





Article

PFA-Nipals: An Unsupervised Principal Feature Selection Based on Nonlinear Estimation by Iterative Partial Least Squares

Emilio Castillo-Ibarra ¹, Marco A. Alsina ², Cesar A. Astudillo ³ and Ignacio Fuenzalida-Henríquez ^{4,*}

¹ Engineering Systems Doctoral Program, Faculty of Engineering, Universidad de Talca, Campus Curicó, Curicó 3340000, Chile; emcastillo@utalca.cl

² Faculty of Engineering, Architecture and Design, Universidad San Sebastian, Bellavista 7, Santiago 8420524, Chile; marco.alsina@uss.cl

³ Department of Computer Science, Faculty of Engineering, University of Talca, Campus Curicó, Curicó 3340000, Chile; castudillo@utalca.cl

⁴ Building Management and Engineering Department, Faculty of Engineering, University of Talca, Campus Curicó, Curicó 3340000, Chile

* Correspondence: ifuenzalida@utalca.cl

Abstract: Unsupervised feature selection (UFS) has received great interest in various areas of research that require dimensionality reduction, including machine learning, data mining, and statistical analysis. However, UFS algorithms are known to perform poorly on datasets with missing data, exhibiting a significant computational load and learning bias. In this work, we propose a novel and robust UFS method, designated PFA-Nipals, that works with missing data without the need for deletion or imputation. This is achieved by considering an iterative nonlinear estimation of principal components by partial least squares, while the relevant features are selected through minibatch K-means clustering. The proposed method is successfully applied to select the relevant features of a robust health dataset with missing data, outperforming other UFS methods in terms of computational load and learning bias. Furthermore, the proposed method is capable of finding a consistent set of relevant features without biasing the explained variability, even under increasing missing data. Finally, it is expected that the proposed method could be used in several areas, such as machine learning and big data with applications in different areas of the medical and engineering sciences.

Keywords: unsupervised feature selection; Nipals; clustering; missing data; interpretability

MSC: 62H30; 68T10



Citation: Castillo-Ibarra, E.; Alsina, M.A.; Astudillo, C.A.; Fuenzalida-Henríquez, I. PFA-Nipals: An Unsupervised Principal Feature Selection Based on Nonlinear Estimation by Iterative Partial Least Squares. *Mathematics* **2023**, *11*, 4154. <https://doi.org/10.3390/math11194154>

Academic Editors: Xunlin Zhu and Lijun Pei

Received: 12 August 2023

Revised: 15 September 2023

Accepted: 17 September 2023

Published: 3 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Dimensionality reduction refers to the transformation of a high-dimensionality dataset into another dataset of lower dimensionality that preserves most of the information content [1,2]. When the transformation maps the original dataset into a transformed space, the corresponding reduction in dimensionality is known as feature extraction. On the other hand, when the transformation seeks to select a subset of relevant features from the original dataset, the reduction is known as feature selection [3].

Algorithms for feature extraction are particularly attractive for machine learning, since the derived features can be constructed to maximize information content while avoiding redundancy, therefore improving the performance of the learning step [3]. For instance, a principal components analysis (PCA) constructs a set of linearly transformed features that are orthogonal and ranked based on the explained variability of the original dataset [3].

There are several instances where feature selection is preferred over feature extraction: the interpretability of the features to understand the relationship between different variables; computational efficiency, where feature selection allows faster computing times; to

reduce overfitting when the datasets used are small; domain knowledge of some important features and allowing them to be selected first; finally, when a high correlation between features is observed, feature selection can identify the most relevant ones and therefore reduce dimensionality.

Therefore, feature selection is an important preprocessing step for machine learning applications, since it removes features that are either redundant or irrelevant, reducing the computational load, it enhances the interpretation of results, and it allows the analysis of datasets that would be computationally unmanageable otherwise [1,4–6]. Methods for feature selection can be categorized as supervised or unsupervised, depending on whether or not a learning algorithm is available to provide feedback on the selection of relevant features [1].

In particular, unsupervised feature selection (UFS) algorithms have received considerable attention in research areas where datasets with thousands of features are present, such as pattern recognition, machine learning, data mining, and statistical analysis [7].

Since unsupervised data in the real world may contain irrelevant or redundant features that can affect the analysis and lead to biases or incorrect results [1], it becomes crucial to eliminate unnecessary features to improve computational efficiency and enhance the robustness of simple models on small datasets due to their lower variation in sample details [1,2]. By explaining the data with fewer features, we gain a deeper understanding of the underlying process, enabling the extraction of valuable knowledge [2]. This becomes especially relevant when seeking to understand the behavior of specific systems or phenomena [8].

In the context of missing data, the problem can be categorized into three types: coverage errors, total nonresponse, and partial nonresponse [9]. Addressing coverage errors is crucial during sample selection, while total nonresponse requires attention during the data collection phase. On the other hand, partial nonresponse can be effectively handled during the analysis phase.

Different strategies have been proposed to deal with missing data, including imputation and deletion. For imputation, the following techniques can be considered:

- Consideration of central tendency metrics (e.g., mean, mode or median) [10].
- Regression techniques [11].
- K-nearest neighbors algorithms [12].
- Expectation–maximization algorithms [13].
- Neural networks [12].

Although imputation methods preserve the original dataset size, they introduce bias in the estimators and often require the fulfillment of certain assumptions for the data distribution. Regarding deletion, the elimination from the dataset of an entire observation that presents missing data (i.e., listwise deletion) is frequently used. Issues associated with deletion include a reduction in the total variability, the elimination of possibly important observations, and a decrease in the power of statistical significance tests [12,14].

This study proposes a novel UFS method based on an iterative nonlinear estimation of components by partial least squares, in which feature weights are calculated for each component and clustered considering a minibatch approach in order to reduce calculation time. These minibatches are subsets of the weights of each feature with respect to the components, finally delivering the cluster of similar features, and selecting the representative of each cluster. Our feature selection method offers a low computational load while preserving most of the original variance, working with missing data without the need for imputation or deletion.

This paper is organized as follows: the remaining portion of the introduction briefly summarizes the state of the art of UFS methods, specifically those based on a principal component analysis; Section 2 details the proposed method, while Section 3 describes the data and the experimental methodology. Subsequently, Section 4 presents and discusses the obtained results, while Section 5 presents the overall discussion and conclusions of this work.

Survey of UFS Methods

In the literature survey regarding the UFS methods, the terms of feature selection and feature extraction for unsupervised features are sometimes understood as the same. However, they are distinct: feature selection refers to the selection of a subset of features from the original space, while the feature extraction technique selects features in a transformed space, which do not have a clear physical meaning [3].

Principal component analysis (PCA) is frequently used as a feature extraction algorithm since it forms linear combinations of all available features. However, these features contemplated by each PCA does not have equal importance in the formation of PCs necessarily, since some features may be critical and others irrelevant or insignificant to the overall analysis [3].

Some authors linked PCs to a subset of the original features by selecting critical variables or eliminating redundant, irrelevant, or insignificant variables. In this sense, B2 (backward elimination) and B4 (forward binding) algorithms are probably the best-known approaches of this type [15]. The B2 algorithm discards variables that are highly associated with the last PCs, while B4 algorithm selects variables that are highly associated with the first PCs.

As shown in [3], fundamental problems are solved: the first is based on finding a UFS algorithm with low computational time and the second is based on giving interpretability to the PCA. For this, the author relies on the evaluation of features for their ability to reproduce the projections on the main axes, by using regression through ordinary least squares (OLS) and by selecting forward features and eliminating backward features.

On the other hand, a new hybrid approach to PC-based unsupervised gene selection is developed in [16], which is divided into two steps: the first retrieves subsets of genes with an original physical significance based on their capabilities to reproduce sample projections into principal components by applying the evaluation through an OLS estimation, and the second step looks for the best subsets of genes that maximize the performance of the cluster. In [7], a moving range threshold is proposed, which is based on quality control and control charts to better identify the significant variables of the dataset.

On the other hand, obtaining the eigenvectors to identify critical original features based on the nearest-neighbor rule to find the predominant features is proposed in [17].

In [18], a method called principal feature analysis (PFA) is proposed. This method uses PCs to calculate the eigenvalues and eigenvectors, which are grouped using k-means and return the feature closest to the center.

Later, a convex scattered principal component algorithm (CSPCA) is proposed in [19], applied to feature learning. It is shown that PCA can be formulated as a robust low-rank regression optimization problem for outliers. They also mention that the importance of each feature can be analyzed effectively according to the PCA criteria, since the new objective function is convex, and they propose an iterative algorithm to optimize it, generating weights for each original feature.

On the other hand, an improved and unsupervised difference algorithm called sparse difference embedding (SDE) to reduce the dimensionality of data with high numbers of features and few observations was developed in [20]. SDE seeks to find a set of projections that not only affects the locality of the intraclass and maximizes the globality of the interclass but also simultaneously applies the lasso regression to obtain a sparse transformation matrix.

An algorithm coupling a PCA regularization with a sparse feature selection model to preserve the maximum variance of the data is proposed in [21], where an iterative minimization optimization is designed in order to optimize an objective function. In this way, it looks for a significant adjustment result and the selection of informative features, respectively.

On the other hand, another type of feature selection algorithm, called the globally sparse probabilistic PCA (GSPPCA) was generated in [22], which is based on a Bayesian procedure that allows the obtention of several sparse components with the same scatter pattern to identify relevant features. This selection of features by GSPPCA is achieved

using Roweis's probabilistic interpretation in PCA and isotropic Gaussian functions in the load matrix, providing calculations of the marginal probability of a Bayesian PCA model.

However, other types of unsupervised feature selection methods exist, based on the usage of filters and not necessarily by means of a principal component analysis. In [23], unsupervised feature selection methods based on filters received more attention due to their efficiency, scalability, and simplicity. Therefore, the most recent articles are presented as follows:

In [24], a method called FSFS (feature selection using feature similarity) is proposed, with the primary objective of reducing the redundancy between the characteristics of a dataset by measuring the dependency or statistical similarity between them using the variance and covariance of the features to be later grouped in clusters. Features with similar properties are located in the same cluster. Finally, the feature selection is performed iteratively using the kNN principle (k-nearest neighbors). At each iteration (one for each feature), FSFS selects one feature of each cluster to create a final subset of features.

In [25], a method called MCFS (multicluster feature selection) is proposed, with the objective of selecting subsets of features from a dataset, preserving the overall structure of the clustered data. MCFS is an algorithm that uses spectral analysis and regression with the L_1 norm to measure the importance of the features, considering the local data structure to later select a subset of features based on the regression coefficients. The final objective is to maximize the preservation of the internal data structure of the selected subset of features.

In [26], a method called UDFS (unsupervised discriminative feature selection) is proposed, with the objective of selecting a subset of features that discriminates among the data to perform regression and classification tasks. The algorithm uses discriminant information from the sparse matrices and correlates the features to assign weights to each characteristic and select the most relevant ones. This facilitates the overall efficiency and precision of the automatic learning models by reducing the number of features while maintaining the most informative attributes.

In [27], a method called NDFS (nonnegative discriminative feature selection) is presented, which is a technique for feature selection combining different steps. First, it uses spectral analysis to learn pseudoclass labels, representing the relations between the data and their features. Then, a regression model with a norm $L_{2,1}$ regularization is performed to be optimized using a specific solver. Finally, p features are selected that relate the best with the previously learned pseudoclass labels. This process allows the authors to identify a set of relevant and discriminant features to solve classification problems.

In [28], a method called DSRMR (dual self-representation and manifold regularization) is proposed. It is a learning algorithm based on a representation of the features and has the objective of selecting an optimal set of relevant features for a learning problem. DSRMR achieves this by considering three fundamental aspects: a representation of the features using an $L_{2,1}$ norm, the local geometrical structure of the original data, and a weighted representation of each feature based on its relative importance. The optimization is performed efficiently using an iterative algorithm and the final results are based on the resulting weighted features.

In [29], an innovative method for feature selection is proposed based on the sample correlations and dependencies between features. This model is composed of two main elements: to preserve the global and local structure of the dataset and to consider the global and local information. This is achieved by using mutual information and learning techniques, such as dynamical graphs and low-dimensionality restrictions to obtain reliable information and a correlation of the feature samples.

Finally, it is worth mentioning that in [30], a method was proposed using two convolutional neural networks (CNN) to extract the principal features of different datasets, obtaining excellent results in comparison to other state-of-the-art methods. This method was tested using IoT devices to study signals and identify them as malicious or normal.

All these algorithms mentioned above propose to take PCs as the central core to select relevant features, either as a means of feature selection or as a means of interpretability

for PCs. However, none of them solve the problem of missing data, having to perform a preprocessing of the datasets to correct them.

2. Description of the Proposed Method: PFA-Nipals

This study introduces a new approach to feature selection, named principal feature analysis based on nonlinear iterative partial least squares or PFA-Nipals. This algorithm was developed to select attributes or features considering unsupervised data using filters, allowing the identification of the most significant variables that also explain the most sources for variability among the data. This approach is especially valuable in that it can adapt effectively to situations where missing data are observed without the need to perform an imputation of them. This algorithm is considered with a filter approach, selecting the most relevant features by examining the data without algorithms performing a particular search.

The algorithm works by selecting variables based on the desired number of features k , and creates k groups that bring together variables that share similar behavior patterns or variability within a transformed mathematical space by means of orthogonal projections. Then, the most representative variable is selected for each group.

The algorithm can be described in three large steps:

1. Dataset rank and orthogonal projection transformation.
2. Groups of features with equal variance.
3. Selection of the representative variables for each group.

2.1. Dataset Rank and Orthogonal Projection Transformation

Let $X_{n,d}$ be a set of data with n observations and d dimensions of rank a . The value of a is used to identify a eigenvalues different from zero in the dataset. To calculate the rank, the average ranking algorithm is used.

When the method is applied to datasets with missing data, the missing values are considered as separate elements, and a special rank is usually considered and typically denoted as NaN . These missing data are treated as values that do not interact with the rank assignments to the rest of the nonmissing elements.

Therefore, for the $X_{n,d}$ dataset, the orthogonal projection based on the principal component analysis is used, described in [31,32] and is presented as follows:

Fundamentally, a singular value decomposition (SVD) of a data matrix is carried out by means of iterative sequences of orthogonal projections [32], using a centered and reduced dataset, defined as $X = x_{i,j}$ with rank a , n corresponds to the number of observations and p the number of features. Hence, the decomposition of the matrix X can be described as follows:

$$X = \sum_{h=1}^a t_h p_h' \tag{1}$$

where $t_h = (t_{h1}, \dots, t_{hm})'$ and $p_h = (p_{h1}, \dots, p_{hp})'$ are the principal components and vectors, respectively, for each component number h_i and the superscript $'$ means a transposition of the corresponding vector.

It is important to mention that p_{hj} represents the regression coefficient $X_{h-1,j}$ before the normalization, over the component t_h . Then, each variable can be written using a principal component analysis, deriving the following approximation:

$$x_j \approx \sum_{h=1}^a t_h p_{h,j}' + Residual, \quad \forall j = 1 \dots p \tag{2}$$

Once convergence is achieved, i.e., the residual is less than a given tolerance, the preceding matrix is deflated to ensure the orthogonality of each of the components.

If there are missing data, in the same way, the components t_h and the vectors P_h are obtained, which then allows us to reconstruct the matrix X . This is because PFA-Nipals works with a series of scalar products as the sum of products of the elements paired. This

allows us to handle missing data, adding available pairs in each operation. Geometrically, the missing elements are taken as if they adjust properly on the regression line [32].

For this stage of the method, the following steps are considered:

1. Residue calculation:

$$X_{h-1,j} = X_j - \sum_{l=1}^{h-1} p_{lj}t_l.$$

2. Nipals initialization:

$$t_h = X_{h-1,1}.$$

3. Regression of $X_{h-1,j}$ in t_h
where p_{hj} is the slope of the line of the least squares that intersects the origin of $(t_h, X_{h-1,j})$, calculated using the available data.

$$p_{hj} = \frac{\sum_i x_{h-1,ji}t_{hi}}{\sum_i t_{hi}^2} \text{ for all available (nonmissing) } x_{ji} \text{ and } t_{hi}.$$

4. Residue calculation:

$$X_{h-1,i} = X_i - \sum_{l=1}^{h-1} t_{li}p_l.$$

5. Regression of $X_{h-1,i}$ in p_h
where p_{hj} is the slope of the least square line that intersect the origin of $(t_h, X_{h-1,j})$, calculated using the available data.

$$t_{hi} = \frac{\sum_j x_{h-1,ji}p_{hj}}{\sum_j p_{hj}^2} \text{ for all available (nonmissing) } x_{ji}$$

6. Repeat steps 3 to steps 5 until the variation of p_h between iterations is equal to or less than 0.1%.

2.2. Cluster of Features of Equal Variance

The direction vectors $p_{d,a}$ are considered as input, and the variables that have similar variability are clustered in k groups, where k is the final number of features to be selected. The clustering is performed using the minibatch K-means algorithm.

The minibatch K-means algorithm is based on massive data clustering [33]. The minibatch K-means algorithm uses small batches of samples taken at random with a fixed size and for each iteration of the loop, until the algorithm converges, small randomly selected batches are taken from the dataset and used to update the clusters, assigning one cluster to each data point in the batch, taking into account the previous location of the cluster centroids. Each minibatch is then processed to update the clusters using a combination of the sample and prototype values, considering the previous locations of the cluster centroids. Therefore, as the number of iterations increases, the new examples are taken into consideration until the algorithm convergence is achieved when there are no changes in the clusters. The great advantage of this algorithm is that it subsamples the data at each iteration, in contrast to what the K-means algorithm does, thus reducing computational costs [34].

In order to help convergence, consistency of results, and computational time, an initialization “ c ” of the minibatch K-means algorithm is proposed, which takes as its initial value the k features that present the greatest weight for each component. In other words, a feature is selected for each component, until reaching k components. In this way, the most representative features for each component are initially selected, reducing the same variability explained between variables by the concept of the orthogonality of the components. The procedure can be described as follows:

1. Initialization of the feature cluster.

$$T_0 = p_h$$

where T_0 is defined as the direction vectors p_h with d attributes and a components. Then, the number of features to select k is identified, and $k < d$ attributes are selected.

2. Cluster initialization
Initialization of each k cluster to analyze.

$$t_i = \max |t_{d,i}| \quad \forall i = 1, \dots, k$$

If $k \leq a$, k features are chosen (one for each component) for the first k components, without repeating selected features.

If $k > a$, a features are chosen, one for each of the a components, without repeating the selected features, and the sequence of steps are repeated for the remaining $k - a$ components.

3. Clustering of features: A variation of the K-means algorithm is used, that considers minibatches to reduce computational time while optimizing the objective function, called the minibatch K-means algorithm. The minimization function is defined as follows:

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|)^2$$

where μ_j is the mean or centroid of the cluster.

In Algorithm 1, the pseudocode is described, which outlines the initialization operation.

Algorithm 1 Minibatch K-means initialization

Input: $k, T_{n,a}$: k : number of features to select; $1 \leq k \leq d$. $T_{n,a}$: weight matrix of the features with respect to each component.

- 1: **Start initialization criterion**
 - 2: $T = []$, initialization set
 - 3: **for** $i = 1, 2, 3, \dots, k$: **do**
 - 4: Take $T_{n,i}$.
 - 5: $t_i = \max(|T_{1,i}|, |T_{2,i}|, |T_{3,i}|, |T_{4,i}|, \dots, |T_{n,i}|)$
 - 6: Identify which x_i feature corresponds to the selected t_i
 - 7: $T = [x_1, x_2, x_3, x_4, x_5, \dots, x_k]$
 - 8: **return** T , initialization list
-

2.3. Selection of Each Group Representative

The centers of each group are calculated and the closer feature to the center is selected. Therefore, it can be considered that the feature closest to the center is the central dominant and less redundant variable of the group, compared to the characteristics of the other groups. This stage can be described as follows:

- Calculation of the center of each k group.
- For each group, the distance between the center of the group and each feature is compared.
- The most representative feature of each group is selected by finding the minimum distance with respect to the center, using the following expression:

$$\min(d(a_{d,k} - c_k)) \quad \forall k = 1, 2, \dots, k$$

where $d(a_{d,k} - c_k)$ is the Euclidean distance of $a_{d,k}$, for the d feature in group k , and c_k is the group k center.

- Output: return the k feature more representative of each group.

In Algorithm 2, the operation of PFA-Nipals is described.

Algorithm 2 PFA-Nipals

Input: k : Number of features to select; $1 \leq k \leq d$ y $X_{n,d}$, a data matrix of n observations and d dimensions centered and reduced, $X_{i,j} \in \mathcal{R}$ y $X_{n,d}$ of rank a

- 1: **Start eigenvectors**
- 2: $X_0 = X$
- 3: **for** $h = 1, 2, 3, \dots, a$: **do**
- 4: t_h first column of X_{h-1}
- 5: Start of convergence of p_h
- 6: $p_h = X'_{h-1} \frac{t_h}{t'_h t_h}$
- 7: Normalize $p - h$ a 1
- 8: $t_h = X_{h-1} \frac{p_h}{p'_h p_h}$
- 9: **if** there is convergence of p_h **then**
- 10: Go to step 13
- 11: **else**
- 12: Go back to step 5
- 13: $X_h = X_{h-1} - t_h p'_h$
- 14: Extract the p_h
- 15: **Start minibatch K-means**
- 16: $T = p_h$, where $T_{d,a}$, a data matrix of d features and a components, $T_{d,a} \in \mathcal{R}$
- 17: k clusters
- 18: For each c initialized $\in C$, t is taken according to the initialization of T .
- 19: $v \leftarrow 0$
- 20: **for** $i = 0$ **to** it **do**
- 21: $M \leftarrow mb$
- 22: **for** $t \in M$ **do**
- 23: $d[t] \leftarrow f(C, t)$. The center closest to t should be cached
- 24: **for** $t \in M$ **do**
- 25: $c \leftarrow d[t]$. (The cached center for t is obtained)
- 26: $v[c] \leftarrow v[c] + 1$. (Updating of the counts that exist per center)
- 27: $\eta \leftarrow \frac{1}{v[c]}$. (Learning rate per center)
- 28: $c \leftarrow (1 - \eta)c + \eta t$. (Gradient calculation)
- 29: Obtain k clusters.
- 30: Calculate the Euclidean distance between the center of the cluster with respect to all the features of the cluster.
- 31: Select $\min(|attribute_{i,k} - centroid_k|), \forall k = 1, 2, 3, \dots, k$.
- 32: **return** k selected features

3. Materials and Methods

Two experiments were performed: one based on synthetic data which shows the qualities of the selection of features through PFA-Nipals reflected in different indexes; the second is based on selecting features through different algorithms mentioned in the literature on different datasets and comparing the results with clustering metrics with respect to our algorithm.

3.1. Unsupervised Learning Problem with Synthetic Dataset

For the first experiment, a synthetic problem was used that considered two principal features of 3 Gaussian clusters as shown in Figure 1. In total, 600 observations were generated and divided into 3 clusters with 200 observations each. The features to be clustered were defined as $x_1 \sim N(\mu = 0, \sigma = 1)$, $x_1 \sim N(\mu = 4, \sigma = 1)$ and $x_1 \sim N(\mu = 5, \sigma = 1)$ for the x_1 feature and $x_2 \sim N(\mu = 0, \sigma = 1)$, $x_2 \sim N(\mu = 4, \sigma = 1)$ and $x_2 \sim N(\mu = -1, \sigma = 1)$ for the x_2 feature, where μ and σ correspond to the population mean and standard deviation, respectively.

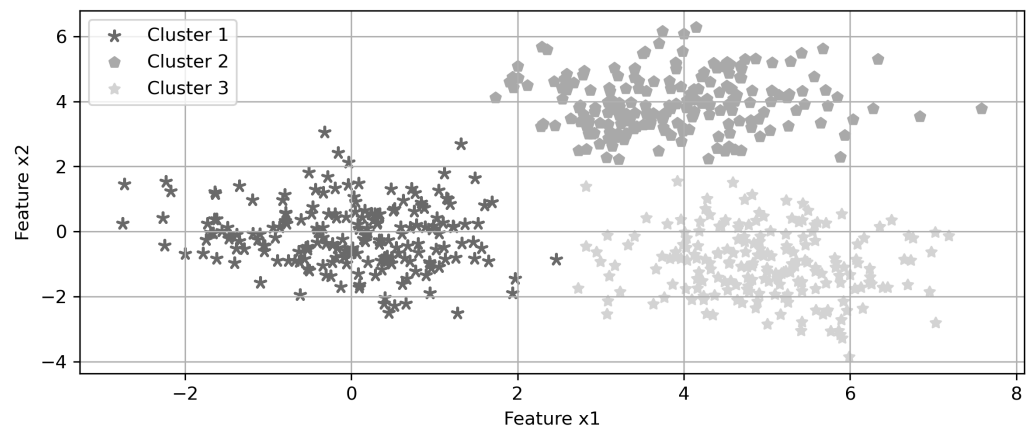


Figure 1. Three clusters with a Gaussian data distribution.

Then, ten white noise features were applied, which were generated as $a_i \sim N(\mu = 0, \sigma = 1), \forall i = 1, 2, \dots, 10$. With these twelve (principal plus noise) features created, the synthetic dataset was generated.

Subsequently, the selection of features was made through PFA-Nipals, with which from 1 to 10 features were selected. Through the K-means algorithm, the 3 clusters mentioned above were calculated.

In order to evaluate the results of the clusters found by the K-means algorithm, two performance measures were calculated. The first one was the homogeneity metric of the labels of the clusters, which states that a clustering result satisfies the homogeneity condition, if all its clusters contain only data points that are members of a single class and its mathematical formulation [35] is:

$$h = 1 - \frac{H(C | K)}{H(C)} \tag{3}$$

where $H(C | K)$ is the conditional entropy of the classes given the cluster assignments and is given by:

$$H(C | K) = - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \cdot \log\left(\frac{n_{c,k}}{n_k}\right) \tag{4}$$

and $H(C)$ is the entropy of the classes and is given by:

$$H(C) = - \sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log\left(\frac{n_c}{n}\right) \tag{5}$$

with n the total number of samples, n_c and n_k the number of samples that belong to the class c and cluster k , respectively, and finally, $n_{c,k}$ the number of samples of class c assigned to cluster k .

The second performance measure was the normalized mutual information (NMI), which is a normalization of the mutual information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation), and its mathematical formulation is presented in [36]. In order to understand the NMI score, suppose two label assignments (of the same N objects), U and V . Their entropy is the amount of uncertainty for a set of partitions, defined by:

$$H(U) = - \sum_{i=1}^{|U|} P(i) \cdot \log(P(i)) \tag{6}$$

where $P(i) = \frac{|U_i|}{N}$ is the probability that a randomly selected object U falls into class U_i .

The previous equations are also applied to V , shown as follows:

$$H(V) = - \sum_{j=1}^{|V|} P'(j) \cdot \log(P'(j)) \tag{7}$$

where $P'(j) = \frac{|V_j|}{N}$.

The mutual information (MI) between U and V is calculated by:

$$MI(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \cdot \log\left(\frac{P(i, j)}{P(i) \cdot P'(j)}\right) \tag{8}$$

where $P(i, j) = \frac{|U_i \cap V_j|}{N}$ is the probability that a randomly selected object belongs to both classes U_i and V_j .

Finally, the normalized mutual information is defined as:

$$NMI(U, V) = \frac{MI(U, V)}{\text{mean}(H(U), H(V))} \tag{9}$$

To calculate these performance measures, a cross-validation was performed with 80% of the data for training and 20% of the data as validation, and the K-means algorithm was executed 30 times. Thus, a mean and a standard deviation were calculated for both metrics.

To better understand the methodology described for the synthetic problem, Figure 2 is presented below.

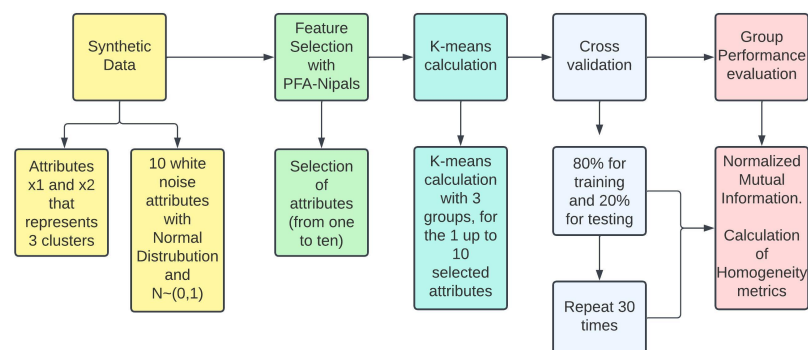


Figure 2. Methodology for unsupervised learning problem with synthetic dataset.

3.2. K-Means Clustering Problem through Unsupervised Feature Selection with Missing Data

In this section, we described how we evaluated the performance of the proposed algorithm for different datasets, comparing it with feature selectors as presented in [19] and described below:

To compare the performance of the proposed algorithm, 4 UFS methods were considered with the objective to perform an empirical comparison regarding the performance of the comparison methods with the quality of the solution obtained using PFA-Nipals. The algorithms were the following:

1. Laplacian score: It is selected to preserve the structure of local variability. The features consistent with the Laplacian matrix are selected, in which the importance of each feature is determined by its power [37].
2. SPEC: It is based on spectral regression models. The features are selected one by one, taking advantage of the work of spectral graph theory [38]
3. MCFS: It is based on spectral analysis and sparse regression. This algorithm specializes in the selected features better preserving the structure of multiple data clusters [25].

4. UDFS: Features are selected by a joint framework of discriminative analysis and $L_{2,1}$ -norm minimization. UDFS selects the most discriminative feature subset from the whole feature set in batch mode [26].

These algorithms were selected considering the following reasons:

- Different algorithms were chosen based on the performance using filter methods. Laplacian score, SPEC, MCFS, and UDFS are based on the category of spectral/sparse learning methods [1].
- The more relevant methods within the filter methods were considered, considering the quality of the selected features and the execution time [1,19].

3.2.1. Data

For this comparison, the datasets shown in Table 1 were considered, and an index called “ratio” was calculated, which is defined in the following equation:

$$ratio = \frac{obs}{dim} \quad (10)$$

where *obs* and *dim* correspond to the number of observations and dimensions, respectively.

This index allows the quantification of the ratio between the number of observations and the number of features, as presented in Table 1.

Table 1. Datasets used in this study. The table shows the name of the datasets, as well as the number of observations and dimensions, and the calculated ratio.

Dataset	Observations (<i>n</i>)	Dimensions (<i>d</i>)	Ratio = <i>n/d</i>
Arrhythmia	452	279	1.620
Lung Cancer	73	325	0.2246
Lymphoma	96	4026	0.0238
Colon Cancer	61	2000	0.0305
NCI	60	9712	0.0061
Soybean	307	35	8.7714
Leukemia	72	7070	0.01018
Epileptic Recognition	11,500	179	64.2458

The selected datasets have no missing data and were drawn from two sources. The first was from the microarray gene expression sets for the application of feature selection of maximum relevance and minimum redundancy [4] (<http://home.penglab.com/proj/mRMR/>, accessed on 1 April 2021), the second was obtained from the UCI Machine Learning Repository [39] (<https://archive.ics.uci.edu/ml/index.php>, accessed on 1 April 2021).

Since the selected datasets did not have missing data, these were artificially created for 0%, 1%, 2%, 3%, 4%, and 5% of the total observations of each dataset. These percentages were selected since the bias introduced by the missing data is proportional to the number of losses so that 10% or more is unacceptable and up to 5% is admissible [40].

3.2.2. Feature Selection

The feature selection was divided into two conditions. The first was based on algorithms that did not need to estimate missing data, which, in our case, was PFA-Nipals, and the second for algorithms that needed the estimation such as the Laplacian score, MCFS, and SPEC. For the first condition, the selection of features was done directly, without the need to estimate.

For the second condition, an estimate of missing data was performed by imputation to complete the missing data by replacement using the mean of each feature.

The number of features selected for each dataset was from 2 to 25 features for each of the missing data percentages. The number of clusters for each dataset is shown in Table 2.

Table 2. Number of selected features and number of clusters of the datasets.

Dataset	Number of Clusters	Number of Selected Features
Arrhythmia	3	
Colon	2	
Lung	2	
Lymphoma	3	[2, 3, 4, 5, 6, 7, 8, ..., 25]
NCI	2	
Soybean	3	
Leukemia	2	
Epileptic Recognition	5	

Herewith, it was expected that each algorithm would deliver the features selected for each dataset.

3.2.3. Cluster Analysis

For this analysis, the datasets without missing data were considered, since we wanted to analyze the effectiveness of the selector algorithms with missing data, but not that of some cluster algorithm. With the features selected by each algorithm, clusters were built for each dataset through the K-means algorithm.

The number of clusters for each dataset was established by performing a cluster silhouette analysis, choosing the number of clusters that gave the best silhouette coefficient [41]. These results can be observed in Table 2.

3.2.4. Validation and Performance Indexes

For the datasets under study, the true labels of the clusters were not known, so the evaluation was carried out using the cluster model itself. Therefore, two metrics were considered. The first was the silhouette coefficient, which generates limited scores between -1 (for incorrect clusters) and 1 (for correct clusters), where a higher score for the silhouette coefficient is related to a model with better-defined clusters and is established for each sample and composed of two scores defined for a sample as follows:

$$s = \frac{b - a}{\max(a, b)} \tag{11}$$

where a is the mean distance between a sample and all other points of the same class, and b corresponds to the mean distance between a sample and all other points in the next closest cluster.

The silhouette coefficient for a set of samples is given as the mean of the silhouette coefficient for each sample. The second metric is the Davies–Bouldin index [42], where a lower Davies–Bouldin index is related to a model with a better separation between clusters. The Davies–Bouldin index is defined as the average similarity of each cluster C_i for $i = 1, 2, \dots, k$ and its closest similarity C_j . In the context of this index, similarity is defined as a mean $R_{i,j}$.

$$R_{i,j} = \frac{s_i + s_j}{d_{i,j}} \tag{12}$$

where s_i is the mean distance between each point in the cluster i and the center of that cluster (also known as the cluster diameter), and $d_{i,j}$ is the distance between the cluster centroids i and j .

Therefore, the Davies–Bouldin index is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} R_{i,j} \tag{13}$$

where k corresponds to the maximum number of clusters.

To calculate these performance measures, a cross-validation was performed with 80% of the data for training and 20% of the data as validation, and the K-means algorithm was executed 30 times. Thus, the mean value and standard deviation of each of the mentioned indexes were calculated. To better understand what has been described, Figure 3 is shown below.

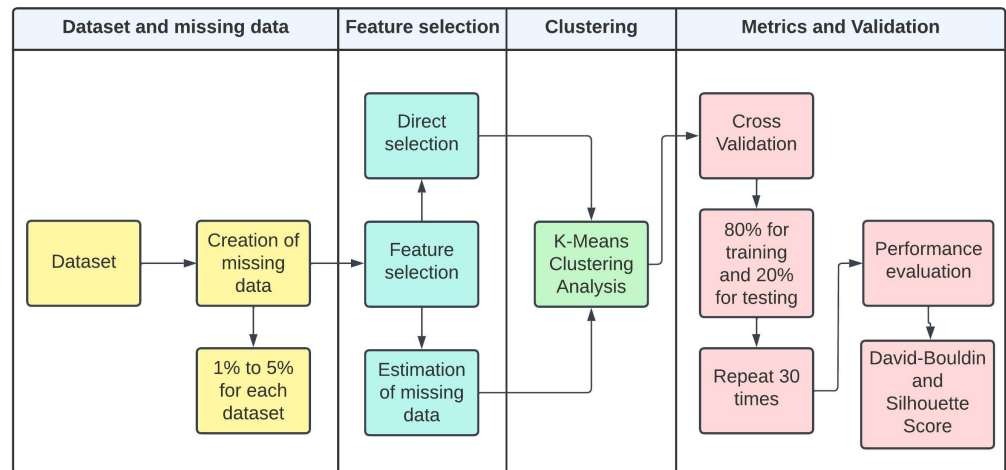


Figure 3. Methodology applied to the K-means clustering problem through the selection of unsupervised features with missing data.

4. Results and Discussion

The results obtained for each of the problems described in the previous section are shown below.

4.1. Results of Unsupervised Learning Problem with Synthetic Dataset

As mentioned in the methodology, the homogeneity metrics and normalized mutual information were calculated with the three clusters found by K-means clustering for each cluster of selected features.

Regarding Figure 4, we can observe that the number of optimal features to select is two. This is because the homogeneity metric and normalized mutual information for two features is one of the highest with a minimum standard deviation.

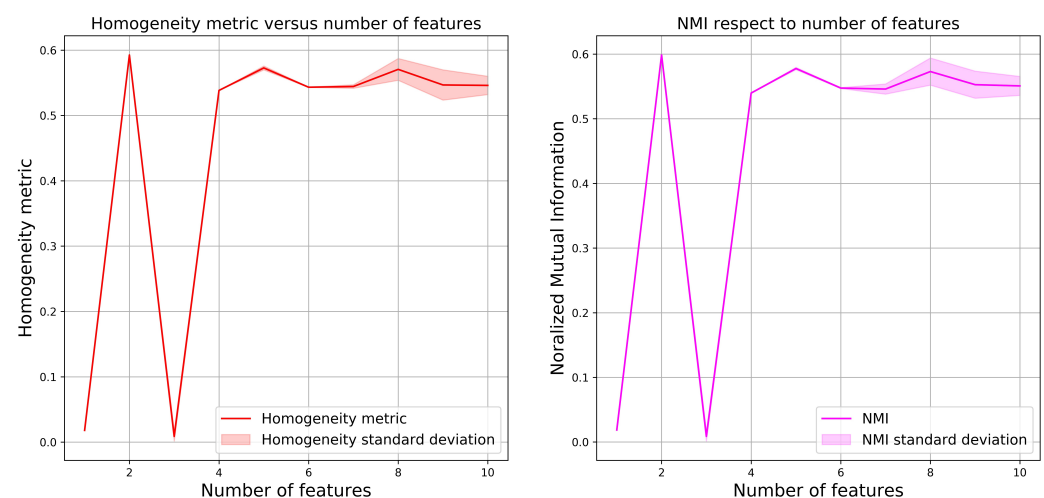


Figure 4. Metric of a cluster versus number of features selected for data synthetic.

With what was described above, we can mention that in each of the 30 repetitions, 480 observations were used as training and 120 observations as validation, assigning them to the closest cluster. The results, for both homogeneity metric and normalized mutual information, show that only two features are necessary for clustering and the two features are correct (two principal features).

4.2. Results of the K-Means Clustering Problem through the Selection of Unsupervised Features with Missing Data

In this section, a table is shown for each dataset, indicating the best performance of each algorithm with respect to the missing data analysis for each dataset under study. These performance values are shown regardless of the available number of features for a result. Along with the above, a graph was created, which reflected the evolution of each index with respect to the number of selected features, considering 5% of missing data.

The results were analyzed for each of the datasets and showed the following results:

4.2.1. Colon Cancer

For the Colon Cancer dataset, Table 3 shows that for most of the analysis with missing data, PFA-Nipals performed better on the silhouette coefficient than the other algorithms, except for 0% of missing data where the Laplacian score obtained a better result only for the silhouette coefficient, since for the Davies–Bouldin index, the best score was obtained by PFA-Nipals. Regarding the Davies–Bouldin index, PFA-Nipals performed better for all analyses with missing data.

Table 3. Best index results regarding the percentage of missing data. Dataset: Colon Cancer.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.5168	0.5136	0.5814	0.4890	0.3765	0.6475	0.7219	0.7060	0.8008	1.0716
1%	0.5975	0.5186	0.4026	0.4607	0.4396	0.5730	0.7257	0.9974	0.8250	0.9490
2%	0.5756	0.5211	0.0900	0.3843	0.5438	0.5760	0.7277	1.0345	0.9762	0.6673
3%	0.5466	0.5161	0.3177	0.3664	0.4426	0.6964	0.7238	1.1378	1.0047	0.8665
4%	0.5168	0.5157	0.3878	0.4651	0.5020	0.7198	0.7251	0.9745	0.7878	0.7241
5%	0.7126	0.5186	0.4680	0.5603	0.4734	0.3173	0.7257	0.8271	0.6245	0.7899

* Laplacian. Text in bold are the best index values.

Furthermore, the Colon Cancer dataset for the 5% missing data case in Figure 5 shows that the best results for all algorithms were found in the selection of the first 10 features. Along with this, it can be mentioned that for this dataset, PFA-Nipals obtained very good results with very few selected features.

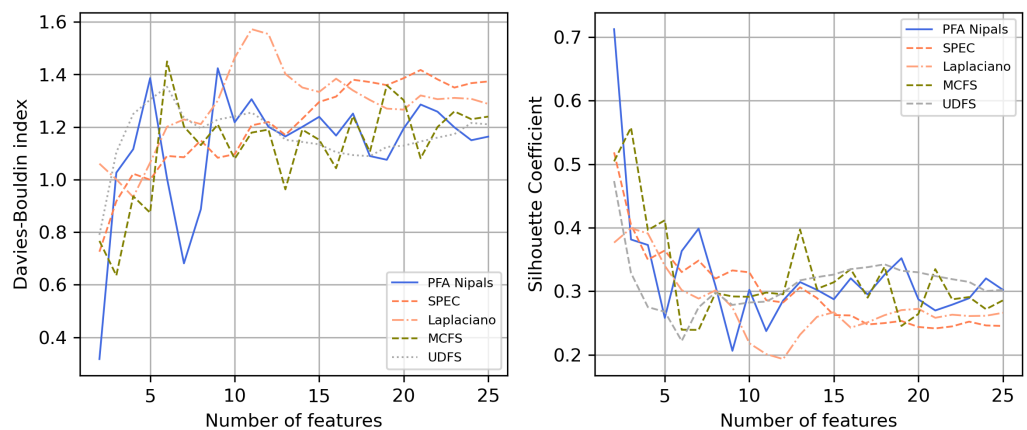


Figure 5. Evolution of indexes with respect to the number of selected features for 5% of missing data. Colon Cancer dataset.

For the Colon Cancer dataset, we can have indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

4.2.2. Lung Cancer

For the Lung Cancer dataset, Table 4 shows that PFA-Nipals performed better on the silhouette coefficients for all cases. Regarding the Davies–Bouldin index, PFA-Nipals also obtained the best results for all cases. Furthermore, for the 5% missing data case, Figure 6 again shows that the best results were found in the selection of the first 10 features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

Table 4. Best index results regarding the percentage of missing data. Dataset: Lung Cancer.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.6543	0.5308	0.4376	0.4373	0.5566	0.4862	0.6720	0.9019	0.9871	0.6616
1%	0.6543	0.3467	0.5230	0.4508	0.4066	0.4862	1.3523	0.7521	0.8116	0.9906
2%	0.6543	0.3998	0.4484	0.3953	0.4194	0.4862	1.0238	0.8971	0.9768	0.9595
3%	0.6713	0.4422	0.3209	0.3567	0.5321	0.4447	0.9180	1.2143	1.0036	0.7535
4%	0.5737	0.2977	0.3908	0.4353	0.5077	0.5304	1.6252	1.1407	0.9784	0.7993
5%	0.7430	0.5116	0.3697	0.4208	0.3877	0.3333	0.7076	1.1525	0.9186	1.0460

* Laplacian. Text in bold are the best index values.

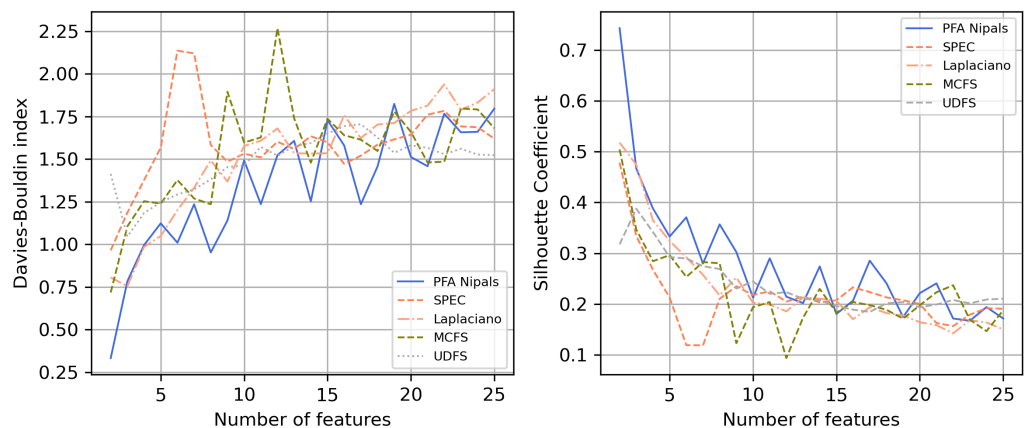


Figure 6. Evolution of indexes with respect to the number of features selected for 5% of missing data. Lung Cancer dataset.

For the Lung Cancer dataset, we can again mention that there are indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

4.2.3. Lymphoma

For the Lymphoma dataset, Table 5 shows that PFA-Nipals performed better on the silhouette coefficient for all cases of missing data. Regarding the Davies–Bouldin index, PFA-Nipals again obtained the best results for all cases of missing data.

In addition, for the 5% missing data case, Figure 7 shows again that the best results were found in the selection of the first 10 features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

For the Lymphoma dataset, we can have indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

Table 5. Best index results regarding the percentage of missing data. Dataset: Lymphoma.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	1.0000	0.4473	0.4245	0.5359	0.5454	0.0000	0.9064	0.9508	0.8127	0.8472
1%	1.0000	0.4473	0.4922	0.3524	0.5720	0.0000	0.8996	0.8256	0.7278	0.6824
2%	1.0000	0.4473	0.4598	0.4336	0.4660	0.0000	0.9030	0.8757	0.8668	0.8463
3%	1.0000	0.4465	0.4569	0.4100	0.4487	0.0000	0.9047	0.8459	0.8373	0.9340
4%	1.0000	0.4482	0.5158	0.4497	0.4237	0.0000	0.9217	0.8129	0.8165	0.8932
5%	1.0000	0.4465	0.4941	0.5326	0.5076	0.0000	0.9013	0.8862	0.7453	0.8348

* Laplacian. Text in bold are the best index values.

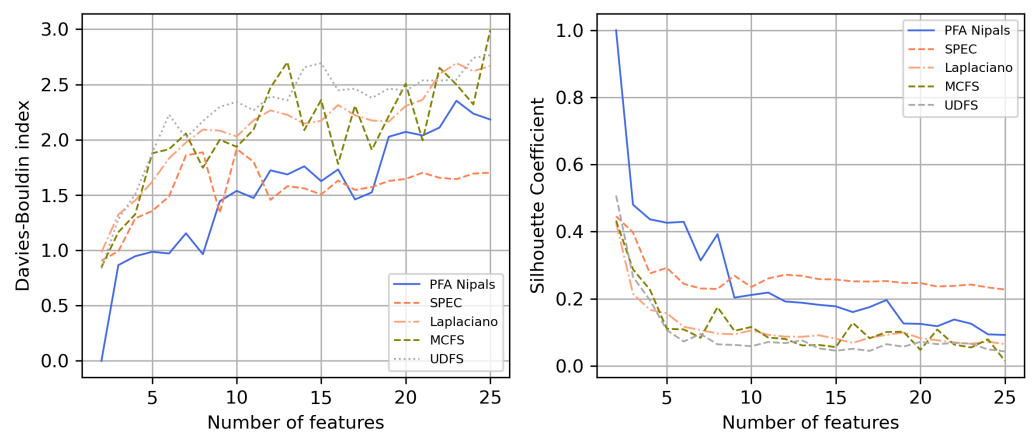


Figure 7. Evolution of indexes with respect to the number of features selected for 5% of missing data. Lymphoma dataset .

4.2.4. Arrhythmia

For the Arrhythmia dataset, Table 6 shows that PFA-Nipals performed better on silhouette coefficients for 2% and 3% of missing data. Regarding the Davies–Bouldin index, PFA-Nipals achieved the best results for the 2% and 3% of missing data cases.

Table 6. Best index results regarding the percentage of missing data. Dataset: Arrhythmia.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.9131	0.5117	----	0.9852	0.5695	0.5537	0.6354	----	0.0000	0.4751
1%	0.9110	0.5117	----	0.3438	0.9925	0.2460	0.6354	----	0.9666	0.0000
2%	1.0000	0.5117	----	0.9619	0.4312	0.0000	0.6354	----	0.2138	0.6578
3%	0.9909	0.5117	----	0.4912	0.3776	0.0098	0.6354	----	0.6325	0.8512
4%	0.5008	0.5117	----	0.6152	0.9923	0.5763	0.6355	----	0.4504	0.0168
5%	0.5008	0.5715	----	0.4801	0.5654	0.5763	0.5383	----	0.6094	0.5733

* Laplacian. Text in bold are the best index values.

Furthermore, for the 5% missing data case, Figure 8 shows that the best results were found in the selection of the first five features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

For the Arrhythmia dataset, we can have indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

It is important to mention that the Laplacian score was not calculated due to a nonconvergence of this parameter for the present dataset.

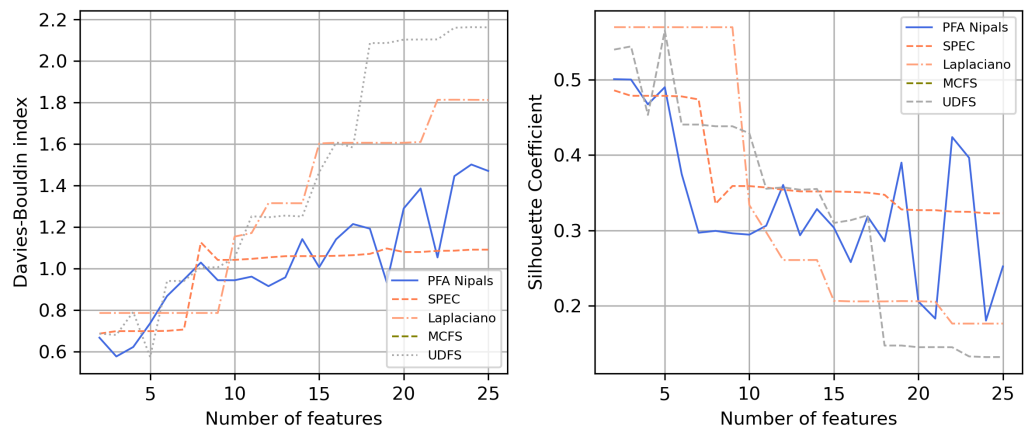


Figure 8. Evolution of indexes with respect to the number of selected features for 5% of missing data. Arrhythmia dataset.

4.2.5. NCI

For the NCI dataset, Table 7 shows that PFA-Nipals performed better on silhouette coefficients for 0%, 1%, 4%, and 5% of missing data. Regarding the Davies–Bouldin index, PFA-Nipals achieves the best results for the 0%, 1%, 4%, and 5% missing data cases.

Table 7. Best index results regarding the percentage of missing data. Dataset: NCI.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.6457	0.5264	0.5085	0.5410	0.5912	0.4205	0.6752	0.7912	0.7676	0.6243
1%	0.7064	0.5264	0.6914	0.3349	0.5229	0.3737	0.6752	0.5068	1.1221	0.7491
2%	0.5251	0.5264	0.3806	0.5924	0.5996	0.6051	0.6752	1.1095	0.5997	0.7221
3%	0.6168	0.3859	0.4291	0.5085	0.9051	0.5429	1.0486	0.8688	0.4573	0.1522
4%	0.6933	0.3859	0.5032	0.5766	0.5994	0.4840	1.0486	0.7490	0.6553	0.7221
5%	0.9052	0.6023	0.5318	0.6524	0.9051	0.2424	0.4561	0.8318	0.5473	0.2522

* Laplacian. Text in bold are the best index values.

In addition, for the 5% missing data case, Figure 9 shows that the best results were found in the selection of the first three features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

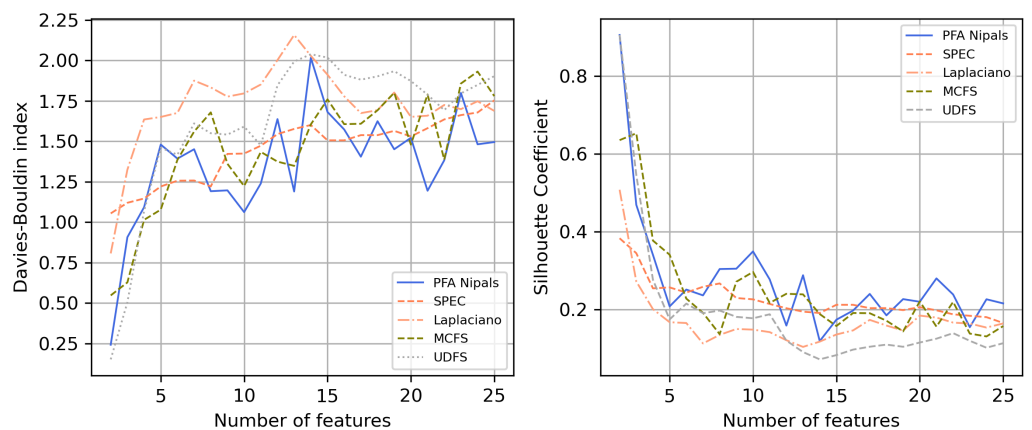


Figure 9. Evolution of indexes with respect to the number of selected features for 5% of missing data. NCI dataset.

For the NCI dataset, we can have indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

4.2.6. Soybean

For the Soybean dataset, Table 8 shows that PFA-Nipals performed better on the Davies–Bouldin index, considering 1% of missing data.

Table 8. Best index results regarding the percentage of missing data. Dataset: Soybean.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.8803	0.9544	0.9372	0.9470	0.8387	0.2170	0.2484	0.1270	0.0462	0.3356
1%	0.8803	0.9750	0.6623	0.8220	0.9750	0.2170	0.2389	0.5579	0.4435	0.2924
2%	0.8011	0.9217	0.9585	0.9596	0.6968	0.3789	0.3590	0.0873	0.1791	0.6064
3%	0.8803	0.9580	0.9585	0.8047	0.4917	0.2170	0.0571	0.0873	0.5262	0.8638
4%	0.8011	0.9750	0.9372	0.9029	0.8268	0.3789	0.2595	0.1270	0.2364	0.3655
5%	0.7454	0.9620	0.7910	1.0000	0.7120	0.4377	0.4209	0.5876	0.0000	0.6117

* Laplacian. Text in bold are the best index values.

In addition, for the 5% missing data case, Figure 10 shows that the best results were found in the selection of the first five features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

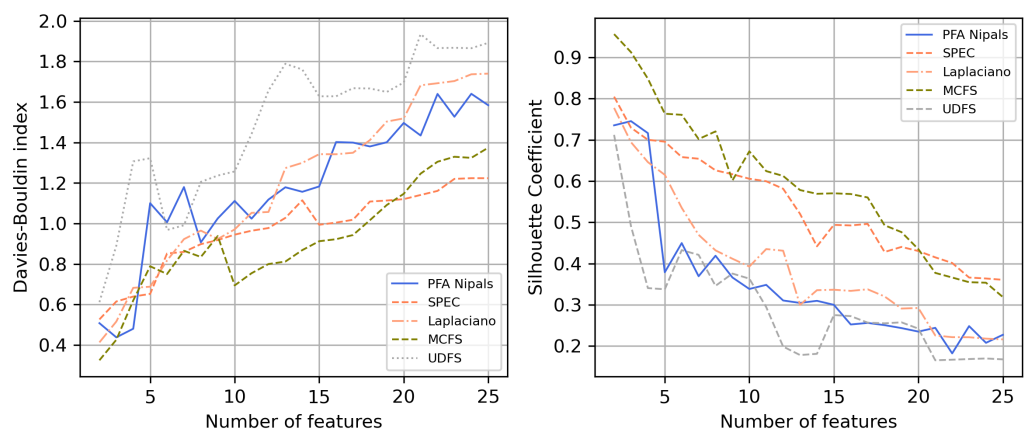


Figure 10. Evolution of indexes with respect to the number of selected features for 5% of missing data. Soybean dataset.

4.2.7. Leukemia

For the Leukemia dataset, Table 9 shows that PFA-Nipals performed better on silhouette coefficients for 0%, 1%, 2%, 3%, 4%, and 5% of missing data. Regarding the Davies–Bouldin index, PFA-Nipals obtained the best results for the 0%, 1%, 2%, 3%, 4%, and 5% of missing data cases.

Furthermore, for the 5% missing data case, Figure 11 shows that the best results were found in the selection of the first 25 features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

For the Leukemia dataset, we can have indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

Table 9. Best index results regarding the percentage of missing data. Dataset: Leukemia.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.7273	0.3865	0.4848	0.3746	0.4555	0.3246	1.0151	0.7080	1.0307	0.8881
1%	0.7273	0.3861	0.4468	0.4509	0.4087	0.3246	1.0114	0.8028	0.9111	0.5875
2%	0.5967	0.3859	0.3413	0.2821	0.3779	0.5628	1.0088	1.1591	1.2071	1.1325
3%	0.6611	0.3866	0.5176	0.4722	0.3779	0.4488	1.0047	0.6912	0.8161	1.1320
4%	0.6493	0.3877	0.4536	0.3842	0.4576	0.4944	1.0023	0.7820	1.0667	0.8857
5%	0.6183	0.3855	0.4590	0.3715	0.4156	0.5500	1.0118	0.8901	1.0864	0.8555

* Laplacian. Text in bold are the best index values.

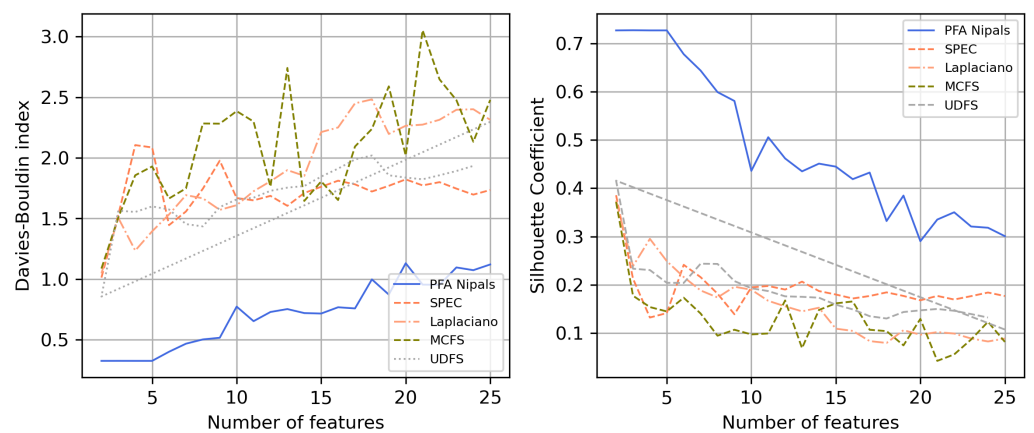


Figure 11. Evolution of indexes with respect to the number of selected features for 5% of missing data. Leukemia dataset.

4.2.8. Epileptic Recognition

For the Epileptic Recognition dataset, Table 10 shows that PFA-Nipals performed better on silhouette coefficients for all cases of missing data, being surpassed in all cases by UDFS. Regarding the Davies–Bouldin index, PFA-Nipals obtained the best results for the 0%, 2%, 4%, and 5% of missing data cases. In this case, PFA-Nipals excelled at obtaining more separate and less sparse clusters.

Table 10. Best index results regarding the percentage of missing data. Dataset: Epileptic Recognition.

% of Missed Data	PFA-Nipals	Silhouette Coefficient				Davies–Bouldin Index				
		SPEC	Lap *	MCFS	UDFS	PFA-Nipals	SPEC	Lap	MCFS	UDFS
0%	0.6840	0.2928	0.3720	0.3209	0.6852	0.8578	0.9020	0.8891	0.8893	1.0936
1%	0.6477	0.2926	0.3723	0.3049	0.6975	0.8925	0.9038	0.8946	0.8638	0.9899
2%	0.6592	0.2927	0.3729	0.2998	0.7102	0.8641	0.9012	0.8949	0.8673	0.9159
3%	0.6548	0.2937	0.3734	0.2921	0.7102	0.9198	0.9003	0.8932	0.9141	0.8995
4%	0.6759	0.2918	0.3730	0.2916	0.6971	0.8500	0.9030	0.8887	0.8884	0.8895
5%	0.6882	0.2916	0.3741	0.2977	0.6943	0.8409	0.8992	0.8887	0.8516	0.8958

* Laplacian. Text in bold are the best index values.

In addition, for the 5% missing data case, Figure 12 shows that the best results were found in the selection of the first five features. Along with this, it can be noted that PFA-Nipals achieved very good results with few selected features.

For the Epileptic Recognition dataset, we can have indications that as the number of missing data increases, PFA-Nipals selects features that, through K-means clustering, find better-defined clusters and with a better separation between the clusters.

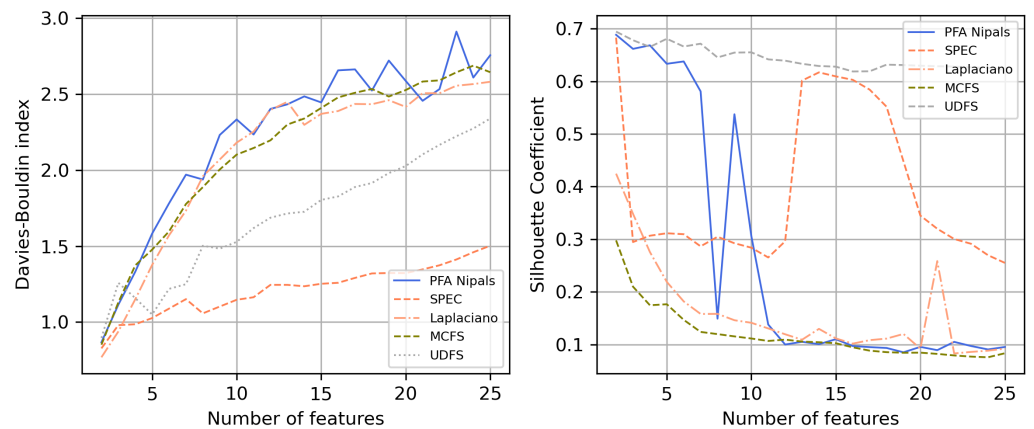


Figure 12. Evolution of indexes with respect to the number of selected features for 5% of missing data. Epileptic Recognition dataset.

4.3. Analysis of Results

We can mention that for the Colon Cancer dataset with 5% missing data, PFA-Nipals improved the definition of the clusters between 14% and 18% and also improved the separation between the clusters between 8% and 28% with respect to SPEC, Laplacian, and MCFS. Moreover, for the Lung Cancer dataset with 5% missing data, PFA-Nipals improved the definition of the clusters between 43% and 65% and also improved the separation between the clusters between 8% and 28% with respect to SPEC, Laplacian, and MCFS. Nonetheless, for the Lymphoma dataset with 5% missing data, PFA-Nipals improved the definition of the clusters between 20% and 84% and improved the separation between the clusters between 35% and 81% with respect to SPEC, Laplacian, and MCFS. The least favorable results for PFA-Nipals were obtained with the Soybean dataset, observing a reduction in the percentage variation between 8% and 25% for the definition of clusters, and between 0% and 369% for the separation between clusters.

In order to determine the observed differences between the performance of the proposed PFA-Nipals and other algorithms and the statistical significance of the results, a nonparametric test was conducted, the denominated Mann–Whitney U-test for the least favorable results obtained, i.e., using the silhouette coefficient results.

The hypothesis tests were the following:

$$H_0 : Sample_1 - Sample_2 = 0$$

$$H_a : Sample_1 - Sample_2 > 0$$

where $Sample_1$ are the silhouette coefficient results of PFA-Nipals and $Sample_2$ are the results of the silhouette coefficient for all other algorithms.

A 95% confidence level for the hypothesis tests was considered and 10,000 Monte Carlo simulation were used to calculate the p -value.

From Table 11, the hypothesis test results indicated that for the Colon Cancer, Lymphoma, NCI, and Leukemia datasets, the p -value calculated was smaller than the significant level $\alpha = 0.05$. Hence, the null hypothesis H_0 was rejected and the alternative hypothesis H_a was accepted for all comparison tests.

For the Lung Cancer dataset, hypothesis H_0 was rejected for the PFA Nipals–Laplacian score and PFA Nipals–SPEC comparisons.

For the Arrhythmia dataset, hypothesis H_0 was rejected for PFA Nipals–MCFS, PFA Nipals–Laplacian score, and PFA Nipals–UDFS comparisons.

For the Soybean dataset, hypothesis H_0 was rejected only for the PFA Nipals–UDFS comparison, and for the Epileptic Recognition dataset, hypothesis H_0 was rejected only for the PFA Nipals–MCFS comparison.

Table 11. Mann–Whitney test/right-sided test.

Dataset	UFS Algorithms and <i>p</i> -Test Value			
	PFA_Nipals–MCFS	PFA_Nipals–Laplacian Score	PFA_Nipals–SPEC	PFA_Nipals–UDFS
Colon Cancer	0.0002	0.0000	0.0000	0.0000
Lung Cancer	0.1959	0.0491	0.0435	0.9914
Lymphoma	0.0000	0.0000	0.0099	0.0000
Arrhythmia	0.0000	0.0000	0.9999	0.0283
NCI	0.0001	0.0000	0.0132	0.0000
Soybean	0.9999	0.9602	0.9999	0.0000
Leukemia	0.0000	0.0000	0.0000	0.0000
Epileptic	0.0000	0.9071	0.9999	0.9999

Text in bold are the *p*-values lower or equal than 0.05.

With these results, we can mention that the estimation of missing data causes a bias in the selection of unsupervised features. Therefore, when it comes to being in the presence of missing data, it is recommended to use a feature selector that works with missing data, such as PFA-Nipals.

Finally, we can see that with more observations, PFA-Nipals achieves better results in the presence of missing data compared to other UFS algorithms. We can also notice that since Lymphoma is the dataset with the greatest difference between algorithms followed by Lung Cancer, Colon Cancer, Leukemia, NCI, Arrhythmia, Epileptic, and Soybean, respectively, PFA-Nipals obtains better results for datasets that present a greater quantity of features in relation to the system observations, which are reflected in the ratio index shown in Table 1.

It can be mentioned that the significance test performed to determine the observed differences for the silhouette coefficient is important in relation to all algorithms in datasets where the number of features is higher than the number of observations (Colon Cancer, Lymphoma, NCI, and Leukemia).

In addition, we can mention that the dataset space can be transformed through iterative nonlinear estimations by partial least squares to calculate the weights of each feature for each component and then separate clusters of weights of equal variance, minimizing the inertia within the cluster and using mini-batches in order to select a subset of features from the original dataset that contains most of the total variability to be explained and that can work with missing data, without having to be estimated previously.

4.4. Execution Times

In terms of execution times for PFA-Nipals, the results depicted in Figure 13 illustrate a significant increase in execution times when dealing with missing data. This trend is further clarified in Table 12, where we observe varying degrees of computational time increases. Notably, the smallest increase in computational time was recorded for the Soybean dataset, with a small 10.67% rise between 0% and 1% of missing data. Comparatively, the Lung Cancer dataset experienced a 40.45% increase, the Colon Cancer dataset showed a 67.34% increase, and the Epileptic dataset presented the most substantial increase, with 406.14%.

These findings indicate a correlation between computational time increase and the quantity of data being analyzed, as well as the disparity between features and observations. Specifically, the results suggest that datasets with a higher ratio of features to observations exhibit a relatively low increase in computational time compared to datasets where the number of observations surpasses the number of features while maintaining an equivalent number of data points. This relationship becomes evident when comparing the NCI and Leukemia datasets to the Arrhythmia dataset.

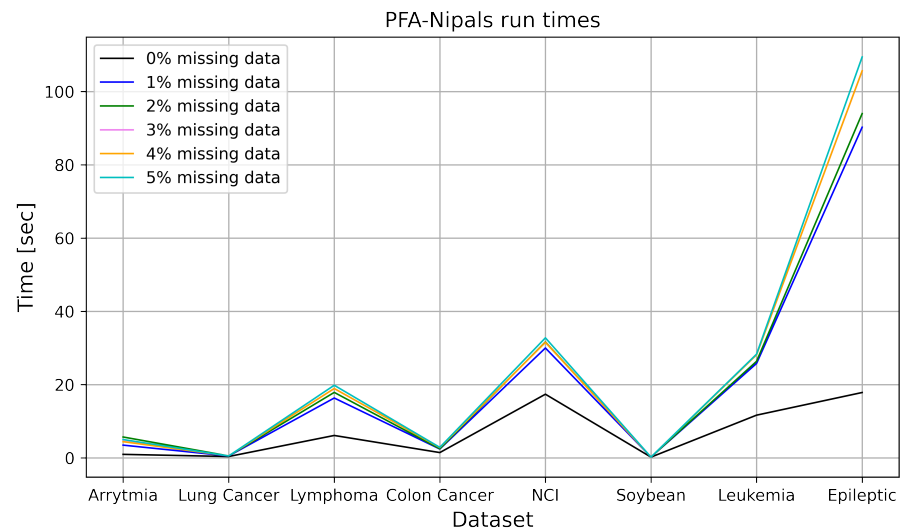


Figure 13. PFA-Nipals execution times for 0%, 1%, 2%, 3%, 4%, and 5% of missing data.

Table 12. Percentage variation of computational times between 0% and 1% of missing data for each dataset.

Dataset	Percentage Variation of Computational Times
Arrhythmia	266.80%
Lung Cancer	40.45%
Lymphoma	167.52%
Colon Cancer	67.34%
NCI	72.63%
Soybean	10.67%
Leukemia	121.18%
Epileptic	406.14%

5. Discussion

Two highly important problems were considered in this study. The first focused on the reduction in dimensionality, which is an important preprocessing step for machine learning, and the second was based on the absence or missing data on the features of systems to be studied.

The importance of the aforementioned is that when it is decided that a feature is unnecessary, we save costs by extracting it; in addition, simple models are more robust in small datasets, by having less variation in sample details. Together with the above, the fewer features, the better the behavior of a particular system or phenomenon. Regarding missing data, the estimation or elimination of observations produces biases when selecting relevant features for the system.

Therefore, in this paper, we proposed an unsupervised feature selection method called PFA-Nipals, for which we applied a synthetic clustering problem which was able to select the two principal features to group the clusters through the K-means algorithm correctly.

To validate our algorithm and prove that it performed well with missing data for feature selection, experiments were performed on datasets which were treated as clustering data. It can be observed from the results that the proposed algorithm outperformed all competitors in six out of eight datasets without the presence of missing data.

In addition, we can mention that as the number of missing data increased, PFA-Nipals selected features that, through K-means clustering, found better-defined clusters with a better separation between the clusters compared to other UFS algorithms. Finally, it was observed that the proposed algorithm obtained faster execution times with respect to the ones presented in the state of the art, reducing overall computational times.

Therefore, we conclude that the proposed algorithm is robust in selecting features with missing data and has excellent performance for datasets with more dimensions than observations. Its benefits make it especially suitable for unsupervised learning.

We expect that the proposed algorithm could be used widely within different areas, for example, for early detection of diseases in medical sciences, to analyze weather data to forecast extreme events, and also to study the key features in attenuation laws applied to earthquake engineering, as in all mentioned cases, experimental data are usually large datasets with inherent missing data.

Finally, as a summary, the main contributions of the proposed method PFA-Nipals are: for unsupervised feature selection, the overall computational time was reduced when unnecessary features were eliminated; it improved the robustness of simple models with small datasets by reducing irrelevant variations; it helped explain the data with fewer features, leading to a deeper understanding of the underlying processes; it can work with missing data without the need to estimate or remove observations; it is reliable when the number of attributes exceeds the number of observations, and the observations are insufficient for effective feature selection training; it can find a consistent set of relevant features without biasing the explained variability.

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

Funding: This work was partly funded by the BECA ESTUDIO DE DOCTORADO, UNIVERSIDAD DE TALCA. The APC was funded by the Faculty of Engineering, Campus Curicó, University of Talca.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Solorio-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **2020**, *53*, 907–948. [[CrossRef](#)]
2. Baştanlar, Y.; Ozuysal, M. Introduction to machine learning. *Methods Mol. Biol.* **2014**, *1107*, 105–128. [[CrossRef](#)] [[PubMed](#)]
3. Mao, K. Identifying Critical Variables of Principal Components for Unsupervised Feature Selection. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2005**, *35*, 339–344. [[CrossRef](#)] [[PubMed](#)]
4. Ding, C.; Peng, H. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the 2003 IEEE Bioinformatics Conference*; CSB: Stanford, CA, USA, 2003; pp. 523–528. [[CrossRef](#)]
5. Ding, C.; Peng, H. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.* **2005**, *3*, 185–205. [[CrossRef](#)] [[PubMed](#)]
6. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [[CrossRef](#)] [[PubMed](#)]
7. Kim, S.B.; Rattakorn, P. Unsupervised feature selection using weighted principal components. *Expert Syst. Appl.* **2011**, *38*, 5704–5710. [[CrossRef](#)]
8. Zhao, Z.A.; Liu, H. *Spectral Feature Selection for Data Mining*; CRC Press: Boca Raton, FL, USA, 2011; Volume 1, p. 216.
9. Groves, R.M. *Survey Errors and Survey Costs*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1989. [[CrossRef](#)]
10. Schafer, J.L. *Analysis of Incomplete Multivariate Data*; Chapman and Hall/CRC: Boca Raton, FL, USA, 1997. [[CrossRef](#)]
11. Buck, S.F. A Method of Estimation of Missing Values in Multivariate Data Suitable for Use with an Electronic Computer. *J. R. Stat. Soc. Ser. B (Methodol.)* **1960**, *22*, 302–306. [[CrossRef](#)]
12. Pastor, J.B.N.; Vidal, J.M.L. Análisis de datos faltantes mediante redes neuronales artificiales. *Psicothema* **2000**, *12*, 503–510.

13. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm A. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–38. [[CrossRef](#)]
14. Rosas, J.F.M.; Álvarez Verdejo, E. Métodos de imputación para el tratamiento de datos faltantes: Aplicación mediante R/Splul. *Rev. MéTodos Cuantitativos Para Econ. Empresa* **2009**, *7*, 3–30.
15. Jolliffe, I.T. Discarding Variables in a Principal Component Analysis. I: Artificial Data. *Appl. Stat.* **1972**, *21*, 160. [[CrossRef](#)]
16. Kim, Y.B.; Gao, J. Unsupervised Gene Selection For High Dimensional Data. In Proceedings of the Sixth IEEE Symposium on BioInformatics and BioEngineering (BIBE'06), Arlington, VA, USA, 16–18 October 2006; pp. 227–234. [[CrossRef](#)]
17. Li, Y.; Lu, B.L.; Zhang, T.F. Combining Feature Selection With Extraction: Component Analysis. *Int. J. Artif. Intell. Tools* **2009**, *18*, 883–904. [[CrossRef](#)]
18. Lu, Y.; Cohen, I.; Zhou, X.S.; Tian, Q. Feature Selection Using Principal Feature Analysis. In Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany, 24–29 September 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 301–304. [[CrossRef](#)]
19. Chang, X.; Nie, F.; Yang, Y.; Zhang, C.; Huang, H. Convex Sparse PCA for Unsupervised Feature Learning. *ACM Trans. Knowl. Discov. Data* **2016**, *11*, 1–16. [[CrossRef](#)]
20. Wan, M.; Lai, Z. Feature Extraction via Sparse Difference Embedding (SDE). *KSII Trans. Internet Inf. Syst.* **2017**, *11*, 3594–3607. [[CrossRef](#)]
21. Zhu, Y.; Zhang, X.; Wang, R.; Zheng, W.; Zhu, Y. Self-representation and PCA embedding for unsupervised feature selection. *World Wide Web* **2018**, *21*, 1675–1688. [[CrossRef](#)]
22. Bouveyron, C.; Latouche, P.; Mattei, P.A. Bayesian variable selection for globally sparse probabilistic PCA. *Electron. J. Stat.* **2018**, *12*, 3036–3070. [[CrossRef](#)]
23. Solorio-Fernández, S.; Ariel Carrasco-Ochoa, J.; Martínez-Trinidad, J.F. A systematic evaluation of filter Unsupervised Feature Selection methods. *Expert Syst. Appl.* **2020**, *162*, 113745. [[CrossRef](#)]
24. Mitra, P.; Murthy, C.; Pal, S. Unsupervised feature selection using feature similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 301–312. [[CrossRef](#)]
25. Cai, D.; Zhang, C.; He, X. Unsupervised Feature Selection for Multi-Cluster Data. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 25–28 July 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 333–342. [[CrossRef](#)]
26. Yang, Y.; Shen, H.T.; Ma, Z.; Huang, Z.; Zhou, X. L_{2,1}-Norm Regularized Discriminative Feature Selection for Unsupervised Learning. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, 16–22 July 2011; AAAI Press: Menlo Park, CA, USA, 2011; Volume 2, pp. 1589–1594.
27. Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; Lu, H. Unsupervised Feature Selection Using Nonnegative Spectral Analysis. *Proc. Natl. Conf. Artif. Intell.* **2012**, *2*, 1026–1032. [[CrossRef](#)]
28. Tang, C.; Liu, X.; Li, M.; Wang, P.; Chen, J.; Wang, L.; Li, W. Robust unsupervised feature selection via dual self-representation and manifold regularization. *Knowl.-Based Syst.* **2018**, *145*, 109–120. [[CrossRef](#)]
29. Liu, T.; Hu, R.; Zhu, Y. Completed sample correlations and feature dependency-based unsupervised feature selection. *Multimed. Tools Appl.* **2022**, *82*, 1–22. [[CrossRef](#)]
30. Alabsi, B.A.; Anbar, M.; Rihan, S.D.A. CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks. *Sensors* **2023**, *23*, 6507. [[CrossRef](#)] [[PubMed](#)]
31. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [[CrossRef](#)]
32. Tenenhaus, M. *La Régression PLS Théorie et Pratique*; Editions TECHNIP: Paris, France, 1998.
33. Sculley, D. Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web, Raleigh North, CA, USA, 26–30 April 2010; pp. 1177–1178. [[CrossRef](#)]
34. Steinbach, M.; Karypis, G.; Kumar, V. A Comparison of Document Clustering Techniques. 2000. Available online: <https://conservancy.umn.edu/handle/11299/215421> (accessed on 20 Januray 2022).
35. Rosenberg, A.; Hirschberg, J. V-Measure: A conditional entropy-based external cluster evaluation measure. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007, Prague, Czech Republic, 28–30 June 2007; pp. 410–420.
36. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2003**, *3*, 583–617. [[CrossRef](#)]
37. He, X.; Cai, D.; Niyogi, P. Laplacian Score for Feature Selection. *Adv. Neural Inf. Process. Syst.* **2005**, *18*, 507–514.
38. Zhao, Z.; Liu, H. Spectral Feature Selection for Supervised and Unsupervised Learning. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 1151–1157. [[CrossRef](#)]
39. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: <https://ergodicity.net/2013/07/> (accessed on 1 April 2021).
40. Dagnino, J. Bioestadística y Epidemiología DATOS FALTANTES (MISSING VALUES). *Rev. Chil. Anest.* **2014**, *43*, 332–334.

41. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [[CrossRef](#)]
42. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.