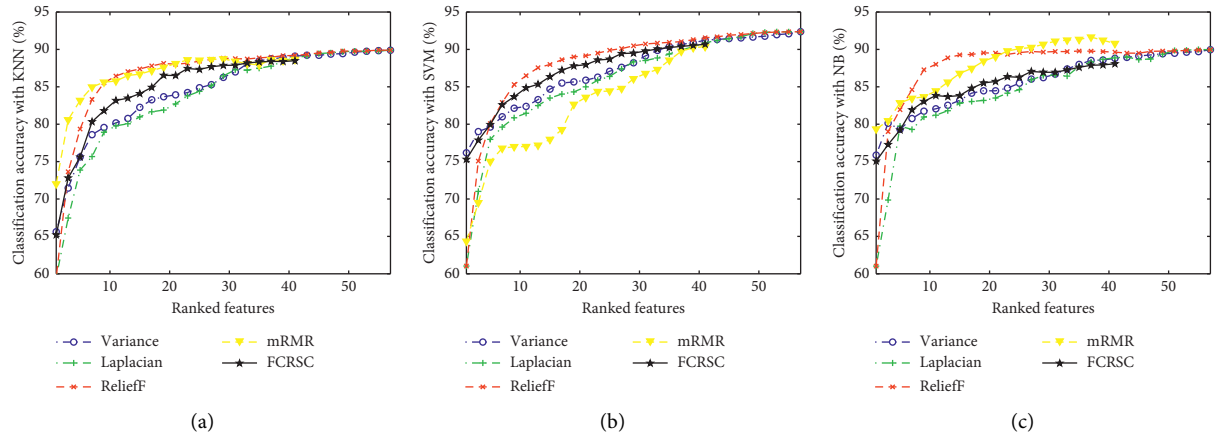

Figure 8: The averaged classification accuracy rates using (a) KNN, (b) SVM, and (c) NB classifiers vs. the number of ranked features obtained by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC over 10 independent runs on Spambase dataset.

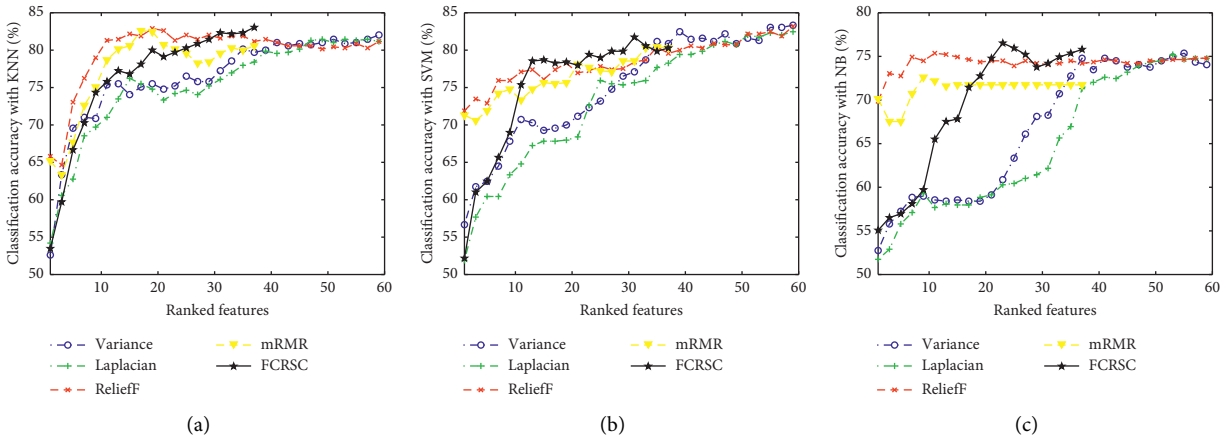

Figure 9: The averaged classification accuracy rates using (a) KNN, (b) SVM, and (c) NB classifiers vs. the number of ranked features obtained by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC over 10 independent runs on Sonar dataset.

using FCRSC with *K*NN (figure 10(a)) compared to 52.93% on 279 features without feature selection and 55.5% on 141 features using ReliefF and mRMR.

In conclusion, FCRSC aims at finding a relevant and nonredundant feature subset that is said to either maintain the classification accuracy obtained on the full feature space or provide an enhanced accuracy performance (through removing the irrelevant and noisy features). It is important to note that although FCRSC chooses a final subset of features ($\text{size}(\mathbf{F}_s) < F$), it is still a ranking feature selection method, which means, similarly to the other feature selection methods mentioned in this paper, subsets that are also smaller than $\mathbf{F}_s$ can be obtained. Moreover, as FCRSC utilizes the Relief-Sc algorithm for its margin maximization objective, it was clear from the results that removing redundant features in addition to irrelevant ones allows better classification performance. Finally, using more than one classifier with different decision-making natures provided a fair evaluation of the used filter feature selection methods and proved the general independence between them and the classifiers.

3.8. Execution Time Comparison. In this section, we consider the different supervised, unsupervised, and semisupervised feature selection methods from the execution time perspective. Thus, Table 2 presents the average execution time (in ms) of each feature selection method mentioned in Section 3.2 over 10 independent runs.

For instance, from this table we can see that variance score, as expected, had the least execution time with the least differences between datasets since it is the simplest method among all. However, Laplacian, ReliefF, and Simba-Sc appeared to have a great increase in execution time as the number of data points increase significantly as was the case between Spambase (4601 data points) and Sonar (208 data points) with approximately equal number of features. Although Relief-Sc and FCRSC also had a significant increase in the latter case, this increase was less steep. Noting that Simba-Sc, Relief-Sc, and FCRSC are all dependent on constraints and are provided with the same number on each dataset, we declare that Simba-Sc consumes much more time due to it being repeated from 5 different starting points (gradient ascent method) to optimize its objective function.

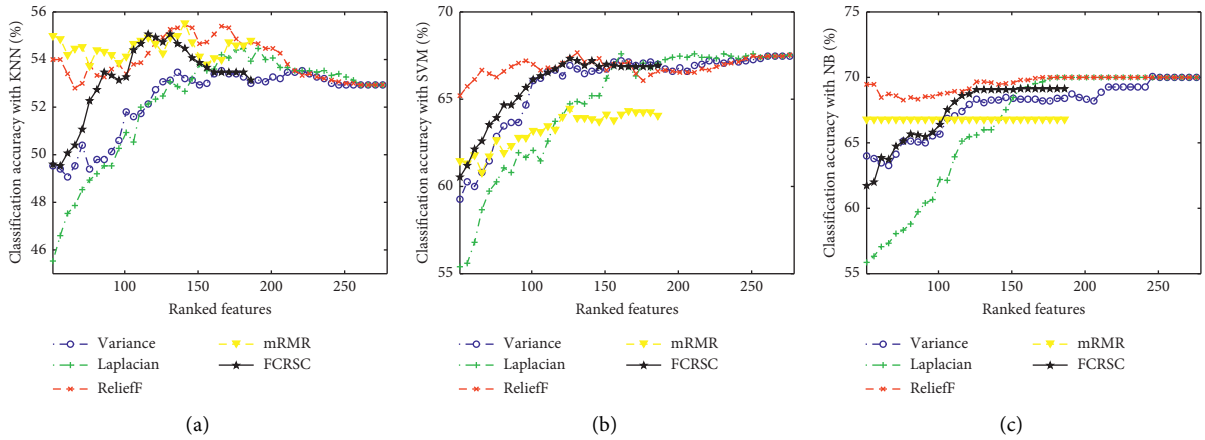

Figure 10: The averaged classification accuracy rates using (a) KNN, (b) SVM, and (c) NB classifiers vs. the number of ranked features obtained by variance score, Laplacian score, ReliefF, mRMR, and the proposed FCRSC over 10 independent runs on Arrhythmia dataset.

Table 2: Average execution time (in ms) of the different unsupervised, supervised, and semisupervised feature selection methods over 10 independent runs.

Dataset	Feature selection method						
	Unsupervised		Supervised		Semisupervised		
	Variance	Laplacian	ReliefF	mRMR	Simba-Sc	Relief-Sc	FCRSC
Wine	0.41	9.06	71.62	13.66	569.04	1.70	9.61
WDBC	0.81	73.68	205.70	43.25	3166.88	5.97	47.31
Ionosphere	0.34	29.66	128.20	28.39	1032.83	2.50	41.80
Spambase	1.92	5408.62	15290.15	342.65	67081.99	755.62	1905.42
Sonar	0.59	16.06	108.83	35.76	694.99	2.73	89.55
Arrythmia	2.46	93.64	4727.31	446.71	1336.32	5.47	1463.52
Average	1.09	938.45	3421.97	151.73	12313.67	129.00	592.87

On the other hand, as the number of features increased significantly between 13 for Wine and 60 for Sonar (having close number of data points and the same number of provided constraints, i.e., 20), the execution time by all the algorithms increased reasonably; however, Relief-Sc increased very little compared to FCRSC, and this is due to the time needed by FCRSC to apply the steps of feature clustering and cluster-representative choosing.

Hence, the overall average execution time for each method on all datasets, as can be seen in the last row of Table 2, shows that FCRSC was generally faster than Laplacian, ReliefF, and Simba-Sc. Thus, in terms of computational complexity, we can say that FCRSC is mainly related to the following:

(i) The number of cannot-link constraints $|\mathscr{C}|$

(ii) The number of data points N

(iii) The dimension of feature space F

In big O notation, the constrained Relief-Sc can be calculated in $O(|\mathscr{C}|\mathrm{NF})$. Furthermore, the ranking step that is used within all ranking feature selection methods needs $O(F\log(F))$. The proposed FCRSC is divided into multiple steps, some of which can be done in parallel (clustering the features and calculating their margin weights). For instance, the construction of the sparse graph costs $O(F^2)$ [36], and the single-linkage hierarchical clustering of features also needs $O(F^2)$. Hence, the overall computational complexity by FCRSC can be calculated as $O(\mathrm{MAX}(|\mathscr{C}|\mathrm{NF}, F^2, F\log(F)))$.

4. Conclusion

In this work, a semisupervised feature selection approach was proposed. The main contribution is the novel combination of feature clustering upon a sparse graph with a margin-based objective function called FCRSC.

This approach is said to handle the two core aspects of feature selection (relevance and redundancy) in three main building blocks where it first constructs the similarity matrix between features through sparse representation. Second, on top of the latter, feature clustering is applied simultaneously with the application of the margin-based algorithm Relief-Sc. Finally, FCRSC obtains its final feature subset by choosing the feature that most enlarges the margin from each cluster of features, hence maximizing relevance while minimizing redundancy. The performance of this approach was compared to that of supervised, unsupervised, and semisupervised filter feature selection methods using four different classification schemes on six UCI well-known benchmark machine learning datasets. The results showed

the satisfactory performance of FCRSC that outperformed the unsupervised and semisupervised methods on most datasets and also competed with the supervised ones.

We believe that this research can be eye-opening to some interesting future work tackling some limitations or ideas that were not covered in this paper. This includes, first, tailoring the work to a specific application like document classification where the used cannot-link constraints can be domain-specific and actively chosen (instead of being randomly selected as in this work) which we believe would enhance their quality and thus enhance the classification performance of FCRSC. Second, the problem of dendrogram cutoff choice for finding the final clustering solution can sometimes be tricky; this can hold more analysis with a domain-specific dataset. Although in our work we empirically chose cutoff when clusters of clusters start to be grouped together, it can be a drawback when the feature space becomes too large.

Data Availability

The data used to support the findings of this study are openly available in UCI archive.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was fully funded by the Image Processing and Computer Graphics Department at Ho Chi Minh City Open University in Vietnam.

References

[1] M. Liu and D. Zhang, "Sparsity score: a novel graph-preserving feature selection method," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 4, Article ID 1450009, 2014.

[2] J. Cai, J. Luo, S. Wang et al., "Feature selection in machine learning: a new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[3] Y. Zhang, S. Cheng, Y. Shi et al., "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Systems with Applications*, vol. 137, pp. 46–58, 2019.

[4] J. Jiye Liang, F. Feng Wang, C. Chuangyin Dang et al., "A group incremental approach to feature selection applying rough set technique," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 294–308, 2014.

[5] P. Moradi and M. Rostami, "A graph theoretic approach for unsupervised feature selection," *Engineering Applications of Artificial Intelligence*, vol. 44, pp. 33–45, 2015.

[6] H. Xie, L. Zhang, C. P. Lim et al., "Feature selection using enhanced particle swarm optimisation for classification models," *Sensors*, vol. 21, no. 5, p. 1816, 2021.

[7] Y. Zhang, D.-w. Gong, X.-z. Gao et al., "Binary differential evolution with self-learning for multi-objective feature selection," *Information Sciences*, vol. 507, pp. 67–85, 2020.

[8] Z. Zhang and E. R. Hancock, "Kernel entropy-based unsupervised spectral feature selection," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 5, Article ID 1260002, 2012.

[9] Y. Zhang, Q. Wang, D.-w. Gong, and X.-f. Song, "Nonnegative Laplacian embedding guided subspace learning for unsupervised feature selection," *Pattern Recognition*, vol. 93, pp. 337–352, 2019.

[10] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016.

[11] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proceedings of the 18th International Conference on Neural Information Processing Systems, Advances in Neural Information Processing Systems*, vol. 18, pp. 507–514, Vancouver, Canada, December 2006.

[12] L. M. Q. Abualigah, *Introduction, Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering*, Springer, Berlin, Germany, 2019.

[13] X. Li, P. Yi, W. Wei et al., "LNNLS-KH: A Feature Selection Method for Network Intrusion Detection," *Security and Communication Networks*, vol. 2021, Article ID 8830431, 2021.

[14] H. Hanchuan Peng, F. Fuhui Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[15] J. Martínez Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognition*, vol. 43, no. 6, pp. 2068–2081, 2010.

[16] Y. Zhou, W. Zhang, J. Kang, X. Zhang, and X. Wang, "A problem-specific non-dominated sorting genetic algorithm for supervised feature selection," *Information Sciences*, vol. 547, pp. 841–859, 2021.

[17] U. Kaya and M. Fidan, "Parametric and nonparametric correlation ranking based supervised feature selection methods for skin segmentation," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021.

[18] J. Li, K. Cheng, S Wang et al., "Feature selection: a data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 94, 2018.

[19] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proceedings of the 2007 SIAM International Conference on Data Mining, SIAM*, pp. 641–646, Minneapolis, MN, USA, April 2007.

[20] R. Sheikhpour, M. A. Sarram, S. Gharaghani et al., "A survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, pp. 141–158, 2017.

[21] Q. Pang and L. Zhang, "A recursive feature retention method for semi-supervised feature selection," *International Journal of Machine Learning and Cybernetics*, pp. 1–19, 2021.

[22] S. Hijazi, D. Hamad, M. Kalakech et al., "Active learning of constraints for weighted feature selection," *Advances in Data Analysis and Classification*, vol. 15, pp. 337–377, 2021.

[23] M. Yang and J. Song, "A novel hypothesis-margin based approach for feature selection with side pairwise constraints," *Neurocomputing*, vol. 73, no. 16, pp. 2859–2872, 2010.

[24] K. Benabdeslem and M. Hindawi, "Efficient semi-supervised feature selection: constraint, relevance, and redundancy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1131–1143, 2014.

[25] D. Zhang, S. Chen, and Z.-H. Zhou, "Constraint score: a new filter method for feature selection with pairwise constraints," *Pattern Recognition*, vol. 41, no. 5, pp. 1440–1451, 2008.

[26] L. Zheng, F. Chao, N. M. Parthaláin, D. Zhang, and Q. Shen, "Feature grouping and selection: a graph-based approach," *Information Sciences*, vol. 546, pp. 1256–1272, 2021.

[27] H. Liu, X. Wu, and S. Zhang, "Feature selection using hierarchical feature clustering," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM*, pp. 979–984, Scotland, UK, October 2011.

[28] X. Zhao, W. Deng, and Y. Shi, "Feature selection with attributes clustering by maximal information coefficient," *Procedia Computer Science*, vol. 17, pp. 70–79, 2013.

[29] C. Krier, D. François, F. Rossi et al., "Feature clustering and mutual information for the selection of variables in spectral data," in *ESANN European Symposium on Artificial Neural Networks*, pp. 157–162, Bruges, Belgium, April 2007.

[30] J. Jiao, X. Mo, and C. Shen, "Image clustering via sparse representation," in *International Conference on Multimedia Modeling, Lecture Notes in Computer Science*, pp. 761–766, Springer, Chongqing, China, January 2010.

[31] D. M. Witten and R. Tibshirani, "A framework for feature selection in clustering," *Journal of the American Statistical Association*, vol. 105, no. 490, pp. 713–726, 2010.

[32] D. Ienco and R. Meo, "Exploration and reduction of the feature space by hierarchical clustering," in *Proceedings of the 2008 SIAM International Conference on Data Mining, SIAM*, pp. 577–587, Atlanta, Georgia, Apri 2008.

[33] S. Hijazi, M. Kalakech, D. Hamad, and A. Kalakech, "Feature selection approach based on hypothesis-margin and pairwise constraints," in *Communications Conference (MENACOMM), IEEE Middle East and North Africa*, pp. 1–6, IEEE, Lebanon, Lebanese Republic, April 2018.

[34] X. Huang, L. Zhang, B. Wang, Z. Zhang, and F. Li, "Feature weight estimation based on dynamic representation and neighbor sparse reconstruction," *Pattern Recognition*, vol. 81, pp. 388–403, 2018.

[35] Y. Zhu, X. Zhang, R. Wang, W. Zheng, and Y. Zhu, "Self-representation and PCA embedding for unsupervised feature selection," *World Wide Web*, vol. 21, no. 6, pp. 1675–1688, 2018.

[36] M. Liu, D. Sun, and D. Zhang, "Sparsity score: a new filter feature selection method based on graph," in *Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE*, pp. 959–962, Tsukuba Science City, JAPAN, November 2012.

[37] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 331–341, 2010.

[38] J. Xu, G. Yang, H. Man et al., "L 1 graph based on sparse coding for feature selection," in *International Symposium on Neural Networks, Advances in Neural Networks—ISNN 2013*, pp. 594–601, Springer, Dalian, China, July 2013.

[39] X. Zhu, X. Li, S. Zhang et al., "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1263–1275, 2017.

[40] C. Chenping Hou, F. Feiping Nie, X. Xuelong Li, D. Dongyun Yi, and Y. Yi Wu, "Joint embedding learning and sparse regression: a framework for unsupervised feature selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 793–804, 2014.

[41] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in Neural Information Processing Systems*, pp. 849–856, MA, USA, January 2002.

[42] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[43] J. Liu, S. Ji, J. Ye et al., *SLEP: Sparse Learning with Efficient Projections*, Vol. 6, Arizona State University, Tempe, AZ, USA, 2009.

[44] W. H. Wai-Ho Au, K. C. C. Chan, A. K. C. Wong et al., "Attribute clustering for grouping, selection, and classification of gene expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 2, pp. 83–101, 2005.

[45] T. Ronan, Z. Qi, and K. M. Naegle, "Avoiding common pitfalls when clustering biological data," *Science Signaling*, vol. 9, no. 432, p. re6, 2016.

[46] M. Liu and D. Zhang, "Pairwise constraint-guided sparse learning for feature selection," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 298–310, 2016.

[47] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the Ninth International Workshop on Machine Learning*, pp. 249–256, Aberdeen, UK, July 1992.

[48] I. Kononenko, "Estimating attributes: analysis and extensions of relief," in *Proceedings of the European Conference on Machine Learning, Machine Learning: ECML-94*, pp. 171–182, Springer, Catania, Italy, April 1994.

[49] Y. Sun and J. Li, "Iterative relief for feature weighting," in *Proceedings of the 23rd International Conference on Machine Learning, ACM*, pp. 913–920, Pittsburgh, PA, USA, June 2006.

[50] D. Dua and E. Karra Taniskidou, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml.

[51] C. M. Bishop, *Neural Networks: A Pattern Recognition Perspective*, Oxford University Press, New York, NY, USA, 1996.

[52] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine Learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[53] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[54] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.

[55] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, Morgan Kaufmann, Montreal, Canada, August 1995.

[56] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[57] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[58] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A Review of Unsupervised Feature Selection Methods," *Artificial Intelligence Review*, vol. 53, pp. 907–948, 2019.

[59] A. K. Shekar, T. Bocklisch, P. I. Sánchez, C. N. Straehle, and E. Müller, "Including multi-feature interactions and redundancy for feature ranking in mixed datasets," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 239–255, Springer, Skopje, Macedonia, September 2017.

[60] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with relieff," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997.