The Journal of Supercomputing (2024) 80:24212–24240
https://doi.org/10.1007/s11227-024-06368-3

A taxonomy of unsupervised feature selection methods including their pros, cons, and challenges

Rajesh Dwivedi[1] · Aruna Tiwari[1] · Neha Bharill[2] · Milind Ratnaparkhe[3] · Alok Kumar Tiwari[4]

Accepted: 14 July 2024 / Published online: 22 July 2024

Abstract
In pattern recognition, statistics, machine learning, and data mining, feature or attribute selection is a standard dimensionality reduction method. The goal is to apply a set of rules to select essential and relevant features from the original dataset. In recent years, unsupervised feature selection approaches have garnered significant attention across various research fields. This study presents a well-organized summary of the latest and most effective unsupervised feature selection techniques in the scientific literature. We introduce a taxonomy of these strategies, elucidating their significant features and underlying principles. Additionally, we outline the pros, cons, challenges, and practical applications of the broad categories of unsupervised feature selection approaches reviewed in the literature. Furthermore, we conducted a comparison of several state-of-the-art unsupervised feature selection methods through experimental analysis.

Keywords Unsupervised feature selection · Filter method · Wrapper method · Hybrid method · Embedded method · Clustering

1 Introduction

Feature selection refers to the procedure of picking a portion of the original features based on their significance and redundancy. According to Yu and Liu [74], the feature subsets can be categorized into four groups: (1) highly relevant feature subsets, (2) weakly relevant and nonredundant feature subsets, (3) weakly relevant and redundant feature subsets, and (4) completely irrelevant and noisy feature subsets, as shown in Fig. 1. A feature is irrelevant if it does not contribute to the accuracy of the prediction. To construct a decent prediction model, choosing all highly relevant and some weakly relevant features is desirable while excluding irrelevant, redundant, or noisy features. Sometimes, weakly relevant features that are nonredundant

Extended author information available on the last page of the article

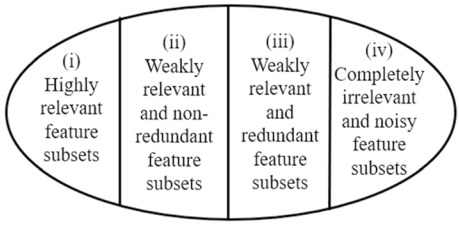

Fig. 1 Categories of feature subsets

and compatible with assessment methods can also help to improve the prediction accuracy. During the feature selection process, redundant ones are usually thrown out because they may have critical statistical relationships with other features, not because they have information that isn't useful. Sometimes, a feature may be unimportant as a standalone entity, but it might be beneficial when paired with other features.

In machine learning, the experimental data may be unlabeled, labeled, or partially labeled. This makes it possible to use unsupervised, supervised, and semisupervised feature selection techniques to select the essential and relevant features. Usually, labeled data are a collection of samples that are annotated with meaningful labels. Supervised feature selection refers to the procedure of picking a group of features based on a set of criteria for figuring out the value and importance of the features. On the other hand, unlabeled data are made up of samples and things that can be seen without labels. Unsupervised feature selection, in which you don't know anything about the underlying functional classes ahead of time, uses data structures like data variance, separability, and distribution to figure out the importance of each feature. In semisupervised feature selection, some portion of labeled data is added to unlabeled data as extra information to make an unsupervised feature selection work better. Nowadays, a lot of literature is available within the scope of supervised and semisupervised feature selection. However, unsupervised feature selection is relatively less explored. So, in this paper, our primary focus is to explore unsupervised feature selection (UFS) approaches [66].

In unsupervised learning, clustering or grouping [23], is the primary operation performed on the unlabeled data to identify the essential clusters. Clustering can be negatively impacted by extraneous and redundant data features, which can deteriorate the cluster quality, lead to extensive computation costs, and increase memory needs. Consequently, to improve the performance, UFS is performed to get rid of redundant and unimportant features. To illustrate this concept, we provided Figs. 2, 3, and 4, which show the clustering of a dataset by taking different feature subsets. Figure 2 shows that f_1 is adequate for identifying distinct clusters. However, Fig. 3 shows that f_3 is redundant and negatively affects the homogeneity of clusters. In Fig. 4, it is shown that f_2 is unimportant and has no effect on the clustering process at all because f_1 is alone capable of identifying the distinct cluster. In addition, various subsets of characteristics, including pertinent information, may provide varying degrees of clustering.

UFS is vital when dealing with large-scale and high-dimensional data. In such cases, the existence of irrelevant and duplicated features can significantly impair

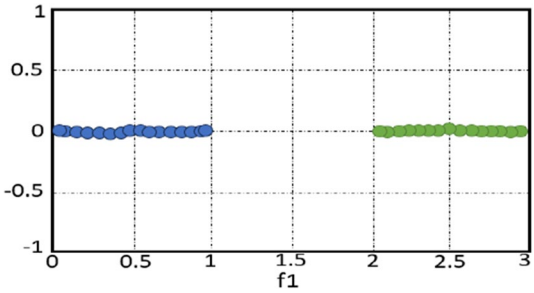

Fig. 2 Clustering by taking only f_1

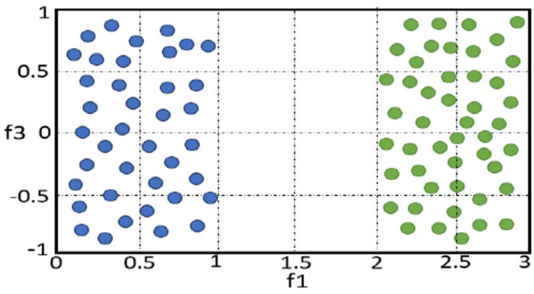

Fig. 3 Clustering by taking f_1 and f_3

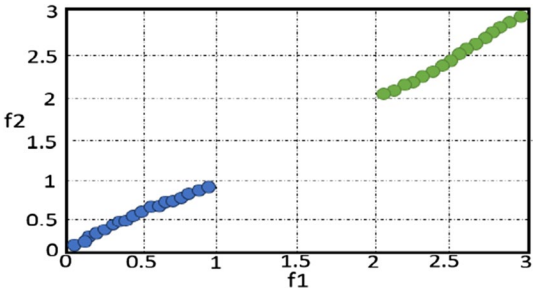

Fig. 4 Clustering by taking f_1 and f_2

the efficiency of clustering methods. In contrast to supervised learning, which relies on labeled data to determine important features, unsupervised learning faces the difficulty of feature selection without such assistance. The importance of UFS stems from its capacity to augment the caliber of clusters by prioritizing the most informative characteristics, hence enhancing the comprehensibility and effectiveness of the learning process.

Furthermore, UFS has the ability to greatly decrease the computational load that is typically linked to datasets with a large number of dimensions. Through the removal of superfluous characteristics, UFS not only enhances the efficiency of the clustering process but also diminishes memory usage, hence enabling the handling of larger datasets. The decrease in dimensionality might result in more resilient and significant cluster formations, which are essential for applications in diverse domains like bioinformatics, text mining, and image analysis.

Therefore, to explore the various methodologies of UFS, we conducted a comprehensive and structured evaluation, ranging from fundamental to cutting-edge approaches, to provide an overview of UFS methods. We outlined their primary features and the underlying concepts upon which these approaches are founded. Additionally, we presented a taxonomy of the reviewed UFS approaches, categorizing them based on their methodology, type, and subtype. We also highlighted the benefits, drawbacks, and challenges. Moreover, through experimental finding, we compared the state-of-the-art UFS methods.

A thorough assessment of these methods is required to provide a clear knowledge of the current state of the art in UFS, highlight the strengths and weaknesses of existing techniques, and propose prospective topics for future research. This review intends to be a significant resource for scholars and practitioners by providing insights into the many methods used in UFS, the settings in which they are most effective, and the trade-offs associated with their implementation. In this review, we propose to advance the field of unsupervised learning by encouraging the development of more effective and efficient feature selection techniques.

The rest of the paper is organized as follows: in Sect. 2, we presented the development process for feature selection. Section 3 introduces various UFS approaches and their taxonomy classifications. The benefits and drawbacks of UFS are summarized in Sect. 4. In Sect. 5, we compared state-of-the-art UFS approaches through experimental findings. Section 6 discusses the practical applications of UFS methods. Section 7 addresses the challenges of UFS approaches. Finally, in Sect. 8, we provided a summary and suggested future directions.

2 Development of unsupervised feature selection

The development of the feature selection process consists of five steps: search direction, search strategy, evaluation criteria, stopping criterion, and result validation, as shown in Fig. 5. These steps are discussed in detail subsequently.

First step:

The first step of the feature selection procedure is to determine the starting point and the search direction. Two ways are available for this process: forward search and backward search. In the forward search, the construction of the feature subset begins with a null subset, and then adding the features occurs in successive iterations. On

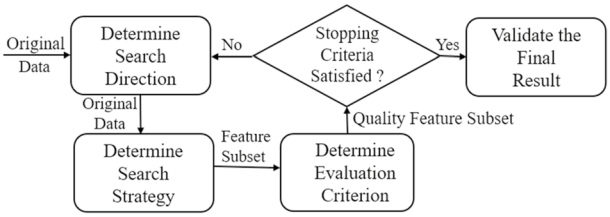

Fig. 5 Development of feature selection process

the other hand, in backward search, the process starts with a complete set of features, and then the elimination of features happens in successive iterations.

Second step:

The second step of the feature selection process determines the search strategy. There are three categories of search techniques: sequential, randomized, and exponential. The sequential search strategy is also called "greedy hill-climbing search," in which the addition of one feature happens at a time. The typically used sequential search approaches are sequential forward selection (SFS) and sequential backward selection (SBS). These search strategies are easy to implement, and the complexity of these strategies is proportional to the number of features. It can handle problems with multiple features that are similar. However, these methods do not work well with indices that aren't monotonic, and they may produce the nesting effect because once a feature is inserted (or deleted), it can't be deleted (or inserted) again. Also, they are sensitive to how features interact, so it's easy for them to get stuck in local minima. To resolve this issue, sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) were introduced by giving users ways to reselect deleted features and remove already included features. A few examples of the sequential search strategy are beam search, best-first search, an improved version of best-first search, and the plus-l take-away-r algorithm (PTA).

Another search technique is the randomized search method, which selects features at random and then employs two distinct search methods. Firstly, it uses search methods like simulated annealing and random hill-climbing that work in a sequential or two-way manner. Secondly, it uses search strategies that don't follow a linear approach, like the genetic algorithm (GA), the Las Vegas algorithm, and the Tabu search.

The exponential search begins with the original features and finds the best solution. But this strategy is hard to use and takes a lot of computing power, particularly for high-dimension datasets. An illustration of this technique is exhaustive searching, which looks at all possible subsets to find the best one.

Third step:

The third step of the feature selection process is the determination of evaluation methods. In this step, the selected subset of features is examined based on specific evaluation method. There are four types of evaluation methods for choosing features: filter, wrapper, hybrid, and embedded, which are elaborated in detail in Sect. 3.

Fourth step:

The fourth step of the feature selection task is determining the stopping criteria. It determines when the feature selection process should end. A selection of good stopping criterion can avoid overfitting, making finding the best feature subset easier and more effective. Decisions taken in earlier phases affect the choice of a termination criterion. Common cutoffs include reaching a certain number of features or iterations, getting better by a certain percentage between iterations, or getting an ideal feature subset formed on some evaluation function.

Fifth step:

The fifth step of the feature selection procedure is validation. Various validation techniques have been proposed to test how well potential feature sets work for the learning algorithm. In the supervised context, the most common ways to estimate

error are cross-validation (CV) and performance measurements based on a confusion matrix. On the other hand, in an unsupervised context, the Rand index and the Jaccard index are used to measure similarity. In previous studies, some additional validation and analysis have also been done. For example, the Kuncheva index (KI) is used to measure stability, and the analysis of variance (ANOVA) is used to measure complexity. The various existing UFS approaches are discussed subsequently.

3 Unsupervised feature selection methods and their types and taxonomy

As previously mentioned, unsupervised feature selection (UFS) methods can be categorized into four types based on evaluation criteria: filter, wrapper, hybrid, and embedded methods. In this section, we discussed each of these feature selection methods in detail and proposed a taxonomy, illustrated in Fig. 6, to organize the various UFS approaches described in the literature. Following this, we explored these approaches by focusing on the underlying concepts and highlighting their primary qualities.

3.1 Unsupervised filter method

The first evaluation method is the filter method, where feature relevance is measured using four distinct categories of evaluation measures: information, distance, consistency, and dependency. Since the filter method doesn't depend on any learning algorithm, it can be used to find general solutions for different classifiers or clustering techniques. The filter method is the oldest and is also called an open-loop method. The working of the filter method is shown in Fig. 7.

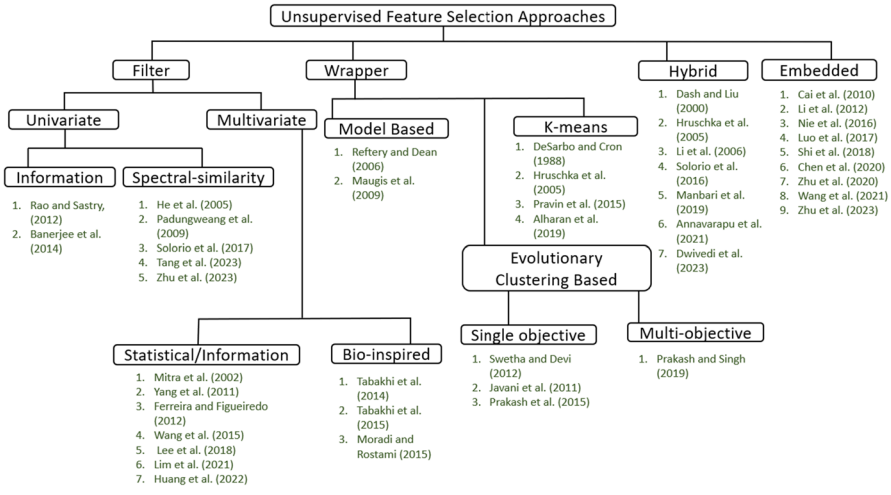

Fig. 6 Taxonomy of unsupervised feature selection approaches

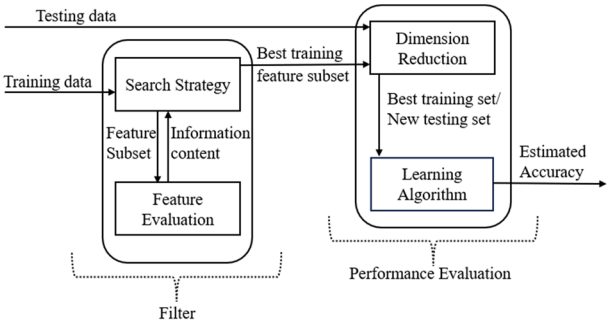

Fig. 7 Filter method for feature selection

The unsupervised filter methods are divided into two parts: the univariate filter method and the multivariate filter method, which are discussed in detail subsequently.

3.1.1 Univariate filter method

In this method, we investigate each feature separately without considering the correlations between features, using various metrics such as mutual information, variance, and Fisher score. The major drawback of this method is that it can select redundant features due to the ignorance of correlations among features. The univariate filter method is divided into two categories: information-based and spectral-similarity-based methods. The information-based method uses information theory to determine the importance of feature subsets, while the spectral-similarity-based method uses spectral analysis and object similarity to assess feature subset importance. These methods are discussed in detail subsequently.

Information-based method

The information-based method identifies cluster patterns in the data by analyzing the extent of data dispersion using metrics such as entropy, mutual information, divergence, and so on. In 2012, Rao and Shastri [59] proposed a univariate filter approach for feature selection in which they employed information theory to rank the feature. They weighted each feature, adopting the idea of representation entropy. The representation entropy is an estimate of data compression derived from the covariance matrix's eigenvalues' entropy. It ranges from 0 to 1, with 1 denoting the highest compression and 0 denoting the lowest compression. Later on, in 2014, Banerjee et al. [6] proposed another UFS method based on singular value decomposition entropy (UFS-SVD). In this method, they selected features by evaluating the entropy of the initial data matrix by observing its singular values. The entropy ranges from 0 to 1. When the entropy value approaches 0 (low), it indicates that the data matrix's spectrum is not uniformly scattered, leading to the formation of a well-defined cluster. On the other hand, when the entropy value is close to 1 (high), the spectrum of the data matrix is consistently scattered, and clustering is not well-defined.

Spectral-similarity-based method

Spectral feature selection approaches detect local or global data structures using the normalized Laplacian score (LS) or eigensystem of Laplacian derived from an object similarity matrix. In 2005, He et al. [27] said that the LS is the most prominently used and helpful in univariate filter UFS algorithms depending on spectral feature selection. In LS, the significance of a feature is determined by its variance and its ability to preserve location. This approach gives high significance to the features that mostly keep the predefined graph structure or manifold structure shown through the Laplacian matrix. Later, in 2009, Padungweang et al. [54] proposed a spectral-similarity-based feature selection method in which they used global topology rather than the local topology to calculate the LS and come up with a new version of the LS named Laplacian++. In 2017, Solorio et al. [65] proposed a feature selection method based on the LS for mixed data. In this method, they evaluated features by assessing the changes in the spectrum distribution (spectral gaps) of the first nontrivial eigenvalues of the normalized Laplacian matrix when each feature is omitted from the entire collection of features individually. After that, they arranged features in downward order according to their individual spectral gaps.

Tang et al. [70] introduced a novel UFS technique in 2023 based on the fusion of multiple graphs and feature weight learning. In this approach, the authors hypothesized that the majority of UFS techniques are hampered by the poor quality of similarity graphs. Moreover, the procedure for reconstructing features is not simple. To resolve this deficiency, a weight matrix was used in the process of feature reconstruction. The weighted features are projected into the label space to generate a high-quality similarity graph. Furthermore, they fuse the multiple predefined similarity graphs that are used to regularize the original data's local data structure. In the end, they used the block coordinate descent algorithm, an optimization technique, to produce the optimal solution. They evaluated their method on six commonly used datasets, including COIL20 [40], YaleB [39], Orlrawslop [62], ORL [62], Lung [28], and Tox-17 [40], and discovered that it outperformed the MGFS [17], SOGFS [53], and other various UFS approaches.

In 2023, Zhu et al. [77] came up with a method for UFS that takes into account the fact that data points belonging to the same class are typically near one another. This concept is known as compactness. In this method, the selection of essential features was based on this compactness score. In addition, to reduce the intricacy of the proposed method even further, they proposed a novel method that calculates the addition of a k-nearest neighbor for each sample. Several public datasets, including Mnist, Colon, Lung, Lymphoma, Brain, and Allaml from the UCI machine learning repository [8], were utilized to evaluate their methodology. The performance was superior to LS [27] and unsupervised discriminate feature selection (UDFS) [73].

3.1.2 Multivariate filter method

This method takes into account the correlation between features. Consequently, this method is capable of removing redundant features. There are primarily two categories of multivariate filter feature selection methods: the statistical/information method and the bioinspired method. As the name suggests, the first type comprises UFS methods that carry out feature selection employing information and/

or statistical theory measures like linear correlation, mutual information, entropy, variance-covariance, and others. On the other hand, the second type consists of UFS methods that employ stochastic search strategies based on the swarm intelligence concept to find a suitable feature subset that meets some quality criterion. Both the multivariate filter feature selection methods are discussed subsequently.

Statistical-based method

Among these kinds of works, feature selection using feature similarity (FSFS) [49] is especially well-known and often cited. In this work, the authors came up with a statistical measure of dependency or similarity called the maximal information compression index (MICI), which depends on the variance-covariance between features. This was done to reduce feature redundancy. The aim behind this technique is to cluster the original collection of features so that features that belong to the same group are very similar and features that belong to other groups are quite dissimilar. Later on, in 2011, Yang et al. [73] proposed a UFS method named unsupervised discriminative feature selection (UDFS) that employs the $l_{2,1}$ norm and discriminative analysis to select the most discriminating subset of features. In 2012, Ferreira and Figueiredo [25] proposed a filter-based feature selection approach named relevance redundancy feature selection (RRFS), in which they selected features in two steps. In the first step, they sorted features according to some criteria (for the unsupervised case, variance, and in the supervised case, either the Fisher's ratio or the mutual information). After that, in the next step, by considering the order generated in the first step, the features are analyzed using a feature similarity metric to figure out how often they are the same. After that, the first n features are selected with the lowest redundancy. After that, in 2015, Wang et al. [72] proposed a statistical-based feature selection approach with the idea of minimum redundancy and maximum projection (MRMP). In this method, all features are projected onto a feature subspace using a linear transformation with the least amount of reconstruction error.

Lee et al. [38] came up with information-theoretic UFS (IUFS), whose goal is to get the most information about how chosen features interact with each other. To do this, they used a greedy method to solve an optimization problem and looked for local optima. In 2021, Lim et al. [43] devised an innovative dependency-based UFS method. This method calculated the dependence between features and incorporated it into a regression-based model for feature selection. They evaluated their method on multiple types of datasets, such as image and gene expression datasets titled Coil20 [51], Colon [3], Leukemia [26], and ORL64 [62]. They discovered that their approach performed better than the conventional UFS approaches: LS [27], UDFS [73], NDFS [42], and IUFS [38].

In 2022, Huang et al. [31] proposed a novel information-based UFS method that addresses the issue of imbalanced neighbors in each data sample. This method is used to preserve the intrinsic properties of data and performs well when the features are not linearly independent. Using an adaptive graph and dependency score (AGDS), they ensured the graph's k-connectivity and removed the most redundant nodes. Mutual information and entropy were used to calculate the dependency score. In this method, the weights of the k adjacent neighbors of each data sample are assigned adaptively, thereby resolving the issue of imbalanced neighbors. On several benchmark datasets (object dataset, shape dataset, face dataset, warpAb10P,

warp1E10P) and bioinformatics datasets (colon, lung, lymphoma, leukemia), they evaluated the performance of the ADGS approach. They discovered that their strategy outperforms LS [27], UDFS [73], NDFS [42], and SOGFS [53].

Bioinspired technique

Bioinspired methods are very useful in selecting features to achieve a near-optimal solution. In 2014, Tabakhi et al. [68] proposed the first UFS method using ant colony optimization (UFSACO). They choose the feature set with the lowest cosine similarity among features using the cosine similarity measure. In 2015, Tabakhi et al. [69] proposed another feature selection approach using the same idea as the previous one, named microarray gene selection using ant colony optimization (MGSACO), in which, they used feature redundancy and variance to assign fitness to a feature. Later on, Moradi and Rostami [50] proposed an integrated strategy. In this strategy, graph clustering was combined with ant colony optimization (ACO) to accomplish the feature selection task. They transformed the problem space into a graph where features are nodes and the connections between them represent the similarity between features. In the next phase, features are aggregated using a community detection strategy. The novel ACO-based strategy is used to select the final subset of features in the final phase. They evaluated their methodology using several benchmark UCI machine learning repository datasets [8].

The main drawback of the above filter-based feature selection approaches is that the chosen features may not perform well across all learning models. Features selected based on specific criteria might not be universally effective across different algorithms. This limits their applicability and effectiveness, highlighting the need to consider the characteristics of individual models when selecting features. To overcome this drawback, researchers [32, 57, 67] have explored wrapper-based feature selection methods, which are discussed subsequently.

3.2 Unsupervised wrapper method

The second evaluation method is the wrapper method [60], which binds the feature selection process around the learning algorithm and makes use of performance accuracy or the learning error rate as a criterion for evaluating feature quality. Unlike the filter method, the wrapper method selects the most useful feature subset by minimizing the error of a specific learning approach. The wrapper method typically yields better results compared to the filter method, since it directly incorporates bias from the learning algorithm and considers feature dependency, it tends to provide less generalized features. This is because the wrapper method uses a particular learning algorithm to evaluate feature fitness, making it uncertain that the selected features will be optimal for other learning algorithms. The working of the wrapper method is illustrated in Fig. 8.

The feature selection process in wrapper methods involves a two-step procedure that continues until the stopping criteria are met. Initially, an iterative search strategy is used to obtain the initial feature subset, followed by an evaluation of its effectiveness. Consequently, the wrapper method is effective in achieving more precise

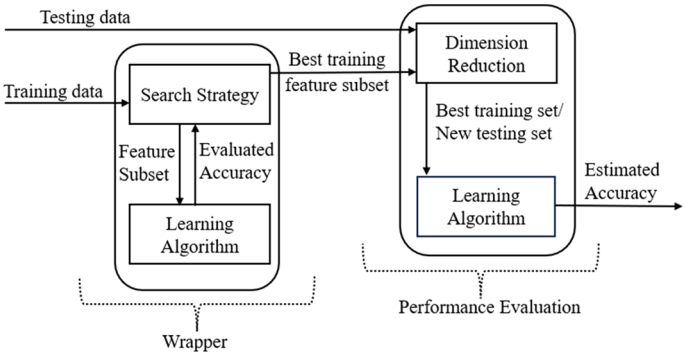

Fig. 8 Wrapper approach for feature selection

results based on the learning algorithm employed, albeit with higher computational costs.

The unsupervised wrapper method is categorized into three types: the wrapper approach for model-based clustering, the wrapper approach for evolutionary clustering, and the wrapper approach for K-means clustering, which is discussed as follows:

3.2.1 Wrapper approach for model-based clustering

In general, model-based clustering takes all attributes into account throughout the modeling process. As discussed above, many features are unimportant and redundant, and considering these features may degrade the model's performance. In addition, the model will be affected by the problem of dimensionality, also known as the "curse of dimensionality." To resolve this issue, various approaches have been proposed by multiple researchers. In 2006, Reftery and Dean proposed the wrapper feature selection for model-based clustering. In this method, they used linear regression to assume that irrelevant features depended on important features. The problem with this method is that regression needs more parameters, but this method does not significantly improve clustering efficiency. Later, Maugis et al. [48] presented an expanded version of Raftery and Dean [58] using greedy feature selection in the regression. In this approach, due to regression, they let irrelevant features be independent of essential features. This strategy is intended to improve clustering performance. However, the approach's general timeline gets considerably more intricate.

3.2.2 Wrapper approach for evolutionary clustering

The performance of evolutionary computation (EC)-based methods for clustering is better than that of well-known techniques like K-means and fuzzy c-means [36]. Data with irrelevant or duplicate features can also hurt the performance of EC-based clustering algorithms. To overcome this problem, a lot of approaches have been proposed by various researchers. In the following study, we will learn more about

choosing features for evolutionary clustering by talking about single-objective and multi-objective strategies.

Single-objective optimization

Swetha and Devi [67] proposed a two-step particle swarm optimization (PSO) feature selection approach for clustering. Initially, they used a two-step PSO to select features and then used clustering on those features. Javani et al. [32] introduced a new PSO-based simultaneous clustering, a method for determining features that use probabilistic K-means clustering, and a new kernelized validity index to counteract the effect of the initial condition of the evolutionary process. However, the major flaw of this approach is that it has not been tested on high-dimensional datasets. Later on, a binary PSO (BPSO)-based feature selection was proposed by Prakash et al. [57] in which each feature subset is encoded by 0 and 1. They used BPSO for the feature selection purpose, and later on, K-means clustering was performed to assess the effectiveness of possible feature subsets in terms of the silhouette index.

Multi-objective optimization

Prakash and Singh [33] proposed a new approach to feature selection and clustering based on genetically inspired multi-objective binary gravitational search in 2019. Feature subset sizes and silhouette indices were used as search objectives to look for potential solution spaces in this approach. For the non-dominated set, they used an external archive as well. The segregated dataset is then subjected to K-means clustering, and the F-score is calculated according to the subset of features that were chosen. The final selection was made based on the best F-score. The results indicate that this approach outperforms the elitist non-dominated sorting GA approach introduced by Deb et al. [14].

3.2.3 Wrapper approach for K-means clustering

The problem associated with K-means clustering is that it treats all features equally important. In other words, each feature adds the same amount to the clustering process, no matter how unimportant, irrelevant, or redundant it is. If there are a lot of irrelevant or duplicate features in the dataset that is used for clustering, the quality of the clustering may go down. It's also possible that adding irrelevant and repetitive features will hurt K-means in the wrong way. This problem has been solved by giving each feature a certain weight, which is called feature weighting [15]. Feature weighting works on the assumption that every feature within the chosen subset can't have the same amount of importance. Rather, it gives each feature a weight, which is generally between 0 and 1. Therefore, feature weighting can be viewed as a generalized form of feature selection.

In 1988, DeSarbo and Cron [15] proposed the first feature weighting technique employed in K-means, named synthesized clustering (SYNCLUS). SYNCLUS looks at the weights of feature groups as well as the weights that show the importance of each feature. In this approach, like K-means, the user must tell the program how many clusters there are and what data to use initially. Even though it is the first task in this area, it may be hard to compute, and its effectiveness is greatly based on how

the parameters are set. Nevertheless, researchers have attempted to improve its efficiency using methods like the Polak–Ribiere optimization strategy [56] and a general linear transformation of features. Some recent work has also been done in this field. In 2005, Hruschka et al. [29] presented silhouette-sequential forward selection (SSFS), a wrapper-based feature selection method. The authors used the k-means clustering algorithm to split the data and pick the feature subset with the highest quality based on the simplified silhouette criterion. Later on, Parvin et al. [55] proposed a weight-local clustering algorithm (FWLAC). They began their investigation by selecting k evenly spaced points as their starting centroids. They then looked at the subspace next to the initial centroids to enhance the original centroids, feature weights, and cluster weights. Nevertheless, their performance is highly dependent on the control parameters.

Alharan et al. [2] proposed a K-means clustering-based feature selection to enhance the classification precision of three image datasets. In this method, each feature was initially evaluated using the five parameters: information gain, gain ratio, oneR, symmetric, and reliefF. In the second phase, a matrix of features and parameters was created, and a K-means clustering algorithm was used to cluster the features into two groups, one containing more relevant features and the other containing noisy features. Comparing their method to those of Kaya et al. [35], Zheng et al. [75], and Al-Sahaf et al. [1], they discovered that their method performed better on the datasets Kylberg [37], Brodatz-1, and Brodatz-2 [1].

The main disadvantage of the above-discussed wrapper feature selection approaches is that they produce results that are closely related to the specific learning model used. While designed to improve performance within a specific model, this approach may limit generalizability across different algorithms or datasets. To overcome this, a hybrid approach has been proposed. This method combines the strengths of filter and wrapper techniques to achieve more robust and effective feature selection. By integrating both strategies, it aims to enhance performance and adaptability across various learning scenarios. The details of the hybrid method are followed next.

3.3 Unsupervised hybrid approaches

The fourth evaluation method is the hybrid method. A hybrid approach can be created by fusing two distinct strategies (such as a filter and a wrapper) or two techniques that share a common criterion or two feature selection methods. The objective of the hybrid method is to take advantage of the best features of both methods. It employs many evaluation criteria at various stages of the search to enhance the accuracy and speed of predictions by making better use of faster computers. There are two different hybridization methods now in use. One approach uses the filter method, which first narrows down the feature set before passing it through the wrapper method to find the optimal feature subset, as shown in Fig. 9. On the other hand, the second strategy couples the filter and wrapper measures to allocate the relevance score to a specific feature.

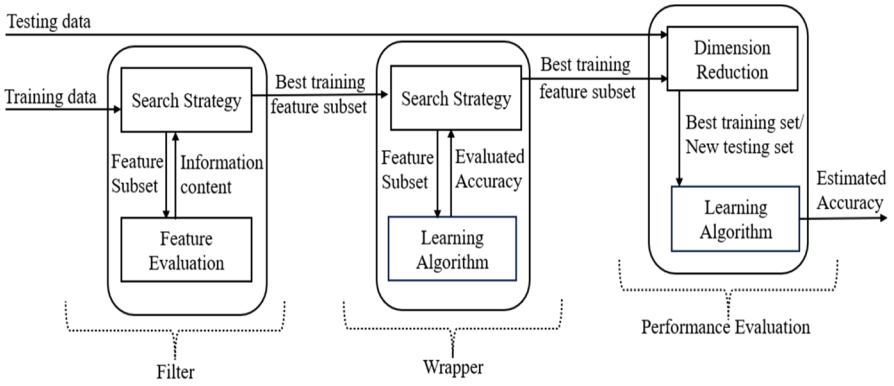

Fig. 9 Hybrid method for feature selection

Dash and Liu [13] came up with a hybrid feature selection approach that derives entropy from data similarity and evaluates features in the filter stage using an entropy-based measure. The wrapper stage employs scatter separability criteria and K-means clustering to select the relevant feature subset. The disadvantage of this approach is that it is computationally intensive. Li et al. [41] later proposed a hybrid feature selection algorithm based on Dash and Liu's [13] concept. To improve performance, they used a fuzzy feature evaluation index (FFEI) in conjunction with an exponential entropy index to evaluate the feature in the filter stage. They used a scatter separability criterion and a fuzzy c-means algorithm for the wrapper stage. Additionally, this approach has a high computational cost.

Hruschka et al. [30] developed an approach that incorporates a Bayesian filter and K-means clustering. They used a Bayesian network with the Markov blanket property as a filtering technique. A limitation of this approach is that only datasets with fewer than 30 features have been tested. Later, Solorio et al. [64] introduced a two-stage hybrid method based on the LS [27] and the Calinski–Harabasz (CH) index [11]. Initially, the LS is used to rank the data features. In the next step, K-means is used to create cluster sets from the selected feature subsets using the forward and backward strategies. The cluster sets are then analyzed with an updated variant of the CH index called the weighted normalized CH index (WNCH). The best feature subset is the one with the greatest WNCH value. The biggest problem with this method is that it uses a sequential search strategy to choose the features. Because of this, it doesn't come up with the global optimal solution.

Manbari et al. [47] introduced a hybrid UFS method for high-dimensional data in 2019. They used a combination of modified binary ant systems (BAS) and clustering to surmount the limitations of search space and process high-dimensional data efficiently. They used a genetic algorithm-inspired damped mutation technique and simulated annealing to avoid becoming confined to local optima. They proposed a novel redundancy reduction strategy to measure the correlation between features. They evaluated their method using the benchmark datasets Wine, Hepatitis, Ionosphere, Spambase, Arrhythmia, Madelon, Colon, and Leukemia from the UCI machine learning repository [8]. They determined that their method performed better in

terms of time and accuracy than the LS [27], UFSACO [68], and MGSACO [69]. Later on, Annavarapu et al. [4] proposed a clustering-based hybrid feature selection method for choosing pertinent and redundant features from high-dimensional microarray data. In the filter stage, they used a combination of K-means and signal-to-noise ratio (SNR). Eventually, the optimal gene subset was determined using a hybrid system comprised of an ant colony optimization method and automated cellular learning. In 2023, Dwivedi et al. [20] [19] proposed a hybrid feature selection approach for data clustering based on ant colony optimization (NCHFS-ACO). In this approach, they used K-means clustering to measure the effectiveness of the NCHFS-ACO, and it is also used to assign the fitness of features in terms of silhouette index along with the LS in the feature selection process. In this method, the feature subset is created randomly so that the system doesn't get stuck in a local optimal solution and the global optimal solution can be attained. They tested their approach on ten benchmarks and three real-life plant genome datasets and found that their method performed better than the method proposed by Solorio et al. [64].

The major drawback of the hybrid method is its extensive computational requirements. To address this, the embedded method was introduced, integrating feature selection directly into the model training process. This approach significantly reduces computation time while maintaining the effectiveness of selecting relevant features.

3.4 Unsupervised embedded approaches

The fourth evaluation method is the embedded method. Unlike the wrapper method, it is a way to choose features built into the learning algorithm and utilizes the properties of the algorithm, which help in deciding how to evaluate features. Even though the performance is the same, the embedded method is more efficacious and easier to compute than the wrapper method. This is because the embedded method eliminates the cost of running the learning algorithm and looking at each feature subset over and over again. Also, this method is less likely to overfit than the wrapper method. Figure 10 illustrates the workings of the embedded method.

The unsupervised embedded approaches comprise UFS methods that use either sparse learning along with spectral analysis or spectral analysis alone. For embedded approaches, choosing a subset of features is part of the learning process. Because of

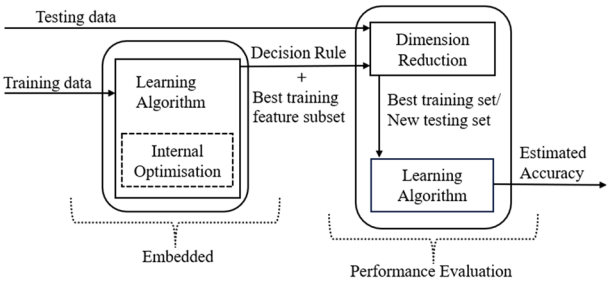

Fig. 10 Embedded method for feature selection

this, embedded methods use less computing power than wrapper approaches. Sparse learning algorithms are very important for embedded methods because they work well and are easily understood. Sparse learning algorithms attempt to find a balance between the degree of quality and the sparsity of the outcome. In the context of sparse learning, we do not only care about the cluster quality or some other performance metric. Additionally, we wish to convey the clustering strategy to a user who is not an expert. In academics, the idea of using sparse learning in clustering has gained significant interest over the past few years. Most of the time, these methods use a clustering technique to find relevant cluster labels, which are then used to convert the UFS into a supervised feature selection, as shown in Fig. 11. Some embedded approaches to feature selection are presented subsequently.

In 2010, Cai et al. [10] proposed a feature selection method termed multi-cluster feature selection (MCFS) for clustering tasks. This embedded, sparse learning approach involves three main steps. Firstly, spectral analysis is utilized to compute the correlation between features. Secondly, an L1-regularized least squares regression model is employed to gauge the goodness of fit for the features. Finally, the features with the highest coefficient values obtained from the regression model are selected. Recent work has been done on sparse learning models that use non-convex sparse regularizer functions and locally linear embedding (LLE). Li et al. [42] introduced a nonnegative discriminative feature selection (NDFS) method for selecting discriminative features from data. In this strategy, they used spectral clustering and feature selection to choose the most discriminative features. Furthermore, they used a nonnegative constraint to learn a more accurate cluster label. They tested their method on several benchmark datasets, including UMIST, AT &T [62], and JAFFE [46]. Later on, Nie et al. [53] suggested a UFS method named unsupervised feature selection with structured graph optimization (SOGFS), in which they specified that the majority of the UFS embedded methods rely on the similarity matrix. However, real-world data contains a substantial amount of noise, so the similarity matrix obtained from the original data cannot be relied upon entirely. To address this shortcoming, they performed feature selection and local structure learning concurrently. In addition, the similarity matrix was constrained to include more pertinent information so that the selected features would be more relevant. They evaluated their methodology on a variety of datasets, including image and biomedical data, handwritten data, etc.

Luo et al. [45] came up with a new UFS method that models the data's manifold structure with LLE. The objective is to describe the intrinsic local geometry with an LLE graph in place of the usual pairwise similarity matrix and a structure regularization term. A feature-level reconstruction rating is set up for each feature using the LLE graph. This score is used to choose the final subset of features. Shi et al. [63], on the other hand, proposed a non-convex sparse learning model. The plan is to choose

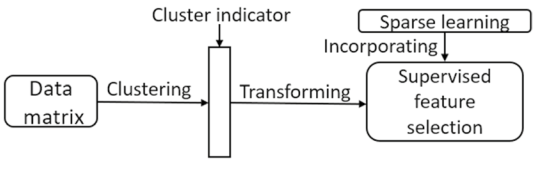

Fig. 11 Working of sparse learning feature selection approach

features using a nonnegative orthogonal constraint sparse regularized model and a novel norm called the $l_{2,1-2}$ norm, which is the difference between the $l_{2,1}$ norm and the frobenius $l_{2,2}$ norm. An iterative algorithm using the alternating direction method of multipliers (ADMM) [9] was also proposed as a way to solve the model quickly. In 2020, Chen et al. [12] proposed an integrated clustering approach that incorporated the feature selection procedure. In this method, the feature selection procedure was conducted by minimizing the l_2 and one output weight norm. Using eigenvalue decomposition, they computed the clustering results. Experiments were conducted using public datasets titled Glass, Ionosphere, Pima, Vehicle, Umist, and WarpIeop from the UCI machine learning repository [8] as well as datasets from WekdTexes at the University of Texas. They determined the effectiveness of the feature selection method by employing the chosen features in clustering. Their algorithm was superior to K-means and US-ELM. Wang et al. [71] proposed a soft-label UFS method in 2021. Due to disturbances, outliers, and redundancies, pseudo-labels were utilized in this method. In addition, they considered the fuzzy nature of the data and employed soft tags. They performed the feature selection task in two phases. In the initial phase, they transformed the data into a lower-dimensional subspace, where the affinity matrix with sparse constraints is learned based on local distances. Later, in the second phase, the feature selection procedure was guided by an affinity matrix. They evaluated their methodology on the COIL20 [51], handwritten binary alphabet (BA) [7], human face dataset ORL [16], voice dataset isolet [24], and biology datasets Lung and ecoli [52]. Their strategy performed superiorly to the UDFS [73], NDFS [42], and SOGFS [53].

In 2020, Zhu et al. [78] came up with a new UFS by building a Laplacian matrix dynamically from a hypergraph. In this method, they attempted to maintain the local structure of samples by employing a Laplacian matrix based on a hypergraph and the global structure by employing orthogonal constraints on the training samples' covariance matrix. In addition, they used an $l_{2,1}$ norm regularizer to determine the most important features. They evaluated their methodology for the clustering and image segmentation tasks. In comparison to LS [27] and SOGFS [53], their method performed well on benchmark datasets taken from the UCI machine learning library [8] and Berkeley segmentation datasets [5]. In 2023, Zhu et al. [76] suggested a UFS approach in which they combined the creation of a similarity matrix and the selection of features into a single framework. This way, the two processes could run at the same time instead of one after the other. In this approach, the similarity matrix adoptively maintains the manifold structure of the data. In addition, the approach uses $l_{2,0}$ norm sparsity constraints to achieve group feature selection. The benefit of this method is that the similarity matrix doesn't stay the same during the process of selecting features. This leads to an optimal solution. They tested their method on various public datasets called JAFFE [46], ORL [62], and COIL20 [40]. They found that their method works better than the LS [27], UDFS [73], and SOGFS [53].

4 Benefits and drawbacks

In the preceding section, Sect. 3, we categorized and reviewed the UFS methods based on their methodology and type. This section will explore some general considerations, pros, and cons of the UFS approaches presented in Sect. 3.

Filter method

The filter methods provide substantial computing benefits because they are not reliant on the clustering process. They possess excellent computing speed and may be effortlessly expanded to manage extensive datasets, rendering them appropriate for high-dimensional data. Furthermore, filter approaches have a tendency to prevent overfitting since they do not directly engage with the clustering process. This segregation leads to enhanced generalization, guaranteeing that the chosen characteristics exhibit excellent performance across diverse learning tasks. In addition, filter techniques generally have better time complexity than wrapper methods, which improves their efficiency. Nevertheless, a significant disadvantage is that filter approaches fail to take into account the unique criteria of the clustering algorithm, potentially resulting in inadequate feature selection for the assigned clustering task.

- **Univariate filter methods:-** Univariate filter methods, including both information-based [6] and spectral-similarity-based approaches [65], are impartial with respect to the dimensionality of the data and have the ability to detect both linear and nonlinear correlations. These methods have a strong theoretical basis for selecting features and are computationally efficient. Nevertheless, these approaches fail to take into account feature interactions, which may result in the omission of crucial feature dependencies, hence potentially diminishing the efficacy of the chosen feature subset.
- **Multivariate filter approaches:-** Multivariate filter approaches [38, 69] overcome the constraints of univariate methods by taking into account the interdependencies among features, leading to the identification of more significant subsets of features. These approaches have superior temporal complexity compared to wrapper methods; however, they are slower than univariate filters. Despite providing a more extensive selection process, multivariate filter approaches encounter difficulties in terms of scalability and processing complexity, particularly when dealing with extremely large datasets.

Wrapper methods

Wrapper approaches combine the process of feature selection with the clustering algorithm, resulting in improved clustering accuracy through the selection of characteristics that are specifically suited to the algorithm's requirements. They consider feature dependencies, which might enhance the quality of the chosen subset of features. However, wrapper techniques do have a number of drawbacks. These tasks require a lot of computer power because they involve repeatedly training and assessing the clustering algorithm using various subsets of features. The significant computational expense of these methods hinders their ability to efficiently handle massive datasets. Moreover, the strong connection between the clustering algorithm and

the risk of overfitting is heightened, especially when dealing with intricate models or limited datasets. Moreover, due to the fact that wrapper approaches are designed to improve a particular clustering algorithm, the chosen characteristics may not exhibit good generalization when used with different algorithms.

- **Model-based and evolutionary computation (EC)-based approaches:-** Model-based [48] and evolutionary computation (EC)-based approaches [33, 57] are highly effective at capturing feature dependencies and optimizing feature subsets. Model-based wrappers improve interpretability and durability, while EC-based wrappers maximize several objectives at the same time, lowering the chance of local optima. However, these methods necessitate meticulous parameter adjustments and involve significant computational resources. EC-based approaches, specifically, exhibit poorer performance compared to filter methods due to their inherent complexity, rendering them less suitable for handling large datasets.

Hybrid method

Hybrid methods [4, 20] seek to merge the advantages of filter and wrapper approaches, providing a trade-off between efficiency and accuracy. They take advantage of the computational efficiency of filter techniques while also benefiting from the customized feature selection of wrapper methods. Hybrid approaches frequently achieve superior performance compared to either approach individually, while also mitigating the risk of overfitting in comparison to pure wrapper methods. Nevertheless, they continue to encounter obstacles pertaining to computational intricacy and scalability. Hybrid methods can be influenced by the settings utilized in both the filter and wrapper components. Although they are more efficient than wrappers, they may still encounter difficulties when dealing with extremely high-dimensional data.

Embedded Method

Embedded approaches [45, 63] incorporate feature selection directly into the clustering algorithm, guaranteeing that the chosen features are extremely pertinent to the clustering job. Compared to wrapper approaches, this integration decreases the likelihood of overfitting and is typically more computationally efficient. Nevertheless, the features used by embedded methods are fine-tuned for the particular clustering algorithm employed, hence potentially restricting their suitability for alternative algorithms. In addition, although embedded techniques are more efficient than wrapper methods, they can still encounter scaling problems when dealing with extremely large datasets.

Based on the above discussion, it can be concluded that every UFS technique possesses distinct advantages and constraints, which are shaped by characteristics such as computational efficiency, scalability, generalizability, and accuracy. When selecting a method, it is important to consider the unique needs of the clustering task, the characteristics of the dataset, and the computational resources that are available. This analysis offers a thorough comprehension of the theoretical consequences of each UFS technique, assisting researchers in choosing the most suitable method for their applications. The benefits and drawbacks of the UFS approaches according to the methodology and their type are presented in Tables 1 and 2, respectively. In the next section, we performed a comparison of four state-of-the-art UFS approaches through experimental analysis.

Table 1 Pros and cons of UFS approaches based on the methodology used

Technique used	Benefits	Drawbacks
Filter methods	1. Computationally fast	1. Does not interact with clustering methods
	2. Can be made scalable	
	3. Avoids overfitting	
	4. Better time complexity than wrapper	
	5. Better generalization	
Wrapper methods	1. Interact with clustering technique	1. Less generalized features
	2. Dependency among features is considered	2. More computational complexity
	3. Better accuracy than filter method	3. Increased chance of overfitting
		4. Difficult to be scalable
Hybrid methods	1. Takes the advantage of both filter and wrapper methods	1. Limited generalizability
	2. Better computational complexity than wrapper	2. More sensitive to the parameters
	3. Lesser risk of overfitting than wrapper	3. Difficult to be scalable
Embedded methods	1. Interact with clustering technique	1. Clustering specific features
	2. Lesser risk of overfitting than wrapper	
	3. Less computationally expensive than wrapper method	

5 Experimental analysis of contemporary unsupervised feature selection methods

We performed an experimental analysis to compare the performance of various techniques and categories of unsupervised feature selection (UFS) methods. For this, we selected four highly relevant and contemporary UFS methods. The aim was to conduct a systematic comparison of these methods, with a specific focus on evaluating the quality of the selected features across different datasets. Specifically, we compared the filter method UFS-SVD proposed by Banerjee et al. [6], the wrapper method SSFS proposed by Hruschka et al. [29], and two hybrid methods, LSWNCH-SR and NCHFS-ACO, proposed by Solorio et al. [64] and Dwivedi et al. [20], respectively. The datasets used in the experimentation and the evaluation measures used to evaluate the performance of these methods are discussed subsequently.

5.1 Dataset details

The above-discussed techniques are tested on ten benchmark datasets obtained from the UCI machine learning library [8]. These datasets go through preprocessing, which includes deleting missing values. The data are then normalized using a standard scaling technique. This scaling assures that each feature has a mean of zero and a variance

Table 2 Pros and cons of UFS approaches based on the taxonomy

Type (Subtype)	Benefits	Drawbacks
Univariate filter (Information based)	1. Unbiased with respect to the data dimensionality 2. Capable of simulating linear and nonlinear relationships	1. Does not consider the correlation among features
Univariate filter (Spectral-similarity based)	1. Facilitate an effective environment for UFS 2. Strong theoretical base	1. Does not consider the correlation among features
Multivariate filter (Spectral/information based)	1. Capable of simulating feature dependencies 2. Lesser time complexity than wrapper method	1. Slower than univariate filter 2. Lesser scalability than univariate filter
Multivariate filter (Bioinspired)	1. Capable of simulating feature dependencies 2. Less chance of getting local optimal solution	1. Slower than univariate filter 2. Extensive computations
Wrapper (Model based)	1. Capable of simulating feature dependencies	1. Needs more number of parameters 2. Extensive computations
Wrapper (EC based) (Single objective)	1. Less chance of getting local optimal solution 2. Capable of simulating feature dependencies	1. Only single objective is optimized. 2. Extensive computations
Wrapper (EC based) (Multi-objective)	1. Multiple objective are optimized simultaneously 2. Capable of simulating feature dependencies	1. Slower in comparison to filter 2. Extensive computations than the single-objective wrapper EC-based methods
Wrapper (K-means)	1. Enhanced interpretability and robustness 2. Noise reduction, enhanced adaptability, and scalability	1. Needs the specification of hyper-parameters in prior 2. Performance is highly dependent on hyper-parameters
Hybrid	1. Takes advantage from both, filter and wrapper methods 2. Generates better results than filter and wrapper methods	1. Extensive computations 2. Lesser scalability than filter
Embedded	1. Strong theoretical base 2. Feature selection is associated with learning algorithm	1. Lesser scalable than the univariate filter methods

of one. The scaling is required because the outputs of feature selection and clustering algorithms are influenced by the fact that these datasets (excluding iris) contain a wide range of values with varying scales. Class labels are ignored and removed from all datasets during the feature selection and clustering process. After preprocessing, the details of the datasets are presented in Table 3.

5.2 Evaluation measure

The performance of any feature selection approach is typically evaluated using a clustering or classification method to determine the extent to which feature selection improves clustering or classification performance. In the context of unsupervised feature selection methods, we used K-means clustering to assess the performance of these methods. For comparison, we applied K-means clustering to the features selected by different approaches and measured the results using the silhouette index.

Silhouette Index

The silhouette index (SI) [61] is a commonly utilized evaluation metric in clustering techniques. The basis of this concept is the similarity of a data point within a cluster, referred to as cohesion, and its proximity to the nearest cluster, known as separation. The SI, or silhouette index, is a numerical measure that goes from – 1 to 1. A high SI value implies that the data points are well-grouped. The silhouette index is determined by computing the mean of the silhouette coefficients for all data points.

If q_r is a r^{th} data point, the silhouette coefficient SC_{q_r} is determined using Eq. (1).

$$SC_{q_r} = (n_{q_r} - m_{q_r})/(max(n_{q_r}, m_{q_r})). \tag{1}$$

Where m_{q_r} refers to the average distance between a certain data point q_r and all other data points inside the same cluster. n_{q_r} on the other hand, represents the average distance between q_r and all other data points in the closest nearby cluster.

Table 3 Dataset details

Dataset	No. of samples	No. of attributes	No. of classes
Iris	150	4	3
Sonar	208	60	2
Vehicle silhouettes	813	18	3
Ionosphere	351	33	2
Pima	768	8	2
Wine	178	13	3
Wdbc	569	30	2
Parkinsons	195	22	2
Pendigits	7494	16	10
Waveform (noise)	5000	40	3

5.3 Experimental comparison and discussion

The experimental findings of the UFS-SVD approach, SSFS, NCHFS-ACO, and Solorio et al. approach on the datasets listed in Table 3 are presented in Table 4.

From the table, it can be observed that the SSFS (wrapper) and NCHFS-ACO (hybrid) approaches performed better than the UFS-SVD (filter) method. This superior performance is likely because wrapper and hybrid approaches use a learning algorithm to evaluate the fitness of the features, resulting in better performance when the same algorithm is used to assess feature selection. Conversely, the UFS-SVD approach evaluates features in a generalized manner, which may lead to improved performance when different clustering algorithms are used. Additionally, when comparing the two hybrid approaches, NCHFS-ACO and LSWNCH-SR, the NCHFS-ACO approach performed better. This can be attributed to NCHFS-ACO's use of both filter and wrapper measures to assign fitness scores, whereas the LSWNCH-SR approach first narrows down the feature set using the filter method before applying the wrapper method. The integrated evaluation in NCHFS-ACO, which simultaneously considers filter and wrapper criteria, leads to a more effective selection of features. In the next section, we described some practical applications of the various UFS approaches.

6 Practical applications of unsupervised feature selection methods

Unsupervised feature selection (UFS) approaches are useful in several sectors where there is a limited availability or high cost associated with obtaining labeled data. In the fields of text mining and natural language processing [21], these techniques play a vital role in reducing the complexity of data and improving the efficiency of tasks such as document clustering and topic modeling. They achieve this by recognizing the most significant words or phrases, even without the requirement for

Table 4 SI values of UFS-SVD approach, SSFS, NCHFS-ACO, and LSWNCH-SR approaches

Dataset	Filter	Wrapper	Hybrid	
	UFS-SVD	SSFS	NCHFS-ACO	LSWNCH-SR
Iris	0.64	**0.73**	**0.73**	0.67
Sonar	0.34	0.64	**0.75**	0.63
Vehicle silhouettes	0.60	0.56	**0.66**	0.56
Ionosphere	0.46	**1.00**	0.75	0.51
Pima	0.60	**0.81**	0.80	0.67
Wine	0.52	**0.68**	0.63	0.57
Wdbc	0.61	0.69	**0.76**	0.60
Parkinsons	0.57	0.70	**0.85**	0.64
Pendigits	0.35	0.68	**0.77**	0.63
Waveform (noise)	0.34	0.33	**0.54**	0.29

labeled training data. UFS in bioinformatics [22] aids in the examination of complex genomic and proteomic data with several variables. It allows researchers to identify noteworthy genes or proteins that distinguish between different circumstances or treatments, without relying on preexisting labels or information.

In the field of image processing and computer vision [18], these techniques aid in the identification of the most pertinent characteristics for tasks such as image segmentation and object recognition. This enables more effective and precise analysis of extensive image datasets. UFS is employed in anomaly detection to uncover patterns or characteristics that can differentiate normal behavior from anomalies in areas such as network security, fraud detection, and industrial monitoring.

Furthermore, UFS plays a crucial role in recommendation systems [44] by discovering important user preferences and item attributes. This leads to an enhanced suggestion process without requiring explicit user feedback. UFS approaches are highly beneficial for deriving significant insights and enhancing the efficiency and efficacy of analytical activities in many domains with limited or unavailable labeled data.

7 Challenges of unsupervised feature selection approaches

By comparing the literature presented, it can be concluded that although significant progress has been made in the UFS field, there are still some main challenges left for researchers:

1. According to the literature study, most UFS methods (filter, wrapper, hybrid, and embedded) need the specification of hyper-parameters such as the count of clusters, count of features, or other feature selection technique-specific parameters. In reality, there is no such information, and it is usually hard to guess the best parameter values for each dataset. Thus, selecting the ideal parameters is a challenge.

2. The UFS approaches must be scalable, i.e., able to handle a huge number of features efficiently. In recent years, datasets consisting of millions of features have grown [34]. Since existing approaches are unable to accommodate a large number of features, therefore, scalable methods are required.

3. The UFS also faces the challenge of selecting relevant features in problems with both numerical and nonnumerical information (mixed data). Many real-world problems have mixed data, such as business, socioeconomics, biological, and healthcare applications. This literature shows most existing approaches are mainly for numerical data. So still, there is a need for UFS approaches that can handle mixed data.

8 Conclusion

Unsupervised Feature Selection (UFS) approaches have garnered significant attention across various research fields due to their ability to select features from unlabeled data. This study examines the most important and up-to-date UFS methods at the forefront of the field. We also developed a taxonomy of UFS approaches,

highlighting the benefits and drawbacks of the broad categories. Additionally, we compared state-of-the-art methods through experimental analysis. This work also addresses current issues such as scalability, hyper-parameter specification, and UFS techniques for mixed data, providing a foundation for future research.

Funding This research is funded by The Council of Scientific and Industrial Research (CSIR), Government of India under grant no. 22(0853)/20/EMR-II.

Availability of data and materials: The corresponding author can provide the datasets used in this study upon reasonable request.

Declarations

Conflict of interest The authors report having no conflict of interest.

References

1. Al-Sahaf H, Al-Sahaf A, Xue B et al (2016) Automatically evolving rotation-invariant texture image descriptors by genetic programming. IEEE Trans Evol Comput 21(1):83–101
2. Alharan AF, Fatlawi HK, Ali NS (2019) A cluster-based feature selection method for image texture classification. Indones J Electr Eng Comput Sci 14(3):1433–1442
3. Alon U, Barkai N, Notterman DA et al (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci 96(12):6745–6750
4. Annavarapu CSR, Dara S et al (2021) Clustering-based hybrid feature selection approach for high dimensional microarray data. Chemom Intell Lab Syst 213(104):305
5. Arbelaez P, Maire M, Fowlkes C et al (2010) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
6. Banerjee M, Pal NR (2014) Feature selection with svd entropy: some modification and extension. Inf Sci 264:118–134
7. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
8. Blake C (1998) Uci repository of machine learning databases. http://www ics uci edu/mlearn/MLRepository html
9. Boyd S, Parikh N, Chu E et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122
10. Cai D, Zhang C, He X (2010) Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 333–342
11. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat Theory Methods 3(1):1–27
12. Chen J, Zeng Y, Li Y et al (2020) Unsupervised feature selection based extreme learning machine for clustering. Neurocomputing 386:198–207
13. Dash M, Liu H (2000) Feature selection for clustering. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp 110–121
14. Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
15. DeSarbo WS, Cron WL (1988) A maximum likelihood methodology for clusterwise linear regression. J Classif 5(2):249–282
16. Douglas-Cowie E, Cowie R, Schröder M (2000) A new emotion database: considerations, sources and scope. In: ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion
17. Du X, Yan Y, Pan P et al (2016) Multiple graph unsupervised feature selection. Signal Process 120:754–760

18. Dwivedi R, Kumar R, Jangam E et al (2019) An ant colony optimization based feature selection for data classification. Int J Recent Technol Eng 7:35–40
19. Dwivedi R, Tiwari A, Bharill N et al (2023) A hybrid feature selection approach for data clustering based on ant colony optimization. In: Part III (ed) Neural Information Processing: 29th International Conference, ICONIP 2022, Virtual Event, November 22–26, 2022, Proceedings. Springer, pp 659–670
20. Dwivedi R, Tiwari A, Bharill N et al (2023) A novel clustering-based hybrid feature selection approach using ant colony optimization. Arab J Sci Eng 48:10727–10744. https://doi.org/10.1007/s13369-023-07719-7
21. Dwivedi R, Tiwari A, Bharill N, et al (2023c) A novel feature extraction approach for the clustering and classification of genome sequences. In: 2023 IEEE symposium series on computational intelligence (SSCI). IEEE, pp 1018–1023
22. Dwivedi R, Tiwari A, Bharill N et al (2024) A novel apache spark-based 14-dimensional scalable feature extraction approach for the clustering of genomics data. J Supercomput 80(3):3554–3588
23. Dwivedi R, Tiwari A, Bharill N et al (2024) An incremental clustering method based on multiple objectives for dynamic data analysis. Multimedia Tools Appl 83(13):38145–38165
24. Fanty M, Cole R (1990) Spoken letter recognition. In: Advances in neural information processing systems, vol 3
25. Ferreira AJ, Figueiredo MA (2012) An unsupervised approach to feature discretization and selection. Pattern Recogn 45(9):3048–3060
26. Golub TR, Slonim DK, Tamayo P et al (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
27. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. In: Advances in neural information processing systems, vol 18
28. Hong ZQ, Yang JY (1991) Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognit 24(4):317–324
29. Hruschka ER, Covoes TF (2005) Feature selection for cluster analysis: an approach based on the simplified silhouette criterion. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). IEEE, pp 32–38
30. Hruschka ER, Covoes TF, Ebecken NF (2005) Feature selection for clustering problems: a hybrid algorithm that iterates between k-means and a Bayesian filter. In: Fifth International Conference on Hybrid Intelligent Systems (HIS'05). IEEE, pp 6
31. Huang P, Yang X (2022) Unsupervised feature selection via adaptive graph and dependency score. Pattern Recogn 127(108):622
32. Javani M, Faez K, Aghlmandi D (2011) Clustering and feature selection via pso algorithm. In: 2011 international symposium on artificial intelligence and signal processing (AISP). IEEE, pp 71–76
33. Jay Prakash Pks (2019) Gravitational search algorithm and k-means for simultaneous feature selection and data clustering: a multi-objective approach. Soft Comput 23:2083–2100
34. Jha P, Tiwari A, Bharill N et al (2023) Scalable kernelized deep fuzzy clustering algorithms for big data. In: 2023 IEEE symposium series on computational intelligence (SSCI). IEEE, pp 1322–1327
35. Kaya Y, Ertuğrul ÖF, Tekin R (2015) Two novel local binary pattern descriptors for texture analysis. Appl Soft Comput 34:728–735
36. Kumar R, Dwivedi R, Jangam E (2019) Hybrid fuzzy c-means using bat optimization and maxi-min distance classifier. In: Advances in Computing and Data Sciences: Third International Conference, ICACDS 2019, Ghaziabad, India, April 12–13, 2019, Revised Selected Papers, Part II 3. Springer, pp 68–79
37. Kylberg G (2011) Kylberg texture dataset v. 1.0. Centre for Image Analysis, Swedish University of Agricultural Sciences
38. Lee J, Seo W, Kim DW (2018) Efficient information-theoretic unsupervised feature selection. Electron Lett 54(2):76–77
39. Lee KC, Ho J, Kriegman DJ (2005) Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans Pattern Anal Mach Intell 27(5):684–698
40. Li J, Cheng K, Wang S et al (2017) Feature selection: a data perspective. ACM Comput Surv (CSUR) 50(6):1–45
41. Li Y, Lu BL, Wu ZF (2006) A hybrid method of unsupervised feature selection based on ranking. In: 18th International Conference on Pattern Recognition (ICPR'06). IEEE, pp 687–690

42. Li Z, Yang Y, Liu J et al (2012) Unsupervised feature selection using nonnegative spectral analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1026–1032
43. Lim H, Kim DW (2021) Pairwise dependence-based unsupervised feature selection. Pattern Recogn 111(107):663
44. Lin W, Zhao X, Wang Y et al (2022) Adafs: Adaptive feature selection in deep recommender system. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp 3309–3317
45. Luo M, Nie F, Chang X et al (2017) Adaptive unsupervised feature selection with structure regularization. IEEE Trans Neural Netw Learn Syst 29(4):944–956
46. Lyons MJ, Budynek J, Akamatsu S (1999) Automatic classification of single facial images. IEEE Trans Pattern Anal Mach Intell 21(12):1357–1362
47. Manbari Z, AkhlaghianTab F, Salavati C (2019) Hybrid fast unsupervised feature selection for high-dimensional data. Expert Syst Appl 124:97–118
48. Maugis C, Celeux G, Martin-Magniette ML (2009) Variable selection for clustering with gaussian mixture models. Biometrics 65(3):701–709
49. Mitra P, Murthy C, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell 24(3):301–312
50. Moradi P, Rostami M (2015) Integration of graph clustering with ant colony optimization for feature selection. Knowl-Based Syst 84:144–161
51. Nene SA, Nayar SK, Murase H (1996) Columbia object library (coil-20). Department of Computer Science, Columbia University, Technical report
52. Nie F, Wang X, Huang H (2014) Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 977–986
53. Nie F, Zhu W, Li X (2016) Unsupervised feature selection with structured graph optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence
54. Padungweang P, Lursinsap C, Sunat K (2009) Univariate filter technique for unsupervised feature selection using a new Laplacian score based local nearest neighbors. In: 2009 Asia-Pacific Conference on Information Processing. IEEE, pp 196–200
55. Parvin H, Minaei-Bidgoli B (2015) A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm. Pattern Anal Appl 18(1):87–112
56. Polak E, Ribiere G (1969) Note sur la convergence de méthodes de directions conjuguées. Revue française d'informatique et de recherche opérationnelle Série rouge 3(16):35–43
57. Prakash J, Singh PK (2015) Particle swarm optimization with k-means for simultaneous feature selection and data clustering. In: 2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI). IEEE, pp 74–78
58. Raftery AE, Dean N (2006) Variable selection for model-based clustering. J Am Stat Assoc 101(473):168–178
59. Rao VM, Sastry V (2012) Unsupervised feature ranking based on representation entropy. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT). IEEE, pp 421–425
60. Ron K, George HJ (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
61. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
62. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of 1994 IEEE workshop on applications of computer vision. IEEE, pp 138–142
63. Shi Y, Miao J, Wang Z et al (2018) Feature selection with \ell_ {2, 1–2} regularization. IEEE Trans Neural Netw Learn Syst 29(10):4967–4982
64. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF (2016) A new hybrid filter-wrapper feature selection method for clustering based on ranking. Neurocomputing 214:866–880
65. Solorio-Fernández S, Martínez-Trinidad JF, Carrasco-Ochoa JA (2017) A new unsupervised spectral feature selection method for mixed data: a filter approach. Pattern Recogn 72:314–326
66. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF (2020) A review of unsupervised feature selection methods. Artif Intell Rev 53(2):907–948
67. Swetha K, Susheela Devi V (2012) Simultaneous feature selection and clustering using particle swarm optimization. In: International Conference on Neural Information Processing. Springer, pp 509–515

68. Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell 32:112–123. https://doi.org/10.1016/j.engappai.2014.03.007 (https://www.sciencedirect.com/science/article/pii/S0952197614000621)
69. Tabakhi S, Najafi A, Ranjbar R et al (2015) Gene selection for microarray data classification using a novel ant colony optimization. Neurocomputing 168:1024–1036
70. Tang C, Zheng X, Zhang W et al (2023) Unsupervised feature selection via multiple graph fusion and feature weight learning. Sci China Inf Sci 66(5):1–17
71. Wang F, Zhu L, Li J et al (2021) Unsupervised soft-label feature selection. Knowl-Based Syst 219(106):847
72. Wang S, Pedrycz W, Zhu Q et al (2015) Unsupervised feature selection via maximum projection and minimum redundancy. Knowl-Based Syst 75:19–29
73. Yang Y, Shen HT, Ma Z et al (2011) l2,1 -norm regularized discriminative feature selection for unsupervised learning. In: IJCAI International Joint Conference on Artificial Intelligence
74. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5:1205–1224
75. Zheng H, Fang L, Ji M et al (2016) Deep learning for surface material classification using haptic and visual information. IEEE Trans Multimedia 18(12):2407–2416
76. Zhu P, Hou X, Tang K et al (2023) Unsupervised feature selection through combining graph learning and l2, 0-norm constraint. Inf Sci 622:68–82
77. Zhu P, Hou X, Tang K, et al (2023b) Compactness score: a fast filter method for unsupervised feature selection. Ann Oper Res 1–17
78. Zhu X, Zhang S, Zhu Y et al (2020) Unsupervised spectral feature selection with dynamic hypergraph learning. IEEE Trans Knowl Data Eng 34(6):3016–3028

Authors and Affiliations

Rajesh Dwivedi[1] · Aruna Tiwari[1] · Neha Bharill[2] · Milind Ratnaparkhe[3] · Alok Kumar Tiwari[4]

✉ Rajesh Dwivedi
rajeshdwivedi@iiti.ac.in

Aruna Tiwari
artiwari@iiti.ac.in

Neha Bharill
neha.bharill@mahindrauniversity.edu.in

Milind Ratnaparkhe
milind.ratnaparkhe@icar.gov.in

Alok Kumar Tiwari
alokk@iiitm.ac.in

[1] Department of Computer Science, IIT Indore, Simrol, Indore, Madhya Pradesh 453552, India

[2] Department of Computer Science, Mahindra University, Ecole Centrale School of Engineering, Hyderabad, Telangana 500043, India

[3] ICAR-Indian Institute of Soybean Research Indore, Indore, Madhya Pradesh 452001, India

[4] Department of Computer Science, ABV-Indian Institute of Information Technology and Management, Gwalior, Madhya Pradesh 474015, India