Article

Unsupervised Feature Selection Using Recursive k-Means Silhouette Elimination (RkSE): A Two-Scenario Case Study for Fault Classification of High-Dimensional Sensor Data

Ahlam Mallak *, Madjid Fathi

Department of Electrical Engineering and Computer Science, Knowledge-based Systems and Knowledge Management, University of Siegen, 57076 Siegen, Germany; fathi@informatik.uni-siegen.de
* Correspondence: Ahlam.mallak@Ymail.com; Tel.: +49-1522-948-8004

Received: date; Accepted: date; Published: date

Abstract: Feature selection is a crucial step to overcome the curse of dimensionality problem in data mining. This work proposes Recursive k-means Silhouette Elimination (RkSE) as a new unsupervised feature selection algorithm to reduce dimensionality in univariate and multivariate time-series datasets. Where k-means clustering is applied recursively to select the cluster representative features, following a unique application of silhouette measure for each cluster and a user-defined threshold as the feature selection or elimination criteria. The proposed method is evaluated on a hydraulic test rig, multi sensor readings in two different fashions: (1) Reduce the dimensionality in a multivariate classification problem using various classifiers of different functionalities. (2) Classification of univariate data in a sliding window scenario, where RkSE is used as a window compression method, to reduce the window dimensionality by selecting the best time points in a sliding window. Moreover, the results are validated using 10-fold cross validation technique. As well as, compared to the results when the classification is pulled directly with no feature selection applied. Additionally, a new taxonomy for k-means based feature selection methods is proposed. The experimental results and observations in the two comprehensive experiments demonstrated in this work reveal the capabilities and accuracy of the proposed method.

Keywords: feature selection; k-means; silhouette measure; clustering; Big Data; fault classification; sensor data; time-series data.

1. Introduction

Human brains can only visualize and imagine three-dimensional spaces. Data with larger dimensions outpaces the human capacity to understand and manually analyse such data. In accordance with the human incapacity to deal with high volume data, data mining is brought to light to discover highly dimensional patterns in this large data, offering new solutions to visualize, analyse and process the big influx of information. Although, machine learning offers a lot of algorithms and methods for big data analysis, finding relevant and non-redundant attributes in data that contains hundreds or thousands of attributes can be challenging.

Irrelevant features may decrease the accuracy of the machine learning model learnt especially when they are many, because they can create an accumulative amount of deviation from the correct patterns. However, redundant features are the features that do not carry new information necessary for learning. So that, redundant feature may not affect the learning process and its accuracy, but indeed will increase the computational cost required for performing such tasks [1]. Therefore, when the dimensionality of the data is high without investigation and analysis of their redundancy and relevance, this would weaken the quality of the data for learning, increase its computational cost and diminish its accuracy. Moreover, besides redundant, and irrelevant features, noisy features can also contribute to degrading the training performance of various learning algorithms.

Feature selection in data mining is the process of choosing a subset of the overall features (variables) in the feature space, by sacrificing the ones carrying little valuable information, unnecessarily redundant ones, and noisy features [2]. Feature selection is most appropriate for multivariate datasets owning to their nature of numerous numbers of attributes and samples. Some examples of datasets require feature selection due to their complexity and size demands are text, genetic information, imaging modalities and time-series data.

Although, one might think the deployment of feature selection in any data mining task is completely optional, incorporating this step has a great potential to add-up many advantages. For instance, reducing the number of features can effectively improve the ability of data understanding and visualization, decrease the computational power and storage requirements needed, noticeably fasten the training and testing times. Finally, increasing the accuracy of the data mining model using smaller inputs and resources. Wherefore, feature selection is a substantial step to eliminate undesirable features, by selecting the ones that are more relevant, non-noisy and non-redundant [3,4].

Feature selection categories are determined based on two main characteristics. (1) The existence of samples' labels or classes, and (2) the search strategy used to select the features [4].

First, according to the label availability, feature selection methods can be classified into; supervised, semi-supervise and unsupervised feature selection methods [5]. Supervised feature selection algorithms depend on the existence of labels to efficiently choose the most discriminant and informative features, by effectively classifying the samples between the classes using the given labels. However, when only some of the data samples contain labels, while the rest are unlabelled, in this case the feature selection is called semi-supervised. The most spontaneous, natural, and common form of data is the unlabelled form. Where the records are stored from their source naturally without being combined with any observations, or any sort of grouping or notations. Due to the absence of guidelines and clues, unsupervised feature selection is considered by far the most difficult type comparing to the former two [6].

Second, another way to determine the feature selection method category is by the process strategy used to select the features, which can be divided into filter, wrapper, and embedded methods. On one hand, filter methods are applied as a pre-processing stage prior to the actual feature selection using wrapping methods. The features are selected by performing various statistical tests to measure the correlation between each feature to find which are more relevant. Some examples of statistical tests to measure the correlation coefficient are, Pearson's correlation, Fisher transformation coefficient also known as F-test, Linear Discriminant Analysis (LDA), Chi-square, and many more. Filter methods utilize the shape of the data to determine the most valuable features. More specifically, by applying a certain condition, methods, or criteria to rank the features, then order them in a descending order based on the rank calculated while selecting the features highest in the order to represent the rest. The work in [7,8] and [9] are examples of filter methods to select features. On the other hand, wrapper methods relies on the continuous selection of various subsets of features from the feature space, and utilize them each to train a machine learning model and infer which subsets to choose and which to eliminate according to the resulting performance of the model. In other words, wrapper methods do not rely on the shape of the data as the filter methods, instead they use machine learning to select the features with the best accuracy when running the machine learning model.

Most of the feature selection algorithms proposed in the literature are classification-based techniques [10,11] and [12], where these methods are dependent on the presence of clear classes or labels to perform the feature selection accordingly, or a clear presence of heuristic information generated by various algorithms such as, genetic algorithms as in [13].

In the past few years, a new cluster-based methods has emerged, also known as unsupervised feature selection algorithms. Unsupervised feature selection methods work by grouping objects in various groups based on their similarity. Where better clusters are the ones with higher within-cluster similarities and lower between-clusters similarities [14,15].

One of the most famous clustering algorithms in data mining is k-means clustering. k-means depends on dividing the data between k main clusters, where the intra-similarity within the cluster and inter-similarity between clusters is measured using silhouette value measurement.

Although, the literature is enriched with numerous supervised learning feature selection methods, it is still scarce when it comes to unsupervised ones and it needs more investigation and research in this field. For this reason, this work introduces a new feature selection algorithm that depends on recursively cluster the data into k groups using k-means clustering, then calculate silhouette value for each member of the individual clusters to decide which feature is the representative for the rest of the cluster, and which are going to be re-clustered for further analysis.

Recently, feature selection research interest drastically shifted to unsupervised methods, mostly because of their strength in identifying relevant features without the need of existing class labels. k-means clustering is one of the most famous clustering algorithms deployed for feature selection due to its simplicity, the availability of numerous existing researches to accurately select various parameters of k-means algorithm. i.e. selecting the best k and initializing the seeds. Moreover, k-means is relatively easier to evaluate comparing to other clustering algorithms, since it has many clear measures to evaluate the quality of the clustering process such as the silhouette measure.

In this work, a new feature selection algorithm based on the deployment of k-means and silhouette measure in an iterative fashion along with a pre-defined threshold is described. The "Recursive k-Means Silhouette Elimination (RkSE)" is introduced, tested, and validated. Moreover, a new taxonomy for literature existing unsupervised feature selection algorithms is provided.

2. Background

2.1 k-means Clustering

k-means clustering algorithm was first created back in 1967 by James MacQueen, the detailed article is shown in [16] .Where data is divided into k number of clusters based on their connectivity and pattern. This method is optimal to find hidden patterns in unlabeled data.

The basic flow of any process must contain inputs, outputs, and the mechanism of the process itself. For k-means clustering algorithm. First, the inputs expected are the unlabeled dataset D, and a predefined number of clusters k. k can be chosen randomly or computed using various methods. In a recent study [17] various methods are introduced to calculate the most optimal number of clusters in k-means algorithm. The most common method to calculate k is using elbow method. The elbow method will be used later in the feature selection algorithm proposed in this chapter, so that it is essential to have some overview of the method and how it works. The fundamental idea of the elbow method is to calculate the sum of squared errors (SSE) for various k values iteratively. While the best k is represented by the k value with the first sudden drop in SSE value, in such way that it looks like an elbow when plotting k and the distortion or SSE.

Second, the process in which k-means clustering work as the following: (1) Randomly select k number of samples from the dataset D. These k samples are considered the seeds of the algorithm which can be chosen randomly or following some specific algorithms to initialize the seeds. (2) The chosen seeds will perform as the initial centroids of the k clusters, in which the distance between all the instances and these centroids is calculated. Each instance is grouped in the nearest centroid's cluster (Minimum centroid distance). (3) Perform iterative or repetitive centroids selection and distance calculations to optimize the centroids locations, to eventually have the best centroids locations that ensures better clustering. The regular equation used to calculate the new centroid each iteration is explained below:

$$C_{i\_new} = \left(\frac{1}{N_i}\right) \sum_{i}^{N_i} x_i, \tag{1}$$

Where N_i is the number of members or instances in the cluster i. And the calculation of C_{i_new} is simply computed by finding the mean point or object of each cluster.

2.2 Silhouette Value Literature

Clustering in general, is the process of defining groups of objects. In a way that, in each group objects tend to be like one another, and different from other objects located in other clusters or groups. How to evaluate these clusters? A high-quality cluster usually have a high intra-cluster similarity value, and low inter-cluster value, which means the objects contained in this cluster are highly similar and connected to one another, and distinctively distant from other objects in other clusters. In other words, the cluster should be well-defined to be considered a high-quality cluster.

There are many similarity measures to compute the intra-cluster and inter-cluster similarity values. Such as, Silhouette value [18].

The Silhouette value for an object *i* in one of the k clusters computed using k-means clustering can be calculated as the following equation:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{2}$$

Where a_i is the average distance between the object *i* and all other objects that belongs to the same cluster. b_i is the minimum average distance between the object *i* and all other object in all neighbouring clusters. S_i is the Silhouette value of the object *i*. Based on the values of a_i and b_i using one of the equations above. S_i is a straightforward approach to know the similarity or dissimilarity measures between the object *i* and its group. If S_i is close to the positive end; positive one that means this object is highly similar to its cluster and appropriately grouped. When S_i is close to the negative end or negative one, then *i* is not appropriate to its original cluster, and the similarity between *i* and its neighbouring clusters are higher, so *i* should have been clustered in the neighbouring group instead. An S_i close to zero means that this object is located in between two main clusters and does not belong to any.

3. K-Means for Feature Selection Related Work

On one hand, the literature is rich with review research papers related to feature selection methods. However, the vast majority of these review papers are focused on supervised and semi-supervised methods. The research in [5,4] represent a thorough analysis of various supervised and semi-supervised algorithms, along with a quick glance at few unsupervised techniques for feature selection. In [19] an inclusive research is done investigating various semi-supervised techniques in various fields and applications. Finally, the work in [20] introduces a new perspective for supervised feature selection methods, including more recent studies and different taxonomies comparing to the ones described in the latter papers.

On the other hand, a few research studies concentrated their efforts to analyze unsupervised methods for feature selection such as, the work in [21] where they pointed out the lack of survey research in this area, and offered a detailed analysis of numerous unsupervised methods along with summarizing their advantages and disadvantages, as well as an experimental comparisons between them. The work in [22] narrowed down the scope of the research in [21] and instead, it focuses specifically on clustering algorithms for feature selection providing various clustering techniques for genetic, text, streaming and linked data. Moreover, they finalized their review with some challenges that clustering algorithms for feature selection witness and elaborated with some suggestions to overcome the proposed challenges.

In this paper, we narrowed down the scope even more, to include clustering feature selection algorithms using k-means clustering alone. This work is essential since k-means clustering for feature selection has already a huge amount of literature with different strategies and mechanisms, which creates the need to add some structure and taxonomy for this influx of studies, to facilitate navigating through them, as well as building up new literature following the legitimate path.

According the literature in the past decade, it is prominent that k-means for feature selection can be divided into the following main categories based on their clustering strategy and the included mechanisms.

- k-mean hybrid approaches: which includes a combination between k-means and other wrapper or filter feature selection methods. The work in [3,6], [23,24] and [25] demonstrate an example of hybrid wrapper k-means algorithms corresponded with various techniques of filter methods.
- k-means based on feature weighting or ranking: The general idea of k-means for feature selection based on feature weighting begins with clustering the dataset into k main clusters. Followed by, using variations of strategies to assign weights to each feature or features' subsets in some literature. In a way the feature or subset of features that minimizes the inter-cluster distance and maximizes the intra-cluster distance is assigned higher ranks or weights. The type of measurements or process responsible for assigning weights or ranks to a feature or a groups of features during clustering are called "Clustering criteria" [26]. Although, the literature introduced numerous clustering criteria, the oldest and most common ones are the silhouette criterion [18] and Davies-Bouldin index (DB) [27], where they contributed as base methods for modern weighting criteria nowadays. Hruscka and Covoes [28] introduced Simplified Silhouette Sequential Forward Selection (SS-SFS) approach for feature selection. In [29] a method called Entropy Weighting k-means (EWKM) is introduced to reduce the intra-cluster distortion and increase negative entropy throughout the clustering. In [26] a new method for unsupervised clustering criterion is introduced. In the recent work proposed [30] the authors introduced a ranking pipeline that includes k-means and various statistical approaches such as, signal-to-noise ratio, t-statistics and significance analysis to rank the features in a highly dimensional microarray. Furthermore, the research in [31,32] are also dependent on weighting or ranking to deploy k-means as a feature selection algorithm.
- k-means with correlation measures: In [33], a new perspective for feature selection using k-means is introduced, where a correlation measure between clusters is the selection or elimination criterion. The correlation measure is used to improve the quality of the feature subsets to be clustered using k-means. This method provides an elimination possibility of both irrelevant features using k-means, and redundant features using the correlation measure applied to each cluster. This method is validated by solving a classification problem using Naïve Bays classifier, applied on microarray and text datasets.

  Additionally, the work in [34] successfully integrated correlation-based k-means clustering to improve the accuracy of the computer-aided diagnosis specified with cardiovascular diseases.
- Sparse k-means feature selection methods: The research done in [35] explains the definition of sparse learning specialized in clustering algorithms for dimensionality reduction. One way to describe sparse learning in k-means is a form of matrix decomposition that yields the matrix A as a lower dimensional and more relevant partition of the original dataset X .Where X is a matrix of $n \times p$ size and it can be approximately decomposed to the matrices A and B, following the formula: $X \approx AB$, As A is a $n \times q$ size, and B is $q \times p$ matrix, known that $q \ll p$. Eventually, the clustering can be formed using the lower dimensional decomposition matrix A instead of the whole dataset X. In the last decade, Witten and Tibshirani [35] proposed a revolutionary framework for feature selection by introducing the concept of sparse clustering. Embedded Unsupervised Feature Selection (EUFS) [36] proposes a new idea of embedding the feature selection process within the clustering algorithm by the deployment of sparse learning. In [37] the research done is based on the novelty algorithm introduced in [36]. This method adopts a similar analogy to EUFS explained earlier. However, the recent work in [37] uses Frobenius-norm as the loss function.

Figure 1, shows the four main categories of k-means for unsupervised feature selection proposed in previous literature review. Based on the acquired knowledge and understanding of the work in the literature.

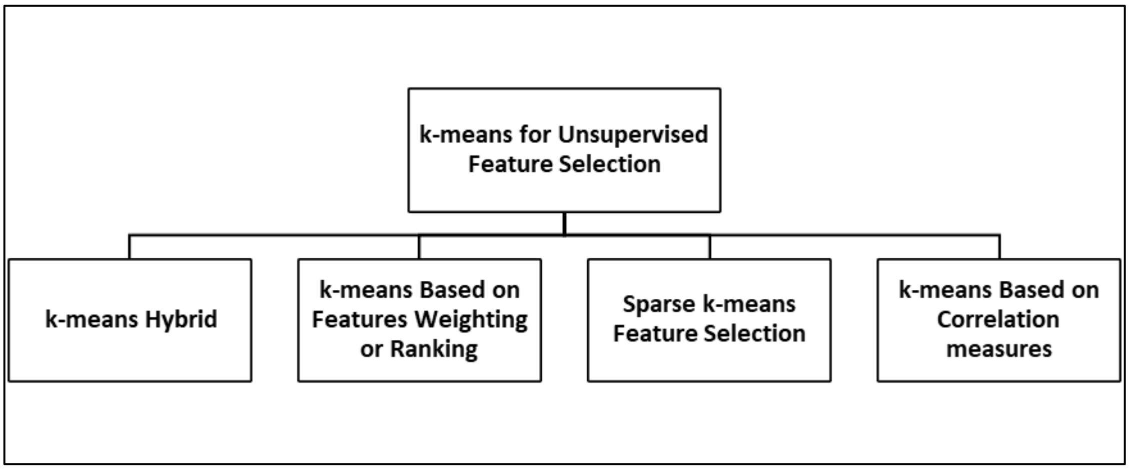

k-means for Unsupervised Feature Selection
k-means Hybrid
k-means Based on Features Weighting or Ranking
Sparse k-means Feature Selection
k-means Based on Correlation measures

Figure 1. k-means for Unsupervised Feature Selection Taxonomy

4. Recursive k-Means Silhouette Elimination (RkSE): Method Overview

Recursive k-means Silhouette Elimination (RkSE): is a dimensionality reduction technique for high dimensional data of various types such as, large time-series datasets, microarrays, text, images and so on. The idea behind RkSE method is similar to any ordinary cluster-based unsupervised feature selection method, where they treat features as objects or samples, and it is required to cluster them into groups based on a computed similarity measure, or with the aid of data mining by applying a suitable clustering method. RkSE keeps recursively applying k-means clustering to group the features with similar patterns in the same cluster. While, applying silhouette criteria iteratively as the selection condition.

RkSE is unique compared to other k-means and silhouette measure algorithms for feature selection in two main criteria: (1) Applying k-means in a recursive elimination style which is a unique selection technique for cluster-based feature selection approaches that normally follow forward selection or backward elimination. (2) The implication of a user-defined threshold that gives the user the freedom in controlling the selection based on the accuracy-complexity trade off.

Start the feature selection with collecting the features that are higher than some user-defined threshold or tolerance value. This threshold represents the strength of the connection between the cluster and the individual features located within, represented by the silhouette measure. The highest selected thresholds, the more connected the feature should be, more likely to be selected, and the more iterations required to complete the feature selection process.

Thereafter, the features with the highest silhouette criteria of each cluster is selected to represent the whole cluster. Within each cluster, neglect all the features higher than the threshold other than the selected highest silhouette feature. Since, the feature with highest silhouette value in the cluster is the one connected the most to this cluster, and the rest of the features within the cluster are either highly connected to the cluster centre (The ones with higher silhouette criteria than the threshold) or weakly connected to the centre (The ones with lower silhouette criteria than the threshold). The highly connected features are similar to each other, hence they are all strongly connected to the same cluster centroid, and by selecting only one of them, particularly the highest silhouette above the threshold, to represent the high pack is only fair and necessary to eliminate the redundant cluster similar features. However, the weakly connected features within the cluster (silhouette lower than selected threshold) are following slightly to highly different patterns than their connected clusters, and these connections can be affiliated to other cluster(s) or other centroids within the same cluster. That is why, these features should be accumulated from all the clusters and stored in a matrix for remaining features. Followed by aggregating them, re-cluster them all together and compute the silhouette over again.

This process keeps repeating recursively between clustering (dividing), silhouette criteria calculation, feature selection (highest silhouette above threshold of each cluster), elimination (silhouette above threshold of each cluster other than the highest) and aggregation (lower than threshold of each cluster) until all features are either selected or eliminated. In other words, the recursion is convergence when the amount features in the remaining features matrix is empty or null.

Assume that $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{d}\times \boldsymbol{N}}$ where $\boldsymbol{d}$ is the number of features or dimensions needed to be clustered, and $\boldsymbol{N}$ is the value of each feature $\boldsymbol{d}$ through the samples or selected subset of the samples. Set the Threshold $\boldsymbol{\sigma}$ to any desired percentage where $\boldsymbol{0} < \boldsymbol{\sigma} < \boldsymbol{1}$. The higher the threshold, the more features to be selected, and the number of iterations or re-clustering before reaching convergence is increased. Moreover, the quality of the feature selection is directly proportional to the threshold $\boldsymbol{\sigma}$ selected. When $\boldsymbol{\sigma} \rightarrow \boldsymbol{1}$ the max number of features $< \boldsymbol{N}$ are selected, and the accuracy of the feature selection is maximized. However, the computational cost and time will rise dramatically in comparison to lower thresholds, due to the increase of the iteration count for the process repetition.

It is crucial to identify some matrices required during the feature selection. $\mathbf{X}_{\mathbf{remain}} \in \mathbb{R}^{\boldsymbol{r}\times \boldsymbol{N}}$ where $\boldsymbol{r}$ is the remain features from past iterations that has not yet been eliminated or selected but require re-clustering to make the choice accordingly. **X_remain** contains the features from all the cluster aggregated, which did not satisfy the condition $\boldsymbol{S_i} \geq \boldsymbol{\sigma}$ in the previous iteration, as well as they showed weak connection to their current cluster, so re-clustering is inevitable to find another more connected pattern in the feature space.

$\mathbf{X}_{\mathbf{selected}} \in \mathbb{R}^{\boldsymbol{s}\times \boldsymbol{N}}$ where $\boldsymbol{s}$ is the number of features selected. The selected features are only the features with the highest silhouette criteria $\boldsymbol{max}\ (\boldsymbol{S_i})|_{\boldsymbol{S_i \epsilon C_k}}$ that is also fulfilling the selection criteria $\boldsymbol{S_i} \geq \boldsymbol{\sigma}$ within each cluster $\boldsymbol{C_k}$ collected recursively throughout the iterations after the aggregation and re-clustering of each phase. Which make the final condition for choosing the feature is $(\mathbf{max}(\boldsymbol{S_i}) \geq \boldsymbol{\sigma})|_{\boldsymbol{S_i \epsilon C_k}}$.

Finally, $\mathbf{X}_{\mathbf{eliminated}} \in \mathbb{R}^{\boldsymbol{e}\times \boldsymbol{N}}$ where $\boldsymbol{e}$ is the number of features eliminated that has the size of $\boldsymbol{e} < \boldsymbol{d}$. The features added to the $\mathbf{X}_{\mathbf{eliminated}}$ matrix are the redundant ones within each cluster collected iteratively throughout the iterations. More specifically, the eliminated features represent the ones that did belong to their representing cluster following that exact iteration. However, they have higher than threshold silhouette value $(\boldsymbol{S_i} \geq \boldsymbol{\sigma})|_{\boldsymbol{S_i \epsilon C_k}}$, but not high enough to represent the whole similar features in the cluster. Which means $((\boldsymbol{S_i} \geq \boldsymbol{\sigma}) \cap (\boldsymbol{S_i} < \mathbf{max}(\boldsymbol{S_i})))|_{\boldsymbol{S_i \epsilon C_k}}$. Eliminating those features even though they possess high intra-cluster relation can massively reduce the features redundancy. Furthermore, another reason to escape the algorithm is when it refuses to reach convergence for a pre-defined number of iterations, where $\mathbf{X}_{\mathbf{remain}}$ keeps constant and fixed for many iterations and no more possible re-clustering that provides the sufficient requested threshold is possible. In this case, all the features in $\mathbf{X}_{\mathbf{remain}}$ will be added to $\mathbf{X}_{\mathbf{eliminated}}$, which ensures $\mathbf{X}_{\mathbf{remain}}$ to have null content, that provokes the completion of the algorithm by reaching convergence.

To develop more precise explanation of the feature selection proposed, the below pseudo code to RkSE is introduced.

RkSE Pseudo Code

```
1.  Initialization of important matrices and parameters:
    $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{d}\times \boldsymbol{N}}$
    $\boldsymbol{\sigma} = \boldsymbol{user\_defined}\ \ \boldsymbol{0} < \boldsymbol{\sigma} < \boldsymbol{1}$
    $\mathbf{X}_{\mathbf{remain}} = \boldsymbol{X}$
    $\mathbf{X}_{\mathbf{eliminated}} = \emptyset$
    $\mathbf{X}_{\mathbf{selected}} = \emptyset$
2.  Apply k-means clustering using $\mathbf{X}_{\mathbf{remain}}$
3.  Calculate $\boldsymbol{S_i}$ for each element within each cluster following
    the equation below:
    $\boldsymbol{S_i} = \dfrac{\boldsymbol{b_i} - \boldsymbol{a_i}}{\mathbf{max}(\boldsymbol{a_i}, \boldsymbol{b_i})}$
4.  Some features will be selected, eliminated, or remain for re-
    clustering based on the following value of $\boldsymbol{S_i}$ within each
    cluster $\boldsymbol{C_k}$ separately.
```

```
$S_i = \begin{cases} \mathbf{X}_{\text{remain}} \leftarrow i, & if\ (S_i < \sigma) \\ \mathbf{X}_{\text{eliminated}} \leftarrow i, & if\ (S_i \geq \sigma) \cap (S_i < \max S_i) \\ \mathbf{X}_{\text{selected}} \leftarrow i, & if\ (S_i \geq \sigma) \cap (S_i = \max S_i) \end{cases}$
5.  Remove $\mathbf{X}_{\text{eliminated}}$ and $\mathbf{X}_{\text{selected}}$ from $\mathbf{X}_{\text{remain}}$
6.  Check if $\mathbf{X}_{\text{remain}}$ is empty
if $\mathbf{X}_{\text{remain}} = \emptyset$ then
Convergence achieved; feature selection is complete.
  Selected features are stored in $\mathbf{X}_{\text{selected}}$
else
Repeat from step 2
```

5. Analysis and Experimental Results

In this section, RkSE method is analysed, evaluated, and tested when applied on two main experiments.

The first experiment aims to study the effect of RkSE in feature selection for univariate time-series data in a shape of windows with a defined length. RkSE when applied to experiment one, is supposed to act as a time-series compression method that chooses the most informative time points in each sliding window and eliminates the redundant and less important time points per window (selection of optimal time points in one modality type).

The second experiment explores the potential of RkSE applied to a multivariate time-series dataset without the sliding window application. In this experiment, RkSE is expected to choose the most informative features within all the time points. i.e. in a dataset of different sensors' readings, RkSE is expected to choose the best sensors as the representative features (selection of best modalities among variations of them during various time frames).

For the sake of validating the results of the two mentioned experiments, it is essential to explain the main methodologies followed to validate unsupervised feature selection methods. The following points describe the main categories for unsupervised feature selection validation techniques as researched in [21].

- Feature selection evaluation by classification accuracy: in this method the selected features using the feature selection method subject of evaluation are used to test a classification problem using one of the common supervised classifiers such as, Support Vector Machines (SVM), K Nearest Neighbor (KNN) or Naïve Bayesian (NB). Then, the classification accuracy or the error rate is measured, and compared to the classification results of the entire dataset prior to classification.
- Feature selection evaluation using clustering criteria: In this case the results of a clustering task such as, k-means clustering is evaluated by using one of the clustering qualities measures. i.e. normalized mutual information and clustering accuracy.

In this work, the evaluation method used is the classification accuracy approach since the dataset available for evaluation has already included the labels. Therefore, it will provide a wide range of comparisons between RkSE using various classification methods besides the original dataset prior to feature selection.

The dataset used for the following experiments is the hydraulic test rig dataset available in [38]. This dataset represents real life measurements of multivariate time-series sensor data of six pressure sensors PS1-PS6, four temperature sensors TS1-TS4 and one vibration sensor VS1. It is also used for classification in a previous work, and described in full detail in [39]. In addition, it was pre-processed differently to fit two scenarios of classification schemas; one to fulfil sensor feature selection within a sliding window schema, while the other is processed for multiple sensors classification using the real-time measured faults in the test rig.

5.1. Experiment One: RKSE for Univariate Time-series Feature Selection within a Window

In this experiment, the sensor PS1 reading from the hydraulic test rig dataset is used for the purpose of sensor fault classification using a window classification schema. Four main types of faults

doi:10.20944/preprints202008.0254.v1

are injected in the PS1 data such as, constant fault (constant high, low and zero), gain fault and bias or offset fault, which makes the resultant PS1 data containing four different labels of faults along with the healthy label. The pre-processed PS1 data that has 28,882 readings that are captured in a second basis, and reshaped into a 7210 sliding windows with 60 seconds length worth of PS1 readings with zero intersection points between each window or offset/delay equals the window size n, as shown below in Figure 2:

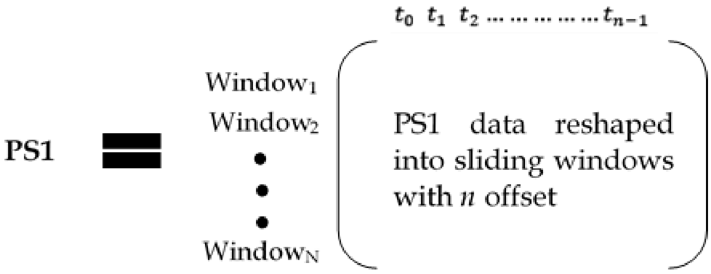

t_0 t_1 t_2 … … … … … t_{n-1}
PS1
$=$
Window$_1$
Window$_2$
Window$_N$
PS1 data reshaped into sliding windows with n offset

Figure 2 PS1 Data Reshape and Preprocessing Framework.

First, as described earlier RkSE has a user-specified threshold that effects the number of features selected and the accuracy of the selection process. Thus, the following table shows the number of features selected, the execution time for RkSE and the number of re-clustering iterations required until reached convergence when changing the threshold between 0.1 until approaching the highest threshold of one. As shown in table 1, the increase of the threshold selected increases the number of features selected, as well as increasing the time and computational complexity of the algorithm by increasing the number of iterations required until reaching the convergence criteria by emptying $\mathbf{X}_{\mathbf{remain}}$. The funnel chart below, Figure 3 emphasizes the relationship inferred above. Notice that when the threshold is 0.98≈ 1 all features in $\boldsymbol{X}$ were selected as if no feature selection is applied in the first place.

Table 1. The effect of Threshold on number of features selected, number of iterations required and execution time in milliseconds.

σ	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	0.95	0.98
Features Selected	3	5	8	8	8	9	10	11	20	46	60
Exe.Time (msec)	76.2	76.99	77.58	77.22	77.18	77.07	77.00	77.45	78.03	80.69	87.14
Iterations	2	2	3	3	3	4	4	7	8	8	10

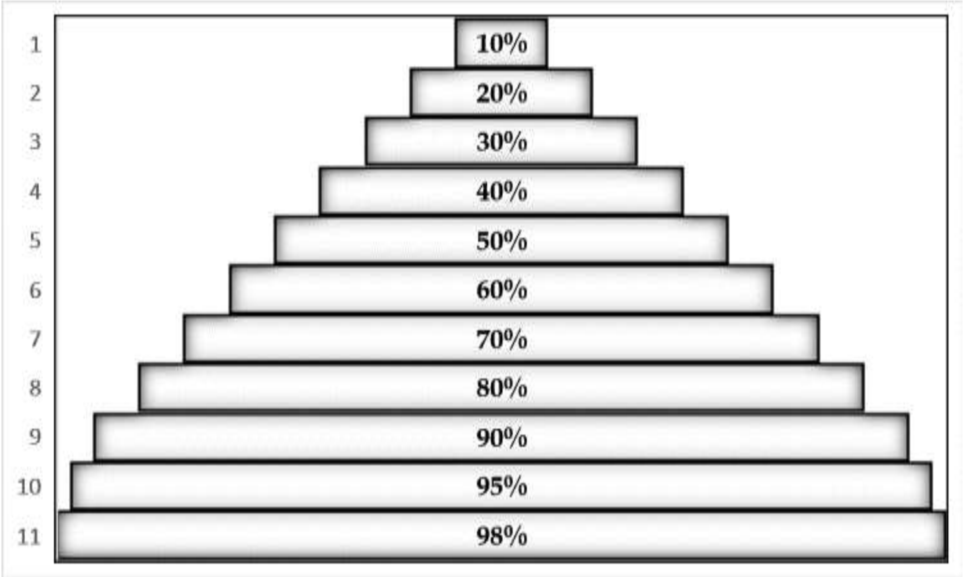


Figure 3. Funnel Graph describing the directly proportional relationship between the threshold and the features selected.

To evaluate the performance of the feature selection algorithm applied on the PS1 data, the features selected by RkSE are used to classify the data into healthy and numerous faulty states. The performance of RkSE is then compared to the original dataset without feature selection (number of features is 60).

Figure 4, shows the mean accuracy of 10-folds when applying 10-fold cross validation technique evaluating the performance of RkSE over various classifiers; Logistic Regression (LR), Linear Discriminant Analysis (LDA), KNN, Decision Tree (CART), NB, SVM and Random Forests (RF). Illustrated in figure 4, the results of the mean 10-fold accuracy for the classifiers applying different number of selected features to show a trade-off between the accuracy and the number of features selected when RkSE is applied.

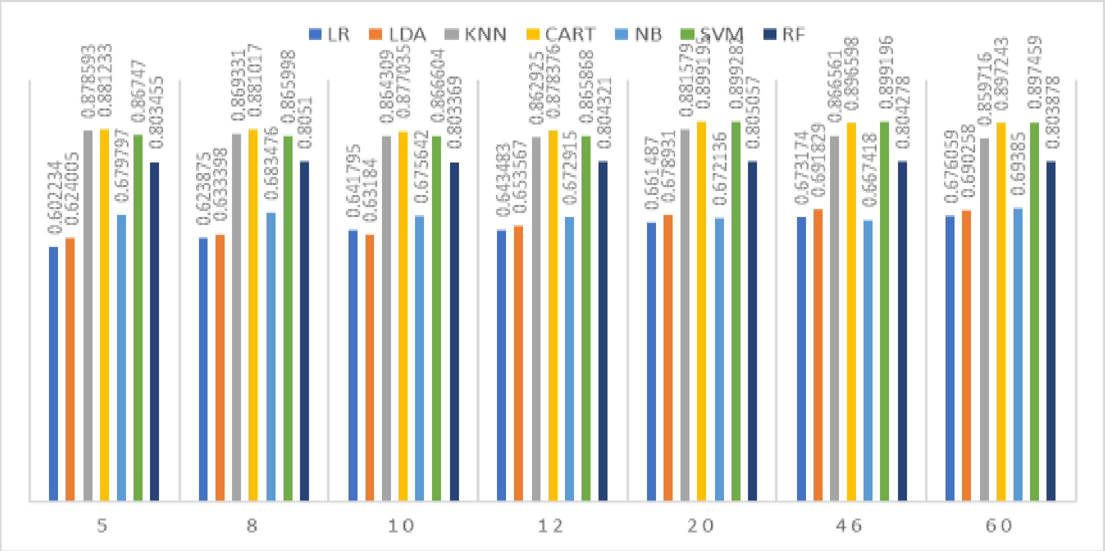


Figure 4. Feature Number and Mean Accuracy Comparisons Applied for Various Classifiers.

As shown, the increase in the number of features selected followed by an increase of classification accuracy for all the features regardless the classification method applied. However, when the threshold is 95% the feature selected are 46 features out of 60 overalls. The 46 features offer higher classification accuracy than the original dataset with less computational complexity. Which implies that RkSE has successfully reduced the dimensionality of a univariate time-series dataset in a sliding window scenario with even an increase of the mean accuracy for most of the classifiers applied. Figure 5 shows the relationship between the classifier accuracy and the threshold applied.

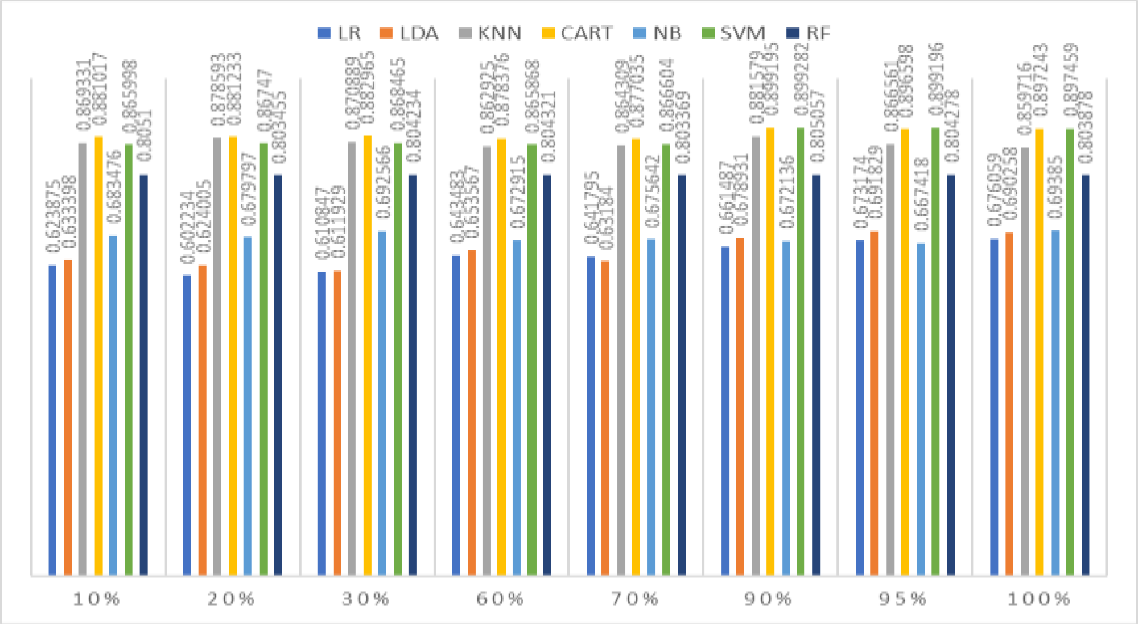
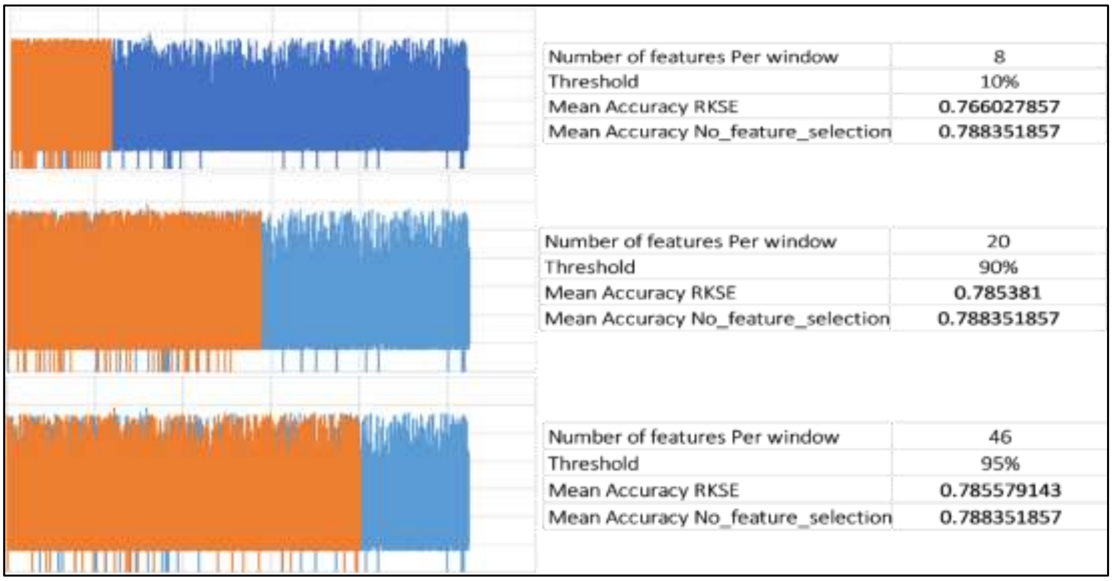


Figure 5. Threshold and Mean Accuracy Comparisons Applied for Various Classifiers.

Figure 6 demonstrates the ability of RkSE in reducing the size of the training time-series data, by selecting small number of features per window instead of taking all the window size for classification. The orange signal is the PS1 after feature selection, while the blue signal is the original PS1 signal without feature selection laying underneath the orange signal. As a conclusion, RkSE noticeably reduced the size of the training dataset without compromising the accuracy of the classification even when the smallest threshold is applied.



Number of features Per window	8
Threshold	10%
Mean Accuracy RKSE	**0.766027857**
Mean Accuracy No_feature_selection	**0.788351857**

Number of features Per window	20
Threshold	90%
Mean Accuracy RKSE	**0.785381**
Mean Accuracy No_feature_selection	**0.788351857**

Number of features Per window	46
Threshold	95%
Mean Accuracy RKSE	**0.785579143**
Mean Accuracy No_feature_selection	**0.788351857**

Figure 6. The effect of RKSE in minimizing the size of the original signal while keeping the accuracy.

5.2. Experiment Two: RkSE for Multivariate Time-series Feature Selection without a Window

In this experiment, the hydraulic test rig dataset is used for component fault classification based on the classification of eleven main sensors: PS1-PS6, TS1-TS4 and VS1. The data is processed in a way that include the fully efficient samples as the healthy form, while the full failure of the cooler, valve, pump, and heater are used to represent the rest of the faulty samples. For a detailed explanation of the dataset and the data-preprocessing involved in this work, refer to [39].

The overall goal of this experiment is to investigate the potential of RkSE for selecting the most important sensors for the component fault classification challenge. When applying RkSE to this experiment, it is crucial to make sure that the features (sensors) are located on the row of the dataset as they are the subject to be clustered iteratively.

When applying RkSE to the multivariate hydraulic test rig dataset, it showed a good performance comparing to when applying the entire eleven-dimensional dataset for classification. When the threshold is set to 0.20 the number of features selected are four, with 0.90 threshold five features are selected, 0.95 with six features, and finally with 0.98 threshold nine features out of eleven are selected. Figure 7, shows the results of all the classifiers applying different thresholds when using RkSE. Hence, different number of features are compared

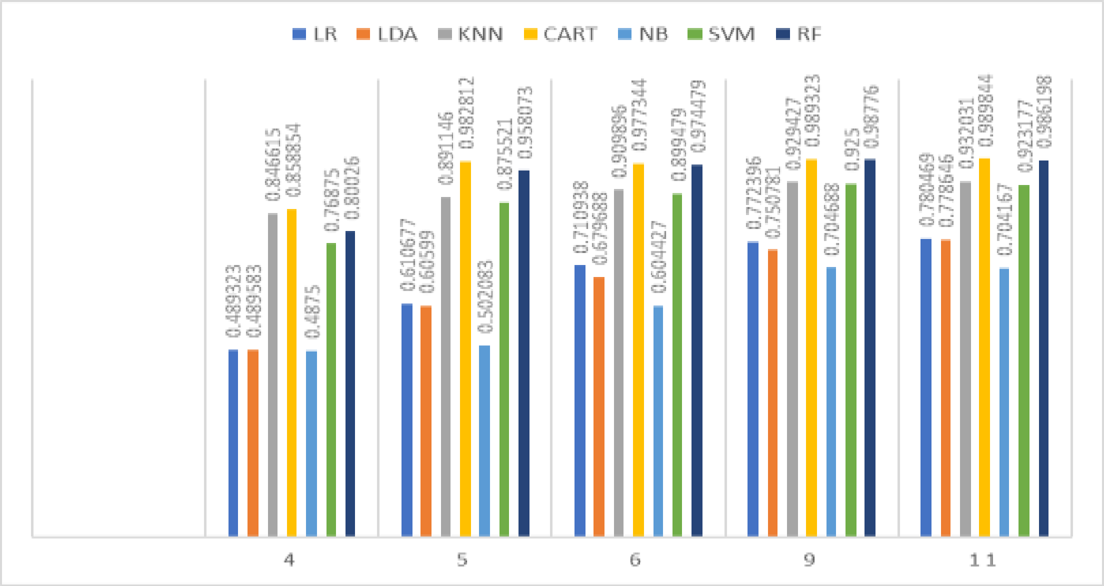


Figure 7. RkSE of Various Threshold Values Applied to Different Classifiers.

Figure 8, shows the average accuracy of all the classifiers mean accuracy at a certain number of features used for classification. It is obviously noticeable that RkSE of thresholds 0.90 and 0.95 with six and nine number of selected features respectively, has shown comparable results to the fully sized dataset of eleven features with lower dimensionality applied.

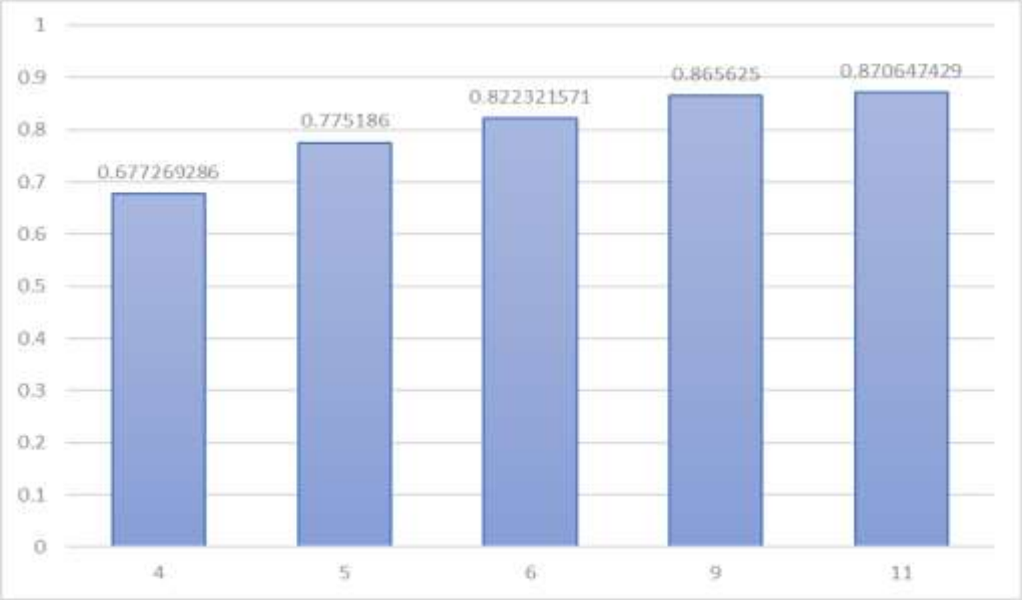


Figure 8. Average Accuracy of All Classifiers for Different Feature Numbers.

4. Discussion and Future Work

In this work Recursive k-means Silhouette Elimination (RkSE) is proposed, tested and validated. RkSE is a new unsupervised feature selection algorithm with a novel idea of applying k-means clustering and silhouette measure beyond its ordinary backward or forward selection style, but as a recursive acquisition approach with the application of a user-defined threshold, that plays a major role in the uniqueness of this approach.

RkSE can be applied for various applications. However, in this work RkSE is applied to reduce dimensionality in univariate and multivariate sensor time-series datasets. For future work, RkSE will be applied on numerous datasets of different applications and fields such as, genetics, microarrays, text, images and so on. Furthermore, this work focuses on the performance of RkSE on univariate and multi-variate time-series datasets. i.e. sensor data.

RkSE is evaluated on a real measured data of a hydraulic test rig and validated to solve a classical fault classification problem in two different experiments: (1) RkSE is used in a univariate time-series dataset, to classify sensor faults in a sliding window format. When RkSE is applied on sliding windows, its function indicates the ability to select the best quality time points to represent the whole window. In other words, RkSE can successfully be used as a signal compression method when, applied to time-series data in a window format. (2) In a multi-variate time-series data without the application of sliding windows, RkSE is applied to select the best modality of features to represent the whole dataset.

In addition, this work structures a comprehensive review survey of feature selection algorithms based on k-means clustering, existing in the literature. Which yielded the creation of a new taxonomy for k-means clustering feature selection methods.

The results of the two extensive experiments proves the uniqueness and efficiency of RkSE method for time-series datasets in a univariate or multi-variate format.

To sum up, RkSE represents an iterative, unsupervised, silhouette-based, k-means clustering feature selection algorithm. Although, RkSE has plenty of advantaged and contributions that exceed the methods mentioned in related-work section, it also has limitations that we hope to eliminate in the future. The following points show the strengths and limitations of RkSE method for feature selection based on its functionality and workflow.

RkSE advantages are the following:

- Can Create a model representation of feature dependencies (Iterative algorithm advantage).
- Feature selection and clustering are made concurrently in one single operation. (Iterative Advantage)
- Unsupervised Feature Selection method, no labels required.
- Simple, robust, and low computational and time costs: Due to the exclusive application of k-means and silhouette criteria, which provides simplicity and reduce time and computational complexity
- User-interactive: allows the user to choose the value of the threshold which give the freedom to select the best number of features which provide the option to loosen the accuracy or the complexity constrains.
- Introducing a new concept of using k-means and silhouette measure in a recursive manner, instead of the common forward and backward approaches.

The limitations of RkSE are as follows:

- Prone to overfitting. (Iterative algorithms disadvantage)
- The choice of the threshold if not chosen probably can drastically affect the quality of the selection, which cannot be guaranteed since σ is user selected.
- The accuracy is a little compromised because the algorithm focuses on the relationship between the feature and the cluster, rather than the relationships between features.

Author Contributions: Conceptualization, A.M. and M.F.; methodology, A.M.; software, A.M.; validation, M.F.; formal analysis, A.M.; investigation, A.M.; resources, M.F.; data curation, A.M.; writing—original draft preparation, A.M.; writing—review and editing, M.F.; visualization, A.M.; supervision, M.F.; project administration, M.F.; funding acquisition, M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the DFG research grants LO748/11-1 and OB384/5-1.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. John, G. H.; Kohavi, R.; Pfleger, K. Irrelevant Features and the Subset Selection Problem. In Proceedings of the Eleventh International Conference on International Conference on Machine Learning; ICML'94; Morgan Kaufmann Publishers Inc.: New Brunswick, NJ, USA, 1994; pp 121–129.
2. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 2003, 3 (null), 1157–1182.
3. Dash, M.; Liu, H. Feature Selection for Classification. Intelligent Data Analysis 1997, 1 (3), 131–156. https://doi.org/10.3233/IDA-1997-1302.
4. Miao, J.; Niu, L. A Survey on Feature Selection. Procedia Computer Science 2016, 91, 919–926. https://doi.org/10.1016/j.procs.2016.07.111.
5. Ang, J. C.; Mirzal, A.; Haron, H.; Hamed, H. N. A. Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2016, 13 (5), 971–989. https://doi.org/10.1109/TCBB.2015.2478454.
6. Dy, J. G.; Brodley, C. E. Feature Selection for Unsupervised Learning. J. Mach. Learn. Res. 2004, 5, 845–889.
7. Sechidis, K.; Spyromitros-Xioufis, E.; Vlahavas, I. Information Theoretic Multi-Target Feature Selection via Output Space Quantization. Entropy 2019, 21 (9), 855. https://doi.org/10.3390/e21090855.
8. Wang, G.; Guan, G. Weighted Mean Squared Deviation Feature Screening for Binary Features. Entropy 2020, 22 (3), 335. https://doi.org/10.3390/e22030335.
9. Lim, H.; Kim, D.-W. Generalized Term Similarity for Feature Selection in Text Classification Using Quadratic Programming. Entropy 2020, 22 (4), 395. https://doi.org/10.3390/e22040395.
10. Fröhlich, H.; Chapelle, O.; Schölkopf, B. Feature Selection for Support Vector Machines Using Genetic Algorithms. Int. J. Artif. Intell. Tools 2004, 13 (04), 791–800. https://doi.org/10.1142/S0218213004001818.
11. Lin, S.-W.; Ying, K.-C.; Lee, C.-Y.; Lee, Z.-J. An Intelligent Algorithm with Feature Selection and Decision Rules Applied to Anomaly Intrusion Detection. Applied Soft Computing 2012, 12 (10), 3285–3290. https://doi.org/10.1016/j.asoc.2012.05.004.
12. Bhattacharyya, D. K.; Kalita, J. K. Network Anomaly Detection: A Machine Learning Perspective; Chapman & Hall/CRC, 2013.
13. Zhou, Y.; Kang, J.; Zhang, X. A Cooperative Coevolutionary Approach to Discretization-Based Feature Selection for High-Dimensional Data. Entropy 2020, 22 (6), 613. https://doi.org/10.3390/e22060613.
14. Arabie, P.; Hubert, L. J. An Overview of Combinatorial Data Analysis. In Clustering and Classification; WORLD SCIENTIFIC, 1996; pp 5–63. https://doi.org/10.1142/9789812832153_0002.
15. Kasim, S.; Deris, S.; Othman, R. M. Multi-Stage Filtering for Improving Confidence Level and Determining Dominant Clusters in Clustering Algorithms of Gene Expression Data. Comput. Biol. Med. 2013, 43 (9), 1120–1133. https://doi.org/10.1016/j.compbiomed.2013.05.011.
16. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations; The Regents of the University of California, 1967.
17. Yuan, C.; Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. J — Multidisciplinary Scientific Journal 2019, 2 (2), 226–235. https://doi.org/10.3390/j2020016.
18. Rousseeuw, P. J. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics 1987, 20, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7.
19. Sheikhpour, R.; Sarram, M. A.; Gharaghani, S.; Chahooki, M. A. Z. A Survey on Semi-Supervised Feature Selection Methods. Pattern Recognition 2017, 64, 141–158. https://doi.org/10.1016/j.patcog.2016.11.003.
20. Cai, J.; Luo, J.; Wang, S.; Yang, S. Feature Selection in Machine Learning: A New Perspective. Neurocomputing 2018, 300, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077.
21. Solorio-Fernández, S.; Carrasco-Ochoa, J. A.; Martínez-Trinidad, J. Fco. A Review of Unsupervised Feature Selection Methods. Artif Intell Rev 2020, 53 (2), 907–948. https://doi.org/10.1007/s10462-019-09682-y.
22. Alelyani, S.; Tang, J.; Liu, H. Feature Selection for Clustering: A Review. In Data Clustering: Algorithms and Applications; 2013. https://doi.org/10.1201/9781315373515-2.

23. Dash, M.; Liu, H. Feature Selection for Clustering. In Knowledge Discovery and Data Mining. Current Issues and New Applications; Terano, T., Liu, H., Chen, A. L. P., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2000; pp 110–121. https://doi.org/10.1007/3-540-45571-X_13.
24. Hruschka, E. R.; Hruschka, E. R.; Covoes, T. F.; Ebecken, N. F. F. 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); 2005; p 6 pp.-. https://doi.org/10.1109/ICHIS.2005.42.
25. Kim, Y.; Street, W. N.; Menczer, F. Evolutionary Model Selection in Unsupervised Learning. Intell. Data Anal. 2002, 6 (6), 531–556.
26. Breaban, M.; Luchian, H. A Unifying Criterion for Unsupervised Clustering and Feature Selection. Pattern Recognition 2011, 44 (4), 854–865. https://doi.org/10.1016/j.patcog.2010.10.006.
27. Davies, D. L.; Bouldin, D. W. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1979, PAMI-1 (2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909.
28. Hruschka, E. R.; Covoes, T. F. 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; Vol. 1, pp 32–38. https://doi.org/10.1109/CIMCA.2005.1631238.
29. Jing, L.; Ng, M. K.; Huang, J. Z. An Entropy Weighting K-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. IEEE Transactions on Knowledge and Data Engineering 2007, 19 (8), 1026–1041. https://doi.org/10.1109/TKDE.2007.1048.
30. Sahu, B.; Dehuri, S.; Jagadev, A. K. Feature Selection Model Based on Clustering and Ranking in Pipeline for Microarray Data. Informatics in Medicine Unlocked 2017, 9, 107–122. https://doi.org/10.1016/j.imu.2017.07.004.
31. Modha, D. S.; Spangler, W. S. Feature Weighting in K-Means Clustering. Machine Learning 2003, 52 (3), 217–237. https://doi.org/10.1023/A:1024016609528.
32. Huang, J. Z.; Ng, M. K.; Rong, H.; Li, Z. Automated Variable Weighting in K-Means Type Clustering. IEEE Trans Pattern Anal Mach Intell 2005, 27 (5), 657–668. https://doi.org/10.1109/TPAMI.2005.95.
33. Chormunge, S.; Jena, S. Correlation Based Feature Selection with Clustering for High Dimensional Data. Journal of Electrical Systems and Information Technology 2018, 5 (3), 542–549. https://doi.org/10.1016/j.jesit.2017.06.004.
34. Wosiak, A.; Zakrzewska, D. Integrating Correlation-Based Feature Selection and Clustering for Improved Cardiovascular Disease Diagnosis https://www.hindawi.com/journals/complexity/2018/2520706/ (accessed Jun 12, 2020). https://doi.org/10.1155/2018/2520706.
35. Witten, D. M.; Tibshirani, R. A Framework for Feature Selection in Clustering. J Am Stat Assoc 2010, 105 (490), 713–726. https://doi.org/10.1198/jasa.2010.tm09415.
36. Wang, S.; Tang, J.; Liu, H. Embedded Unsupervised Feature Selection. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence; AAAI'15; AAAI Press: Austin, Texas, 2015; pp 470–476.
37. Guo, J.; Guo, Y.; Kong, X.; He, R. Unsupervised Feature Selection with Ordinal Locality. In 2017 IEEE International Conference on Multimedia and Expo (ICME); 2017; pp 1213–1218. https://doi.org/10.1109/ICME.2017.8019357.
38. UCI Machine Learning Repository: Citation Policy https://archive.ics.uci.edu/ml/citation_policy.html (accessed Feb 4, 2020).
39. Mallak, A.; Fathi, M. A Hybrid Approach: Dynamic Diagnostic Rules for Sensor Systems in Industry 4.0 Generated by Online Hyperparameter Tuned Random Forest. Sci 2020, 2 (3), 61. https://doi.org/10.3390/sci2030061.