Journal of Big Data

RESEARCH Open Access

GB-AFS: graph-based automatic feature selection for multi-class classification via Mean Simplified Silhouette

David Levin[1*] and Gonen Singer[1]

*Correspondence:
david.levin2@live.biu.ac.il

[1] Faculty of Engineering, Bar-Ilan University, Ramat Gan 5290002, Israel

Abstract

This paper introduces a novel graph-based filter method for automatic feature selection (abbreviated as GB-AFS) for multi-class classification tasks. The method determines the minimum combination of features required to sustain prediction performance while maintaining complementary discriminating abilities between different classes. It does not require any user-defined parameters such as the number of features to select. The minimum number of features is selected using our newly developed Mean Simplified Silhouette (abbreviated as MSS) index, designed to evaluate the clustering results for the feature selection task. To illustrate the effectiveness and generality of the method, we applied the GB-AFS method using various combinations of statistical measures and dimensionality reduction techniques. The experimental results demonstrate the superior performance of the proposed GB-AFS over other filter-based techniques and automatic feature selection approaches, and demonstrate that the GB-AFS method is independent of the statistical measure or the dimensionality reduction technique chosen by the user. Moreover, the proposed method maintained the accuracy achieved when utilizing all features while using only 7–30% of the original features. This resulted in an average time saving ranging from 15% for the smallest dataset to 70% for the largest. Our code is available at https://github.com/davidlevinwork/gbfs/.

Keywords: Multi-class feature selection, Silhouette, Graph-based feature selection, Nonlinear dimensionality reduction

Introduction

Feature selection is a crucial step in the process of developing effective machine-learning models. Selecting the most relevant features from a dataset helps to reduce model complexity, prevent overfitting, and improve model interpretability and performance [1]. In recent years, with the explosion of big data, feature selection has become an increasingly important technique in machine learning, as it can significantly reduce the time and resources required for model development and at the same time maintain prediction accuracy [2].

The rise of big data presents distinctive challenges and opportunities. In today's digital era, the pace of data generation has accelerated, particularly in fields such as genomics,

where both the complexity and volume of data have significantly increased [3]. Effectively managing such extensive and complex data requires preprocessing methods. Feature selection emerges as a pivotal technique in this context, not only simplifying the model by reducing the number of parameters but also enhancing the performance and accuracy of data classification [4]. It plays an instrumental role in addressing the curse of dimensionality, which is a significant challenge in big data environments, by selecting features that are most relevant and non-redundant, thereby preserving the essence of the original data while discarding the redundant [5].

The main goal of feature selection is to find the optimal k-sized subset of features that accurately represents the input data [6]. This technique aims to reduce the impact of irrelevant variables and noise, maintaining prediction accuracy [7]. There are three main types of feature selection methods: wrapper, embedded, and filter. Each type employs distinct strategies for selecting features and comes with its unique benefits and drawbacks [8].

Wrapper methods select feature subsets based on their performance in predictive models, often leading to highly accurate and model-specific feature sets [9]. Their major drawback is computational intensity, as they evaluate numerous feature combinations through repeated model training and validation, making them less feasible for large-scale applications and datasets, or rapid prototyping [6].

Embedded methods integrate feature selection directly into the model training process, typically using regularization techniques to penalize less relevant features [10]. While this approach is efficient and can enhance model performance, it results in feature selections that are closely tied to the specific learning algorithm, potentially limiting their generalizability to other models or data scenarios and may result in overfitting [11].

Filter methods rank features by their statistical characteristics or relevance to the target variable, offering a model-independent, faster, and more efficient approach, particularly for large datasets [12]. However, these methods come with certain limitations. Firstly, the selection process in filtering methods, typically based on the highest score of a feature according to the method's index, may overlook the interrelationships between features [13]. This can result in selecting features with similar rather than complementary abilities. Secondly, most of the filter-based methods rely on user input to determine the size of the feature subset, presuming the user has prior knowledge of the data, which often leads to a trial-and-error approach. Thirdly, as these methods are independent of the learning model, the user needs to evaluate the quality of the selected subset of features, so he needs to perform multiple runs of different classifiers with different calibrations, in order to choose a quality final solution.

This paper proposes a novel filter-based feature selection method for multi-class classification tasks that addresses these shortcomings. Our approach generates a new feature space, defined by how well each feature differentiates between class pairs. Instead of merely picking features with the highest separation scores, common in other filtering-based approaches, our method aims to select a group of features with complementary discrimination capabilities. This is achieved using the K-medoids algorithm in a low dimensional space, which preserves the inherent nonlinear structure of the original feature space, allowing the selection of features from various parts of the

space. Additionally, in contrast to conventional filter methods that require predefined input regarding the subset size of features, our approach stands out by its inherent capability to autonomously discover the minimal combination of features necessary for preserving the overall prediction performance when using the entire feature set, using a newly developed Mean Simplified Silhouette (MSS) index. The MSS index, which is based on the Simplified Silhouette (SS) index [14], evaluates clustering outcomes in the context of feature selection for classification problems. It assesses the effectiveness of the selected subset of features obtained from clustering results, aiming to select features that spans the entire feature space, i.e. with complementary separation capabilities. We demonstrate a strong correlation between the MSS index values and accuracy results across a variety of datasets that differ in size and characteristics, as well as across different classifiers. Leveraging this correlation, we can evaluate the quality of the selected subset of features with the MSS index only, without the need to run classifiers, thus saving significant run-time. The major contributions of this work include:

- A graph-based feature selection method is proposed to identify the minimum set of k features required to preserve the accuracy of predictions when using the entire feature set.
- An agnostic methodology that is independent of specific statistical measures or dimensionality reduction techniques for assessing the features' ability to distinguish between classes.
- A novel Silhouette-based index is developed to evaluate clustering outcomes in the context of feature selection for multi-class classification problems.
- The effectiveness and superior performance of our approach compared to state-of-the-art filtering methods are demonstrated via an experimental analysis.

The remainder of this paper is organized as follows. The next section gives a brief overview of related works. Section Definitions and background introduces some definitions and the required background for a better understanding of the proposed method and experiments. Section Proposed method outlines the proposed method in detail. The experimental results and discussions are in Section Experimental results. Finally, in Section Conclusion and future work, the conclusion and future research directions are discussed.

Related works

Recent advancements in filter-based feature selection have explored the use of graph-based techniques. Graph-based techniques have emerged as a promising field due to their advantages, such as improved interpretability, capturing complex relationships between features [15], and the potential to handle high-dimensional data more effectively [16]. These methods involve constructing a graph that captures pairwise relationships between features in the data, while also considering their relevance and redundancy. These methods rank the features based on specific criteria and select the k features with the highest score. Building on these developments, Briola et al. [17] introduced an innovative unsupervised, graph-based filter feature selection technique leveraging topologically constrained network representations. This approach incorporates a

selection strategy that prioritizes the top k features, refining the process of feature selection. A main drawback of these methods, however, is that selecting features with the highest score may result in selecting features with identical characteristics that do not cover the entire feature space, leading to a loss of information.

Friedman et al. [18] proposed a potential solution to the aforementioned limitation through a filter-based feature selection technique that utilizes diffusion maps [19]. This method encompasses the entire feature space by constructing a new feature space based on the features' separation capabilities. It then selects features with complementary separation capabilities that cover the entire feature space. Similarly, Amin *et al.'s* Multi-label Graph-Based Feature Selection (MGFS) method [20], which also addresses the same problem, employs the PageRank algorithm [21] for efficient feature selection in multi-label data. MGFS constructs a graph with features as nodes linked according to their similarity, which is measured using a correlation distance matrix. The significance of each feature is assessed using PageRank, simplifying the identification of key features in complex datasets. Likewise, Parlak et al. [22] proposed an Extensive Feature Selector (EFS) method utilizes class-based and corpus-based probabilities to select distinctive features for text classification. It incorporates clustering by calculating both corpus-based and class-based probabilities separately, aiming to choose more distinctive features. Collectively, these methods share the goal of selecting features with complementary capabilities to provide a comprehensive understanding of the feature space.

A major drawback of filter-based feature selection techniques is the necessity to determine the number of selected features k as an input parameter. Usually, this value is defined under the assumption that a minimal percentage of features encompasses all necessary information. This limitation presents a considerable challenge since the optimal k value can vary based on the data and the task, resulting in a trial-and-error approach that demands significant time and resources. Hence, there is an increasing demand for a filter-based feature selection algorithm that can automatically determine the minimal combination of features required to sustain prediction performance.

Roffo et al. [23] introduced Infinite Feature Selection (Inf-FS), which addresses this specific challenge. This method represents features as paths in a graph, with nodes denoting features and edges signifying their relevance and non-redundancy. It utilizes matrix power series and Markov chains for ranking, and a clustering algorithm subsequently selects the final feature set based on these rankings. A notable drawback of this method, is the requirement to fine-tune the α parameter, which balances feature relevance and diversity. Different values of the α parameter affect the scoring of the features and the priority given to the selected features, meaning that different α values may lead to completely different final results. Addressing the same limitation, Thiago et al. [24] introduced a filter-based algorithm named Supervised Simplified Silhouette Filter (S^3F) that employs the Simplified Silhouette (SS) [14, 25] index to overcome this specific limitation. This method requires the user to define a search range by determining both a minimum and maximum value for k. The method then proceeds to identify the optimal k within this specified range. This approach exhibits two primary shortcomings. Firstly, if the user selects a minimum and maximum k that fail to encompass the optimal k value, the latter is not identified. Secondly, although the SS index is effective for

assessing clustering quality, it is not designed to evaluate clustering quality for feature selection tasks in classification problems. Although the two methods mentioned above inherently determine the feature subset size within their frameworks, it's important to note that their performance is substantially influenced by user-defined parameters, which have a considerable impact on the outcome.

In summary, the analysis of existing literature reveals a significant need for a filter-based feature selection method, which not only focuses on choosing features with complementary discriminating abilities that cover the entire feature space but also determines the minimal size of the final feature subset. This should be achieved independently of user-defined parameters to facilitate full adaptability and automation of the feature selection process across a broader range of datasets.

Definitions and background

For a better explanation of the proposed method and experiments, in this section, some primary definitions and backgrounds are explained.

Notation

Denote the learned dataset by (X, Y), where X is a $N \times M$ dataset, with N representing the number of samples and M representing the features' dimension. The label is stored in the $N \times 1$ vector Y, which assumes C classes. The dataset X comprises M feature vectors, represented by $F = \{f_1, ..., f_M\}$, where each f_i is of size $1 \times N$.

Jeffries–Matusita distance

In the experiments, we will employ the JM distance within the proposed method, as a statistical measure of the similarity between two probability distributions [26, 27]. The formulation for the JM distance, as adopted from the official documentation in the cited literature, is defined as follows. Given a feature $f_i \in F$, we use the JM distance to construct a $C \times C$ matrix, JM_i, which defines how well the feature f_i differentiates between all pairs of classes. Specifically, the matrix entry $JM_i(c, \tilde{c})$ indicates how well the feature f_i differentiates between the two classes c and $\tilde{c}$, where $1 \leq c, \tilde{c} \leq C$. The matrix entries are computed by:

$$JM_i(c, \tilde{c}) = 2\left(1 - e^{-B_i(c,\tilde{c})}\right) \tag{1}$$

where:

$$B_i(c, \tilde{c}) = \frac{1}{8}\left(\mu_{i,c} - \mu_{i,\tilde{c}}\right)^2 \frac{2}{\sigma_{i,c}^2 + \sigma_{i,\tilde{c}}^2} + \frac{1}{2}\ln\left(\frac{\sigma_{i,c}^2 + \sigma_{i,\tilde{c}}^2}{2\sigma_{i,c}\sigma_{i,\tilde{c}}}\right) \tag{2}$$

is the Bhattacharyya distance. The values $\mu_{i,c}, \mu_{i,\tilde{c}}$ and $\sigma_{i,c}, \sigma_{i,\tilde{c}}$ are the mean and variance values of two given classes c and $\tilde{c}$ from the feature f_i.

t-distributed Stochastic Neighbor Embedding

In the experiments, we will integrate t-SNE [28] into the proposed method as a nonlinear dimensionality reduction algorithm that maps high-dimensional data to a low-dimensional space while preserving local structure. For this paper to be self-contained,

we present the t-SNE method and formulations according to the cited literature. The t-SNE algorithm is an improvement over the original SNE (Stochastic Neighbor Embedding) [29] algorithm, providing more accurate and interpretable visualizations by mitigating the crowding problem and simplifying the optimization process [30].

The t-SNE algorithm works by embedding points from a high-dimensional space $\mathbb{R}^M$ into a lower-dimensional space $\mathbb{R}^R$, while preserving the pairwise similarities between the points ($R \ll M$). Given a dataset of N points $\{u_1, u_2, ..., u_N\} \in \mathbb{R}^M$ in the high-dimensional space, the t-SNE algorithm aims to find a corresponding set of points $\{v_1, v_2, ..., v_N\} \in \mathbb{R}^R$ in the low-dimensional space that best reflects the similarities in the original space.

The algorithm defines pairwise conditional probabilities $p_{j|i}$ as the likelihood that point u_j is u_i's neighbor in the high-dimensional space. These probabilities are defined as:

$$p_{j|i} = \frac{\exp^{\left(-\|u_i - u_j\|^2 / 2\sigma_i^2\right)}}{\Sigma_{k \neq i} \exp^{\left(-\|u_i - u_k\|^2 / 2\sigma_i^2\right)}} \tag{3}$$

where σ_i is the variance of the Gaussian centered at point u_i. The value of $p_{j|i}$ is influenced by the distance between points u_i and u_j, with closer points having higher probabilities. The algorithm defines a symmetric pairwise similarity p_{ij}, which measures the similarity between points u_i and u_j in the high-dimensional space, defined as the average of the conditional probabilities $p_{i|j}$ and $p_{j|i}$:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}. \tag{4}$$

The use of the symmetric pairwise similarity allows for a more balanced representation of similarities between points, mitigating the effects of differences in local densities. In the low-dimensional space, pairwise similarities between points v_i and v_j are defined as q_{ij}:

$$q_{ij} = \frac{\left(1 + \|v_i - v_j\|^2\right)^{-1}}{\Sigma_{k \neq l} \left(1 + \|v_k - v_l\|^2\right)^{-1}}. \tag{5}$$

The t-SNE algorithm seeks to minimize the divergence between the distributions P and Q, which is measured by the Kullback-Leibler (KL) divergence:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{6}$$

Minimizing the KL divergence ensures that the low-dimensional embedding preserves the pairwise similarities between points as accurately as possible.

Silhouette

In the proposed method, we develop a Silhouette-based index for evaluating the quality of clustering in the context of feature selection. The classical Silhouette index [31] is a metric used to evaluate clustering quality by measuring how similar a data point is to

its own cluster compared to other clusters. It has a value between −1 and 1, indicating the level of separation between the clusters and the level of cohesion within each cluster. Specifically, the index calculates, for each point i, the average distance of the point from all other points in the same cluster, $a(i)$, and the average distance of the point from all other points in the closest neighboring cluster, $b(i)$. Thus, the Silhouette value for point i is computed as follows:

$$sil(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{7}$$

where −1 indicates a data point closer to the neighboring cluster, 0 indicates a boundary point, and 1 indicates a data point that is much closer to the other points in the same cluster than to the points of the closest cluster. The Silhouette value of a full clustering is the average value of $sil(i)$ across all data points.

The Silhouette index, being computationally expensive and sensitive to outliers, prompted the development of the Simplified Silhouette (SS) index [14, 25], a faster and more robust alternative. The SS index for a point i is computed as follows:

$$ss(i) = \frac{b(i)' - a(i)'}{\max\left\{a(i)', b(i)'\right\}} \tag{8}$$

where $a(i)^{'}$ is the distance of point i from the centroid of its own cluster and $b(i)^{'}$ is the distance of point i from the centroid of the nearest neighboring cluster (in this work, centroids replaced by medoids). The $ss(i)$ value ranges from −1 to 1. Because at the end of K-means or K-medoids clustering, the distance of a data point to its closest neighboring cluster's centroid or medoid $b(i)^{'}$ is always greater than or equal to the distance to its own cluster's centroid or medoid $a(i)^{'}$, the term $\max\{a(i)^{'}, b(i)^{'}\}$ can be simplified to $b(i)^{'}$ [25]. Therefore, after executing the K-means or K-medoids algorithms, the SS value for a single point can also be simplified as follows:

$$ss(i) = 1 - \frac{a(i)^{'}}{b(i)^{'}}. \tag{9}$$

Similarly to the Silhouette index, the SS index is the average of the SS over all data points.

Kneedle algorithm

In the proposed method, we will employ the Kneedle algorithm as a selection tool to identify the minimal subset of k_{min} features that can effectively classify different classes without experiencing a decline in performance. The Kneedle algorithm [32] is used to identify the points of maximum curvature in a given discrete dataset, commonly referred to as "knees". These knees are generally the set of points on a curve that represent local maxima if the curve is rotated by an angle of θ degrees clockwise about the point (x_{min}, y_{min}) through the line that connects (x_{min}, y_{min}) and (x_{max}, y_{max}) points. The identified points are those that differ most from the straight-line segment connecting the

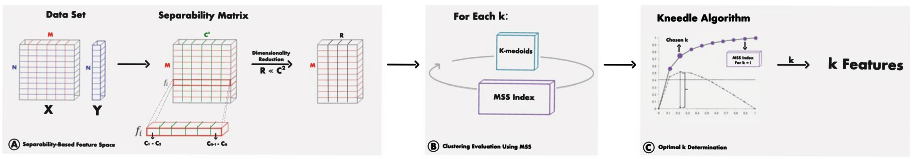

Fig. 1 The GB-AFS architecture: **A** Generate a feature separability matrix and reduce the dimensionality. **B** Perform K-medoids clustering for every $k \in [2, M]$ and calculate MSS. **C** Find the optimal k and return the corresponding k features found during clustering

first and last data points, representing the points of maximum curvature for a discrete set of points.

Algorithm 1 Implementation of the GB-AFS method

Input : Feature vectors $F = \{f_1, ..., f_M\}$
Output: A subset of k_{min} selected features

1 For each feature f_i compute class separability-matrix T_i of size $C \times C$
2 Reshape T_i to a $1 \times (C^2)$ vector, and construct a dataset $\mathcal{Z}$ with the reshaped T_i matrices as its rows
3 Apply nonlinear dimensionality reduction technique to $\mathcal{Z}$, resulting in a new dataset $\mathcal{Q}$ with a lower-dimensional space $\mathbb{R}^R$
4 $\bar{S} \leftarrow \emptyset$ `// Store MSS values`
5 **for** $k \leftarrow 2$ **to** M **do**
6 Apply k-Medoid to $\mathcal{Q}$
7 $\bar{S} \leftarrow$ Calculate MSS value
8 $k_{min} \leftarrow$Apply Kneedle Algorithm to $\bar{S}$
9 Return the k_{min} features that were found in line 6

Proposed method

We now present our graph-based filter method for automatic feature selection (GB-AFS). As explained above, the method determines the optimal subset of features that best represent the data, achieving an effective balance between model performance and computational efficiency. The overall architecture of our method is presented in Fig. 1, while Algorithm 1 outlines the specific steps for implementing the method. Sections Separability-based feature space, Clustering evaluation for feature selection using MSS, and Optimal k determination present the three stages of the proposed GB-AFS method.

Separability-based feature space

Our aim is to preprocess the input data and move them into a reduced feature space that retains the original feature space's ability to distinguish between each pair of classes. For this goal, we will initially create a new feature space $\mathcal{Z}$, defined by the separability capability of each feature with respect to every pair of classes. For each feature $f_i \in F$, we compute a T_i matrix of size $C \times C$ that captures the separation capabilities of each feature with respect to all possible pairs of classes. This computation uses a statistical measure to assess the distance between two probability distributions. Specifically, the matrix entry $T_i(c, \tilde{c})$ indicates how well the feature f_i differentiates between the two classes c

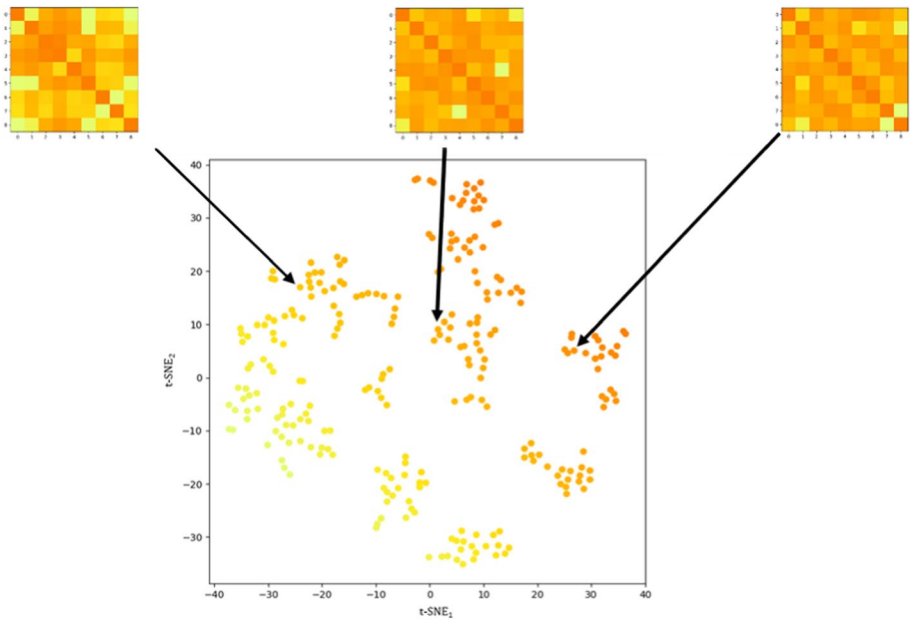

Fig. 2 Application of the Separability-Based Feature Space part of the GB-AFS method to 257 features from the Microsoft Malware Sample dataset. The features are embedded in a 2-dimensional space by t-SNE and are colored by their Jeffries-Matusita value

and $\tilde{c}$, where $1 \leq c, \tilde{c} \leq C$. Each T_i matrix is reshaped into a vector f_i of size $1 \times (C^2)$ to form the new feature space $\mathcal{Z}$. Subsequently, to visualize and organize the separability characteristics of the features, we employ a nonlinear dimensionality reduction technique to obtain the new feature space $\mathcal{Q}$.

In Fig. 2, you can see an example of the construction of a new feature space from the Microsoft Malware Prediction dataset [33], by using Jeffries-Matusita (JM) distance as a statistical measure and t-distributed Stochastic Neighbor Embedding (t-SNE) as a nonlinear dimensionality reduction technique. You can find more details about the JM distance in Section Jeffries–Matusita distance, and the t-SNE technique in Section t-distributed Stochastic neighbor embedding, respectively. The dataset is characterized by a composition of 257 features segmented into 9 classes, hence forming a representation through 257 matrices, each of size 9×9. A matrix is visualized in Fig. 2 as a 9×9 grid, where each entry represents the separation degree between two distinct classes. Pronounced separation is depicted in red, gradually transitioning to yellow as the separation narrows.

Clustering evaluation for feature selection using MSS

The GB-AFS aims to identify the minimal subset of k_{min} features that retain the ability of the entire M features to separate and distinguish different classes. To identify this subset, we follow a two-step procedure for every $k \in [2, M]$ to obtain a score reflecting the capability of a k-sized feature subset to represent the entire feature space's ability to separate and distinguish between different classes.

In the first step, features are selected using the K-medoids [34] algorithm, which selects features from different regions in the low-dimensional space, with complementary separation capabilities. The K-medoids algorithm, however, has a significant drawback

in that it is sensitive to the initialization of centers. To mitigate this issue, we utilize the K-means++ [35] initialization algorithm, which initializes the algorithm more effectively by selecting the initial centers using a probability distribution based on the distances between data points. Since K-medoids is designed to minimize the sum of distances between features and the nearest medoids, the objective of the second step is to use a measure to evaluate the effectiveness of the k subset of features obtained from the K-medoids algorithm in representing the entirety of the feature space by also considering the separation between clusters. For this goal, we propose a new metric, named Mean Simplified Silhouette (MSS) index, which is a variation of the SS index that evaluates the clustering outcome in the context of feature selection in classification problems as explained in the next paragraph. By performing multiple runs of K-medoids for all values of k in the range $[2, M]$, we can identify the minimal subset of k_{min} features that most effectively represents the original feature set and maintain the classification performance when using the entire set of features.

Mean Simplified Silhouette

The goal of feature selection is to identify a subset of features that cover the entire feature space with complementary capabilities, while avoiding the selection of redundant features with the same separation capabilities as the chosen ones. Existing clustering evaluation indices, such as Silhouette and Simplified Silhouette as explained in Section Silhouette, typically evaluate how closely a point is associated with its own cluster or its cluster's medoid, and how distinct it is from the closest cluster to which it does not belong. Furthermore, those indices usually set the value of a point to zero [31, 36] if it happens to be the sole point present in a cluster, which causes the indices to tend to zero as the number of clusters in the space approaches to the number of features in the space. However, when considering clustering outputs in the context of the feature selection task, our goal is to create an index that effectively measures the separation between a given point and all clusters to which it does not belong. This index aims to quantify the extent to which the chosen points encompass the entire feature space comprehensively. Moreover, it considers the proximity of each point within a cluster to its designated representative point, assessing the representative point's capability to capture the characteristics of the other points it represents. Furthermore, in the case where each point constitutes an individual cluster, our metric should reach its highest possible value, indicating complete coverage of the feature space. This approach aims to enhance the feature selection process by ensuring both comprehensive coverage and accurate representation within the feature space.

Our MSS index effectively addresses the limitations associated with both the Silhouette and SS indices. The proposed MSS calculates the distances from each point to all other cluster medoids within the feature space, excluding the medoid of the cluster to which the feature belongs. This modification enables a more reliable assessment of the selected features and ensures that they are sufficiently diverse and complementary to provide full coverage of the entire feature space. Additionally, unlike the other clustering evaluation indices, we exclude clusters that have only one feature in the MSS calculation. This exclusion is based on the rationale that isolating a feature which is significantly distant from its cluster's center to form a new,

single-feature cluster should enhance the clustering outcome and increase the MSS index. Through these modifications, the MSS index becomes an appropriate index for assessing the performance of clustering algorithms within the context of feature selection. It guarantees an optimal arrangement of the feature space, facilitating the identification of a group of features that exhibit complementary characteristics.

The MSS index is calculated based on the distances between each feature and the medoids of each cluster, similar to the SS index. It differs from the Silhouette index, which calculates how close each feature in a cluster is to features in its own cluster compared to features in other clusters. We believe that our approach is preferable because it is more reasonable to calculate distances from the medoids, which are the representative features, rather than the excluded features. In addition, MSS reduces the computational complexity, i.e., when computing distances from the medoids, the computational complexity is estimated as $O(kRM)$, as opposed to $O(RM^2)$, when computing distances from all features within each cluster. This difference is significant when k is much smaller than the number of features in the feature space, M.

To compute the MSS index, we begin by defining the values $a(i)$ and $b(i)$ for every point i in the dataset. The value of $a(i)$ corresponds to the distance $d(\cdot,\cdot)$ between point i and the center of the cluster to which it belongs, C_h, whereas the value of $b(i)$ denotes the average distance of point i from the centers of all other clusters $C_l, l \neq h$; namely:

$$a(i) = d(x_i, C_h) \tag{10}$$

$$b(i) = \underset{l \neq h}{\text{average}}\ d(x_i, C_l) \tag{11}$$

$$mss(i) = 1 - \frac{a(i)}{b(i)} \tag{12}$$

where the MSS index is the average of the MSS coefficients over all data points.

Figure 3 presents the Silhouette, SS, and MSS values depicted as solid lines and the accuracy obtained by three classifiers depicted as dashed lines over different values of k. These results were obtained by executing the first two stages of GB-AFS (Fig. 1) on the Microsoft Malware Prediction dataset [33]. It can be observed that as the value of k increases, both the Silhouette and SS values decrease rapidly toward zero, indicating an increase in single-point clusters. Conversely, the adjustments made to the MSS index enable the assessment of how effectively the clustering algorithm utilizes the feature space, for the feature selection problem. Moreover, a clear correlation was observed between the MSS index and the accuracy results of the classifiers, which emphasizes the effectiveness of the MSS index in solving feature selection tasks in classification problems.

It is important to note that our objective is not solely focused on pinpointing the k value that results in the highest accuracy value. Our objective is also to find the smallest k value that is sufficient for obtaining acceptable accuracy. Even if a higher k value may lead to marginally better accuracy, it may demand excessive resources and computational power, which would not be practical or worthwhile. That is why we

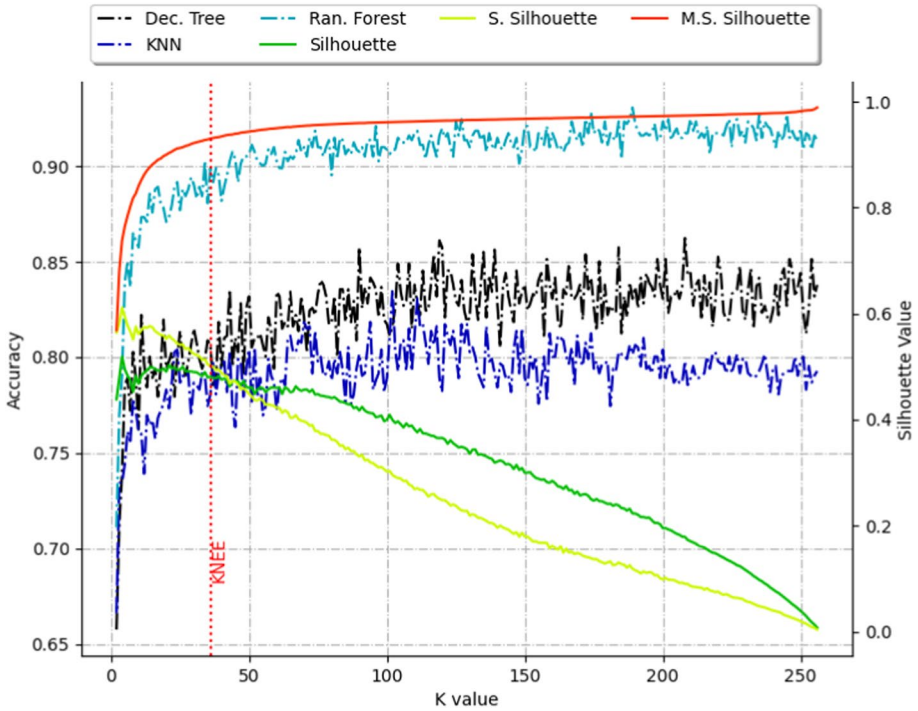

Fig. 3 Silhouette, SS, MSS and accuracy results obtained by three classifiers over different values of *k* on the Microsoft Malware Sample dataset. A vertical dotted line represents the minimum *k* value, the "knee" point found by the Kneedle algorithm

utilize the Kneedle algorithm developed by Satopaa *et al.* [32] as explained in the next section. It enables us to achieve a balance between accuracy and resource usage.

Optimal k determination

In this third stage, our goal is to determine the minimal subset of k_{min} features that can classify different classes effectively without incurring a drop in performance. As presented in Section Clustering evaluation for feature selection using MSS, the MSS index exhibits a correlation with the accuracy results over all possible values of k. Thus, in this stage of the proposed GB-AFS method, illustrated in Fig. 1, we apply the Kneedle algorithm to the MSS graph to find the minimal subset of k_{min} features.

Applying the Kneedle algorithm to the MSS graph enables the identification of the knee point, as illustrated in Fig. 3 by the vertical dashed line. This knee point corresponds to a specific k value representing the minimal number of features needed for classification. Subsequently, the k medoids associated with this k value are retrieved as the minimal subset of features required for classification.

Time Complexity of the GB-AFS method: Constructing the separation matrix has a computational complexity of $\mathcal{O}(NM)$. Dimension reduction on this matrix, in our experiments done using t-SNE, has a computational complexity of $\mathcal{O}(M^2)$ [28]. In the clustering evaluation phase, for each potential feature subset size k, we perform K-medoids clustering with computational complexity of $\mathcal{O}(M^2k)$ and evaluate cluster quality via the MSS index with a computational complexity of $\mathcal{O}(Mk)$, leading to a total time complexity dominated by $\mathcal{O}(M^2k)$ for a given k ; We are doing clustering

Table 1 Overview of the datasets

Dataset	#Instances	#Classes	#Features
Isolet [37]	7797	26	617
Cardiotocography [38]	2126	10	23
Mice Protein Expression [39]	1080	8	77
Music Genre Classification [40]	1000	10	197
Microsoft Malware Sample [33]	1642	9	257

evaluation for each $k \in [2, M]$, which results in $\mathcal{O}(M^3)$. Then, the optimal k value is found using the Kneedle algorithm with a computational complexity of $\mathcal{O}(M)$, which results with GB-AFS method's overall complexity of $\mathcal{O}(M^3)$.

Experimental results

Experiment setup

Datasets

Datasets quality and relevance are crucial in the extensive data analysis field. We have carefully chosen five datasets originating from different domains to ensure broad coverage of diverse scenarios and challenges. Table 1 presents, for each dataset, the number of instances, the number of features, and the number of classes. Below is a short description of each dataset. To conduct a comprehensive analysis, these datasets were chosen specifically for their variability in several dimensions, including the number of features, samples, classes, and level of class imbalance.

Isolet is an imbalanced dataset of 617 voice recording features from 150 subjects reciting the English alphabet, with the goal of classifying the correct letter among the 26 classes.

Cardiotocography comprises 23 distinct assessments of fetal heart rate (FHR) and uterine contraction (UC) characteristics, as documented on cardiotocograms. They were categorized into 10 separate classes by experienced obstetricians.

Mice Protein Expression measures the expression levels of 77 proteins in the cerebral cortex of eight classes of mice undergoing context fear conditioning for evaluating associative learning.

Music Genre Classification is a dataset that offers 1000 labeled audio snippets, each lasting 30 s. With 197 distinct features characterizing each sample, it's a favored choice for machine-learning endeavors aimed at discerning among 10 varied music genres. In total, the dataset presents 1000 samples, categorized into 10 classes, detailed by 197 features.

Microsoft Malware Prediction presents 257 file attributes spread across 9 distinct malware identification classes. Its primary design intention is to serve as a foundation for constructing machine-learning models aimed at malware prediction. This data collection boasts 1642 samples, detailed with 9 classes and 247 features.

Parameter settings

Section Separability-based feature space outlines the first stage of our proposed method. This stage involves preprocessing the input data and transitioning it into a

reduced feature space. This reduced space preserves the original features' capability to distinguish between each pair of classes. The GB-AFS method is designed as an agnostic solution that is not tied to any particular statistical measure or dimensionality reduction technique; it allows users to make these choices. In our experiments, to evaluate the difference between the two probability distributions, we employed the Jeffries-Matusita (JM) distance. Further details on this are provided in Section Jeffries–Matusita distance. Additionally, we select the t-SNE for the nonlinear dimensionality reduction technique. Further details on this are provided in Section t-distributed Stochastic neighbor embedding. Other class separability measures and nonlinear dimension reduction techniques may be used in the first stage of the GB-AFS method. In Section Method generalization, a sensitive analysis is conducted to evaluate the generalization of the GB-AFS method when employing a variety of dimensionality reduction techniques and statistical measures.

For K-medoids clustering, we used the PAM method for cluster assignment with K-means++ initialization. This method selects initial medoids farthest from each other, simulating the idea of choosing features with complementary separation capabilities, which improves the algorithm's efficiency and accuracy.

Baseline methods

The predictive efficacy of our GB-AFS method is evaluated against a total of six state-of-the-art methods.

The first method is the MGFS, as introduced by Amin et al. [20]. This method requires user input to determine the size of the feature subset. Here, we employ the $k_{\min}$ value derived from our GB-AFS method. To execute the method, we utilized the author's official repository[1].

The second method is the Inf-FS, proposed by Roffo et al. [23]. The official repository[2] for this method lacks the implementation of the minimal k-selection step, thus we again utilize the $k_{\min}$ value from our GB-AFS method in this case.

The third method is the S^3F, proposed by Thiago et al. [24]. Unlike the two previous methods, S^3F includes an internal mechanism for selecting the optimal k value during its execution, which means the chosen k may differ from our method.

The fourth method is the ReliefF method [41], which ranks the importance of each feature by measuring how well it distinguishes between instances of different classes while considering the proximity of those instances to each other. This method also requires the user input to determine the size of the feature subset, hence we use the $k_{\min}$ value obtained from our method.

The fifth method is Minimum Redundancy Maximum Relevance (mRMR) method [42], which selects features that are both highly relevant to the target and minimally overlapping, optimizing for predictive power and information uniqueness. Determining the feature subset size in this method requires user intervention. Similar to some of the previously explained methods, we adopted the $k_{\min}$ value obtained by our GB-AFS method.

[1] Repository link: MGFS.

[2] Repository link: Inf-FS.

The sixth method is Correlation-based feature selection (cFS) method [43], which identifies and selects features that are highly correlated with the target variable but minimally correlated with each other, aiming to improve performance by reducing redundancy. For this method, the user should also define the feature subset size as input, similar to previous methods. Thus, the $k_{\min}$ parameter found by the proposed GB-AFS method, was employed.

Experiment method

To avoid features with large numerical ranges from dominating those with small numerical ranges, the data were rescaled to lie between 0 and 1 using the min-max normalization procedure. Then, we split the data randomly such that 75% of the instances (training dataset) were used for applying GB-AFS to determine the set of k_{min} features and build the classifiers. The remaining 25% (test dataset) was used to evaluate the performance of the GB-AFS and resulting classifiers.

To find the k_{min}, we evaluated the MSS over a validation dataset for each set of k features found by applying the GB-AFS on the training set, where $k \in [2, M]$. To reduce the bias when selecting the training and validation data, we used a five-fold cross-validation approach [44], where 80% of the dataset was used to identify the set of k features and the remaining 20% was used for MSS calculation. For each value of k, we calculated the MSS five times, each time using a different subset as the validation dataset. We then averaged these values to generate an averaged MSS graph. Next, we applied the Kneedle algorithm[3] to the averaged MSS graph to obtain the value of k_{min}, which represents the number of features in the final dataset.

After determining k_{min}, we applied GB-AFS to the entire training set to obtain the set of k_{min} features and construct three different classifiers (KNN, Decision Tree and Random Forest) based on the chosen features. It should be noted that these classifiers were chosen randomly to enable the evaluation of our method in relation to other methods. The trained classifiers were then employed to classify instances in the out-of-sample test set and evaluate the accuracy and balanced F-score metrics. To evaluate the statistical significance of the results in comparison to the benchmarked methods, we repeated the entire experiment's methodology 10 times, using a different random split of the training–test sets for each iteration.

Experiment results

We evaluated the performance of our proposed GB-AFS method based on two key metrics: Accuracy and Balanced F-score. The performance of the classifiers was determined by calculating the average values over 10 test sets, as described in Sect. Experiment method. The accuracy and balanced F-score of the proposed GB-AFS were compared to ReliefF, mRMR, cFS, Inf-FS, and MGFS, using the same k value found by GB-AFS, since they take the number of selected features as an input parameter. The results of S^3F are obtained for the k value determined as part of the method. These results are reported in Table 2. For each dataset and classifier, the results of the best method are shown in bold. Since performance results were generated for all feature selection methods in each

[3] Repository link: Kneed.

Table 2 Results over the five datasets. A comparison of the accuracy and balanced F-score results of our method vs. state-of-the-art methods. The results of the best method are in bold for each dataset and experimental setup. Paired *t*-test significance at p-value < 0.05 indicated by underline

	GB-AFS (k=7)		S^3F (k=5)		Inf-FS (k=7)		MGFS (k=7)		ReliefF (k=7)		mRMR (k=7)		cFS (k=7)	
Cardiotocography	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**
KNN	0.585	**0.551**	0.527	0.505	**0.588**	0.550	0.566	0.542	0.470	0.438	0.523	0.517	0.459	0.437
D.T[1]	**0.675**	**0.688**	0.586	0.614	0.661	0.671	0.634	0.626	0.561	0.561	0.587	0.607	0.544	0.537
R.F[2]	**0.746**	**0.761**	0.664	0.687	0.732	0.733	0.727	0.729	0.633	0.629	0.681	0.690	0.617	0.611
Mice Protein Expression	**GB-AFS (k=16)**		**S^3F (k=8)**		**Inf-FS (k=16)**		**MGFS (k=16)**		**ReliefF (k=16)**		**mRMR (k=16)**		**cFS (k=16)**	
	Acc[3]	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**
KNN	**0.553**	**0.582**	0.505	0.555	0.533	0.569	0.536	0.558	0.519	0.527	0.522	0.525	0.508	0.515
D.T[1]	**0.518**	0.516	0.489	0.496	0.517	**0.525**	**0.518**	0.504	0.500	0.499	0.501	0.501	0.492	0.498
R.F[2]	**0.696**	**0.690**	0.625	0.673	0.679	0.673	0.669	0.671	0.656	0.661	0.666	0.659	0.648	0.651
Microsoft Malware Sample	**GB-AFS (k=32)**		**S^3F (k=11)**		**Inf-FS (k=32)**		**MGFS (k=32)**		**ReliefF (k=32)**		**mRMR (k=32)**		**cFS (k=32)**	
	Acc[3]	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**
KNN	0.786	**0.728**	0.758	0.701	0.772	0.722	**0.788**	0.722	0.783	0.719	0.776	0.710	0.757	0.703
D.T[1]	**0.809**	**0.785**	0.774	0.733	0.781	0.745	0.785	0.763	0.771	0.756	0.767	0.749	0.742	0.748
R.F[2]	**0.900**	**0.795**	0.852	0.738	0.878	0.764	0.877	0.766	0.833	0.760	0.841	0.766	0.825	0.740
Music Genre Classification	**GB-AFS (k=33)**		**S^3F (k=12)**		**Inf-FS (k=33)**		**MGFS (k=33)**		**ReliefF (k=33)**		**mRMR (k=33)**		**cFS (k=33)**	
	Acc[3]	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**
KNN	**0.476**	**0.551**	0.368	0.494	0.455	0.533	0.457	0.528	0.414	0.413	0.433	0.507	0.429	0.499
D.T[1]	**0.374**	**0.456**	0.314	0.398	0.360	**0.456**	0.355	0.453	0.272	0.290	0.326	0.422	0.319	0.428
R.F[2]	**0.507**	**0.525**	0.487	0.471	0.483	0.507	0.493	0.476	0.463	0.434	0.474	0.461	0.473	0.452
Isolet	**GB-AFS (k=44)**		**S^3F (k=15)**		**Inf-FS (k=44)**		**MGFS (k=44)**		**ReliefF (k=44)**		**mRMR (k=44)**		**cFS (k=44)**	
	Acc[3]	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**	**Acc[3]**	**B.F[4]**
KNN	**0.837**	**0.815**	0.708	0.712	0.824	0.766	0.818	0.779	0.800	0.782	0.795	0.773	0.786	0.762
D.T[1]	**0.859**	**0.802**	0.711	0.733	0.837	0.758	0.833	0.771	0.803	0.737	0.799	0.746	0.788	0.735
R.F[2]	**0.875**	**0.807**	0.728	0.747	0.849	0.769	0.852	0.789	0.814	0.779	0.811	0.769	0.803	0.771

[1]Decision Tree, [2]Random Forest, [3]Accuracy, [4]Balanced F-score

run among the 10 runs, a paired t-test [45] is employed to compare the result of the best feature selection method with the second-best method in each combination of dataset and classifier. Statistically significant differences at a p-value < 0.05 are indicated by underline.

Table 3 The running times and the percentage of time saved when using a set of k_{min} features obtained by GB-AFS in comparison to the running times when using all features

		Knee point	All features	Estimated time saving
Cardiotocography (k=7)	KNN	0.392	0.518	24.32%
	D. Tree[1]	0.485	0.521	6.91%
	R. Forest[2]	0.523	0.599	12.68%
Mice Protein Expression (k=16)	KNN	0.491	0.799	38.54%
	D. Tree[1]	0.908	1.022	11.15%
	R. Forest[2]	0.909	1.189	23.54%
Microsoft Malware Sample (k=32)	KNN	0.734	1.763	58.36%
	D. Tree[1]	2.152	2.989	28.02%
	R. Forest[2]	2.747	3.723	26.21%
Music Genre Classification (k=33)	KNN	0.624	1.383	54.88%
	D. Tree[1]	2.101	2.624	19.93%
	R. Forest[2]	2.957	3.759	21.33%
Isolet (k=44)	KNN	0.888	3.654	75.69%
	D. Tree[1]	2.551	8.750	70.84%
	R. Forest[2]	4.415	11.805	62.60%

[1]Decision Tree, [2]Random Forest

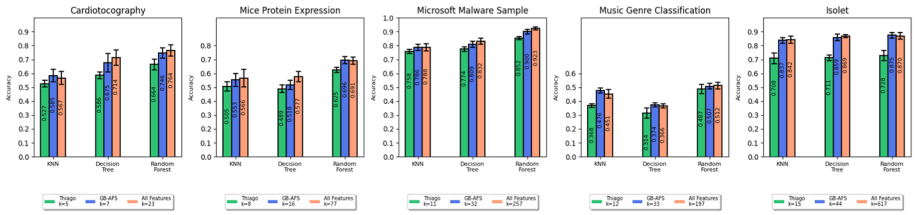

Fig. 4 Comparison of the Accuracy of various classifiers utilizing k features selected by the GB-AFS with the MSS index, k features selected by the S^3F method and the complete feature set available in the dataset

Across all combinations of datasets and classifiers, the GB-AFS shows an average accuracy improvement of 7.5% and an average balanced F-score improvement of 7.7% compared to the other methods. In 12 out of 15 combinations of datasets and classifiers, we observed that the GB-AFS method performs better in terms of accuracy compared to the other methods, showing improvements ranging from 1.6% to 4.2%, with an average improvement of 2.8% compared to the second-best method in each combination. In 11 of these 12 combinations, GB-AFS outperformed the other filter methods with a statistically significant difference (p-value < 0.05). In terms of the Balanced F-score, we observe that the GB-AFS method achieved better results in 13 of the 15 combinations of datasets and classifiers, showing improvements ranging from 0.2% to 4.2%, with an average improvement of 2.8% compared to the second best-method in each combination. Of these 13 combinations, 11 were found to be statistically significant.

Figure 4 displays the average accuracy of the proposed GB-AFS method with a ±95% confidence interval, in comparison to S^3F method, which also automatically finds the

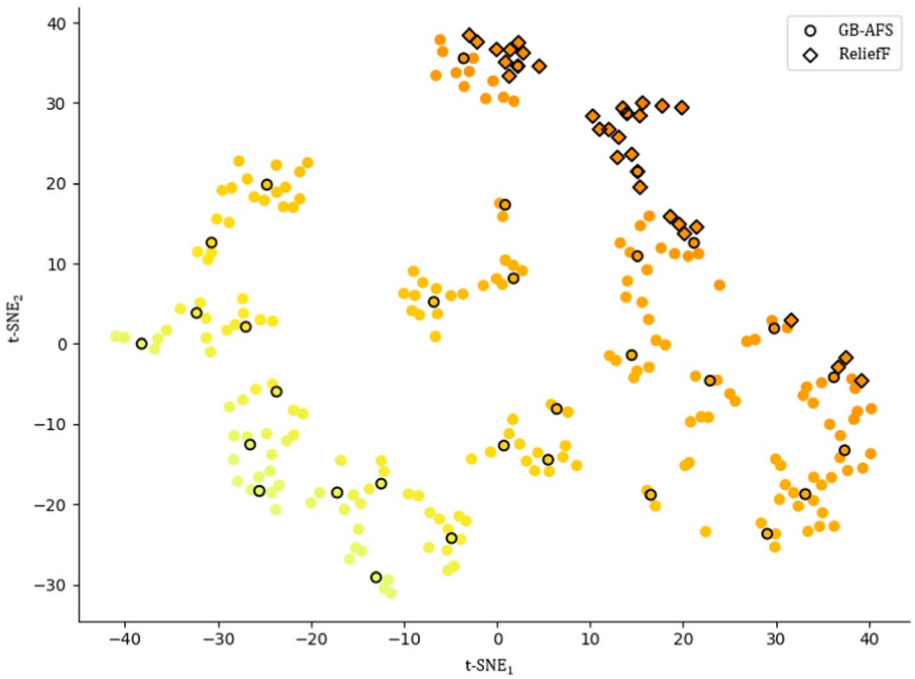

Fig. 5 Microsoft Malware Sample features are color-coded according to their separation capabilities score. Our algorithm found $\frac{32}{257}$ features based on their complementary separation abilities, while ReliefF marked features with similar values and not necessarily with complementary abilities

best k value according to their implementation. The results show that the GB-AFS method achieved significantly better accuracy than when incorporating the S^3F method, with an average improvement of 12.7%. Moreover, the GB-AFS method selected between 7% and 30% of the features in each dataset, while in 14 of 15 combinations of datasets and classifiers, there was no statistically significant difference (p-value < 0.05) in accuracy results between the GB-AFS method and when using all features. Although the accuracy results are similar, the percentage of time saved on average by using a set of k_{min} features ranges from 15% for the smallest dataset to 70% for the largest (Table 3).

Graphical analysis and interpretation

This section illustrates the ability of the GB-AFS method to select features with complementary discriminating capabilities, effectively covering the entire feature space. Figure 5 shows the features of the Microsoft Malware dataset embedded in low-dimensional space accepted by using the JM and t-SNE methods. Each feature i is assigned a color based on its separation capabilities, in such a way that the higher the value, the darker the color tends to be. The k_{min} subset of features chosen by the ReliefF method and the GB-AFS method, assuming that k_{min} was found by our proposed method, are indicated by a rhombus and a circle, respectively. As can be seen from Fig. 5, the ReliefF method tends to select features from the upper right corner of the graph with high separation capabilities, wheras the GB-AFS method selects features that span the entire graph, indicating a wider range of separation capabilities. A similar behavior observed in Fig. 5 regarding the ReliefF method, was also detected in the results obtained from the other benchmarked methods across all datasets.

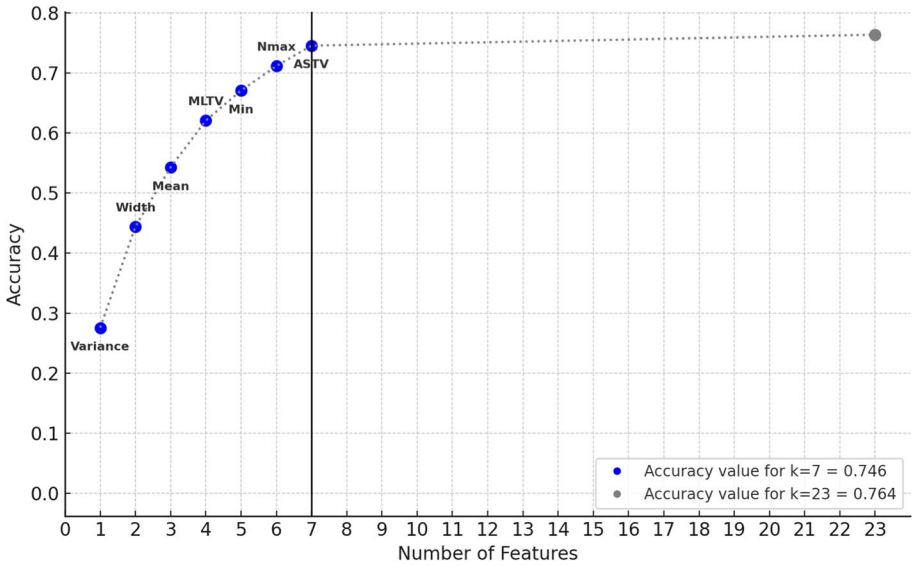

Fig. 6 The results of the GB-AFS run on the Cardiotocography dataset, which led to the selection of a feature subset of 7 features, out of 23 original features. The names of these selected features are provided for reference

Figure 6 presents the ability of the GB-AFS to select a low number of features while preserving the accuracy obtained when using the entire dataset based on the Cardiotocography dataset, using Random Forest classifier. In the graph, the contribution of each one of the 7 selected features, out of the original 23 features, to the obtained accuracy is presented. These selected features are marked with blue dots on the graph. It can be observed from the graph that using only 7 selected features by the GB-AFS method results in an accuracy of 0.746, which is slightly lower than the 0.764 accuracy achieved using all 23 features. This modest decrease in accuracy, while only using about 30% of the available features, highlights GB-AFS's capability to preserve high levels of performance with a significantly reduced number of features.

Method generalization

The GB-AFS method as illustrated in Fig. 1, doesn't assume specific statistical measures to generate the feature separability matrix and specific dimensional reduction techniques to obtain the low-dimensional feature space. Table 4 presents the results of employing the GB-AFS method with various pairings of dimensionality reduction techniques and statistical measures. For this goal, we utilized t-SNE [28], UMAP [46], and diffusion maps [19] as dimensional reduction techniques and JM distance [26, 27], Wasserstein distance [47], and Hellinger distance [48] as statistical measures.

The performance of the classifiers over the datasets for the combination of UMAP and Wasserstein and the combination of diffusion maps and Hellinger in Table 4 was determined by calculating the average values over two runs, while the performance of the combination of t-SNE and JM is directly sourced from Table 2. Upon reviewing the results of Table 4, it can be observed that the GB-AFS yielded similar performances for every combination of dimensional reduction technique and statistical measure, which are superior in most cases to the compared methods in Table 2. More specifically, each

Table 4 A comparison between the accuracy and balanced F-score outcomes of our proposed method, incorporating various combinations of dimensionality reduction techniques and statistical measures. Best-performing methods for each dataset and setup are in bold

Cardiotocography						
	t-SNE + JM (k=7)		**UMAP + Wasserstein (k=9)**		**Diffusion Map + Hellinger (k=7)**	
	Accuracy	**B.F[3]**	**Accuracy**	**B.F[3]**	**Accuracy**	**B.F[3]**
KNN	0.585	**0.551**	**0.589**	0.549	0.569	0.544
D.T[1]	**0.675**	**0.688**	0.671	0.680	**0.675**	0.681
R.F[2]	0.746	0.761	**0.753**	**0.772**	0.749	0.757
Microsoft Malware Sample						
	t-SNE + JM (k=32)		**UMAP + Wasserstein (k=31)**		**Diffusion Maps + Hellinger (k=34)**	
	Accuracy	**B.F[3]**	**Accuracy**	**B.F[3]**	**Accuracy**	**B.F[3]**
KNN	0.786	**0.728**	0.781	0.722	**0.793**	**0.728**
D.T[1]	**0.809**	**0.785**	0.800	0.780	0.804	0.777
R.F[2]	**0.900**	0.795	0.891	0.784	0.899	**0.801**

[1]Decision Tree, [2]Random Forest, [3]Balanced F-score

combination of dimensional reduction technique and statistical measure obtained better balanced F-score and accuracy in 4 to 5 out of 6 combinations of datasets and classifiers. These results highlight the generality of the GB-AFS method with respect to the dimensional reduction technique and statistical measure used. It emphasizes the strength of the GB-AFS method in its uniqueness in selecting features with complementary abilities, a strategy that proves effective across diverse datasets.

Conclusion and future work

This paper presents a novel graph-based filter method for automatic feature selection (GB-AFS) for multi-class classification problems. An algorithm is developed to apply the proposed method to find the minimal subset of features that are required to retain the ability to distinguish between each pair of classes. The experimental results on five popular datasets and three classifiers show that the proposed algorithm outperformed other state-of-the-art filter methods with an average accuracy improvement of 7.5%. Moreover, in 14 of 15 cases, the GB-AFS method was able to identify 7% to 30% of the features that retained the same level of accuracy as when using all features, while reducing the classification time on an average from 15% for the smallest dataset to 70% for the largest. These findings highlight the method's efficiency in handling high-dimensional datasets.

While our study has demonstrated promising results, it is essential to consider certain limitations and opportunities for further research. Firstly, our approach is specifically tailored for tabular datasets. Secondly, our methodology does not consider the class distribution within the dataset. Future research could explore adapting the proposed method to other problems involving diverse datasets, such as audio or EEG signals [49].

Additionally, it would be interesting to investigate methods for incorporating the class distribution within the dataset into the construction of the separation matrix. Another potential direction could involve utilizing the proposed method for constraint-based classification problems, a common research challenge that has received significant attention recently [50, 51]. In many real-world situations, each feature incurs an economic cost for collection, with limited resources being available to tackle the problem at hand. Thus, the adaption of the proposed algorithm, such that the total cost of the selected features meets the constraint, is an interesting direction to explore.

Acknowledgements
The authors are deeply thankful to the editor and reviewers for their valuable suggestions to improve the quality and presentation of the paper.

Funding
None.

Availability of data and materials
The data is given in the paper.

Code availability
The official repository is given in the paper.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
The authors give the Publisher permission to publish the work.

Competing interests
No conflict of interest regarding the paper.

Received: 23 November 2023 Accepted: 7 May 2024
Published online: 31 May 2024

References

1. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007;23(19):2507–17.
2. Liu H, Motoda H. Feature Selection for Knowledge Discovery and Data Mining. vol. 454. Springer, 2012.
3. Tadist K, Najah S, Nikolov NS, Mrabti F, Zahi A. Feature selection methods and genomic big data: a systematic review. J Big Data. 2019;6(1):1–24.
4. Chen R-C, Dewi C, Huang S-W, Caraka RE. Selecting critical features for data classification based on machine learning methods. J Big Data. 2020;7(1):52.
5. Li J, Liu H. Challenges of feature selection for big data analytics. IEEE Intell Syst. 2017;32(2):9–15.
6. Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014;40(1):16–28.
7. Miao J, Niu L. A survey on feature selection. Procedia Comput Sci. 2016;91:919–26.
8. Pereira RB, Plastino A, Zadrozny B, Merschmann LH. Categorizing feature selection methods for multi-label classification. Artif Intell Rev. 2018;49:57–78.
9. Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H. Feature selection: a data perspective. ACM Comput Surveys (CSUR). 2017;50(6):1–45.
10. Jović A, Brkić K, Bogunović N. A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015;1200–1205. IEEE
11. Venkatesh B, Anuradha J. A review of feature selection and its methods. Cybern Inf Technol. 2019;19(1):3–26.
12. Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. In: Icml, 1997;97: 35. Citeseer
13. Pudjihartono N, Fadason T, Kempa-Liehr AW, O'Sullivan JM. A review of feature selection methods for machine learning-based disease risk prediction. Front Bioinformatics. 2022;2: 927312.
14. Hruschka ER, Campello RJ, De Castro LN. Evolving clusters in gene-expression data. Inf Sci. 2006;176(13):1898–927.
15. You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y. Graph contrastive learning with augmentations. Adv Neural Inf Process Syst. 2020;33:5812–23.
16. Cai D, He X, Han J, Huang TS. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell. 2010;33(8):1548–60.
17. Briola A, Aste T. Topological feature selection: a graph-based filter feature selection approach. arXiv preprint arXiv:2302.09543 2023.

18. Friedman S, Singer G, Rabin N. Graph-based extreme feature selection for multi-class classification tasks. arXiv preprint arXiv:2303.01792 2023.
19. Coifman RR, Lafon S. Diffusion maps. Appl Comput Harmon Anal. 2006;21(1):5–30.
20. Hashemi A, Dowlatshahi MB, Nezamabadi-Pour H. Mgfs: a multi-label graph-based feature selection algorithm via pagerank centrality. Expert Syst Appl. 2020;142: 113024.
21. Xing W, Ghorbani A. Weighted pagerank algorithm. In: Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004., 2004;305–314. IEEE
22. Parlak B, Uysal AK. A novel filter feature selection method for text classification: extensive feature selector. J Inf Sci. 2023;49(1):59–78.
23. Roffo G, Melzi S, Castellani U, Vinciarelli A, Cristani M. Infinite feature selection: a graph-based feature filtering approach. IEEE Trans Pattern Anal Mach Intell. 2020;43(12):4396–410.
24. Covões TF, Hruschka ER. Towards improving cluster-based feature selection with a simplified silhouette filter. Inf Sci. 2011;181(18):3766–82.
25. Wang F, Franco-Penya H-H, Kelleher JD, Pugh J, Ross R. An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity. In: Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13, 2017;291–305. Springer.
26. Wang Y, Qi Q, Liu Y. Unsupervised segmentation evaluation using area-weighted variance and Jeffries-Matusita distance for remote sensing images. Remote Sens. 2018;10(8):1193.
27. Tolpekin VA, Stein A. Quantification of the effects of land-cover-class spectral separability on the accuracy of Markov-random-field-based superresolution mapping. IEEE Trans Geosci Remote Sens. 2009;47(9):3283–97.
28. Maaten L, Hinton G. Visualizing data using t-sne. J Mach Learning Res. 2008;9(11).
29. Hinton GE, Roweis S. Stochastic neighbor embedding. Adv Neural Inf Proc Syst. 2002;15.
30. Van Der Maaten L. Learning a parametric embedding by preserving local structure. In: Artificial Intelligence and Statistics, 2009;384–391. PMLR.
31. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65.
32. Satopaa V, Albrecht J, Irwin D, Raghavan B. Finding a "Kneedle" in a haystack: Detecting knee points in system behavior. In: 2011 31st International Conference on Distributed Computing Systems Workshops, 2011;166–171. IEEE.
33. Microsoft: Microsoft Malware Prediction. Kaggle 2019. https://www.kaggle.com/c/microsoft-malware-prediction/data.
34. Kaufman L, Rousseeuw PJ. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 2009.
35. Arthur D, Vassilvitskii S. K-means++ the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2007;1027–1035.
36. Hruschka ER, Covoes TF. 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), 2005;1: 32–38. IEEE.
37. Cole R, Fanty M. ISOLET. UCI Machine Learning Repository. 1994. https://doi.org/10.24432/C51G69.
38. Campos D, Bernardes J. Cardiotocography. UCI Machine Learning Repository. 2010. https://doi.org/10.24432/C51S4N.
39. Higuera C, Gardiner K, Cios K. Mice Protein Expression. UCI Machine Learning Repository. 2015. https://doi.org/10.24432/C50S3Z.
40. Olteanu A. GTZAN Dataset—Music Genre Classification. Kaggle 2020. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.
41. Robnik-Šikonja M, Kononenko I. Theoretical and empirical analysis of relieff and rrelieff. Mach Learn. 2003;53:23–69.
42. Radovic M, Ghalwash M, Filipovic N, Obradovic Z. Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. BMC Bioinformatics. 2017;18(1):1–14.
43. Hall MA. Correlation-based feature selection for machine learning. PhD thesis, The University of Waikato 1999.
44. Arlot S, Celisse A. A survey of cross-validation procedures for model selection. 2010.
45. Manfei X, Fralick D, Zheng JZ, Wang B, Changyong F, et al. The differences and similarities between two-sample t-test and paired t-test. Shanghai Arch Psychiatry. 2017;29(3):184.
46. McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. 2018. arXiv preprint arXiv:1802.03426.
47. Rüschendorf L. The Wasserstein distance and approximation theorems. Probab Theory Relat Fields. 1985;70(1):117–29.
48. Beran R. Minimum Hellinger distance estimates for parametric models. Ann Stat. 1977;445–463.
49. Haba R, Singer G, Naftali S, Kramer MR, Ratnovsky A. A remote and personalised novel approach for monitoring asthma severity levels from EEG signals utilizing classification algorithms. Expert Syst Appl. 2023;223: 119799.
50. Rabkin L, Cohen I, Singer G. Resource allocation in ordinal classification problems: a prescriptive framework utilizing machine learning and mathematical programming. Eng Appl Artif Intell. 2024;132: 107914.
51. Shifman DA, Cohen I, Huang K, Xian X, Singer G. An adaptive machine learning algorithm for the resource-constrained classification problem. Eng Appl Artif Intell. 2023;119: 105741.

Publisher's Note