


Article

# Fault Diagnosis by Multisensor Data: A Data-Driven Approach Based on Spectral Clustering and Pairwise Constraints

Massimo Pacella \*  and Gabriele Papadia 

Department of Engineering for Innovation, University of Salento, 73100 Lecce, Italy;  
gabriele.papadia@unisalento.it

\* Correspondence: massimo.pacella@unisalento.it

Received: 16 November 2020; Accepted: 7 December 2020; Published: 10 December 2020



**Abstract:** This paper deals with clustering based on feature selection of multisensor data in high-dimensional space. Spectral clustering algorithms are efficient tools in signal processing for grouping datasets sampled by multisensor systems for fault diagnosis. The effectiveness of spectral clustering stems from constructing an embedding space based on an affinity matrix. This matrix shows the pairwise similarity of the data points. Clustering is then obtained by determining the spectral decomposition of the Laplacian graph. In the manufacturing field, clustering is an essential strategy for fault diagnosis. In this study, an enhanced spectral clustering approach is presented, which is augmented with pairwise constraints, and that results in efficient identification of fault scenarios. The effectiveness of the proposed approach is described using a real case study about a diesel injection control system for fault detection.

**Keywords:** semi-supervised classification; spectral clustering; PCA; fault detection; fuel-injection system

## 1. Introduction

To accurately inspect the operating conditions of an internal combustion engine, several sensors are used to collect real-time measurements. For instance, to control fuel consumption and emissions of pollutants into the environment, the exhaust after-treatment process of an internal combustion engine is monitored by various classes of sensors. In [1,2] is discussed the problem of fault diagnosis based on data sampled by a large number of sensors, measuring, for example, the vehicle velocity, the average engine rotational speed, and the air mass flow. The measurements sampled by the sensors, disseminated on the supply and after-treatment line of the engine (Figure 1), contain a high amount of information, which is fundamental not only in the regulation of the systems but also to provide an interpretative model of the process, which facilitates a rapid diagnosis of potential faults.

Considering measurements are collected faster than they are analyzed, automatic fault diagnosis procedures are required to rapidly and efficiently process data and provide detailed results [3]. A frame of fault diagnosis based on multiple sensors includes: (i) data acquisition; (ii) feature extraction; (iii) fault diagnosis. In the signal acquisition step, many types of sensors are considered, which provide a large number of signals. In the second step, feature extraction aims to extract representative features from the collected signals through dimension reduction. The objective is to separate sensitive from insensitive information that may affect the diagnosis results as well as computational efficiency. In fault diagnosis, clustering is used for determining groupings within data (with higher similarity within groups and lower similarity among groups) and assigning labels to data points according to these groupings. In Figure 2, the flow of a conventional approach is represented, in which the first step allows the definition of data-filtering criteria for events detection. Once an organized database of

elements is obtained, it is possible to apply a feature extraction approach (such as the Principal Component Analysis—PCA), which returns a transformed database. Clustering is implemented to separate data into clusters (families of events), which allow experienced personnel to diagnose the fault. The approach schematized in Figure 2 was exploited in [1]. A similar approach is implemented in this study.

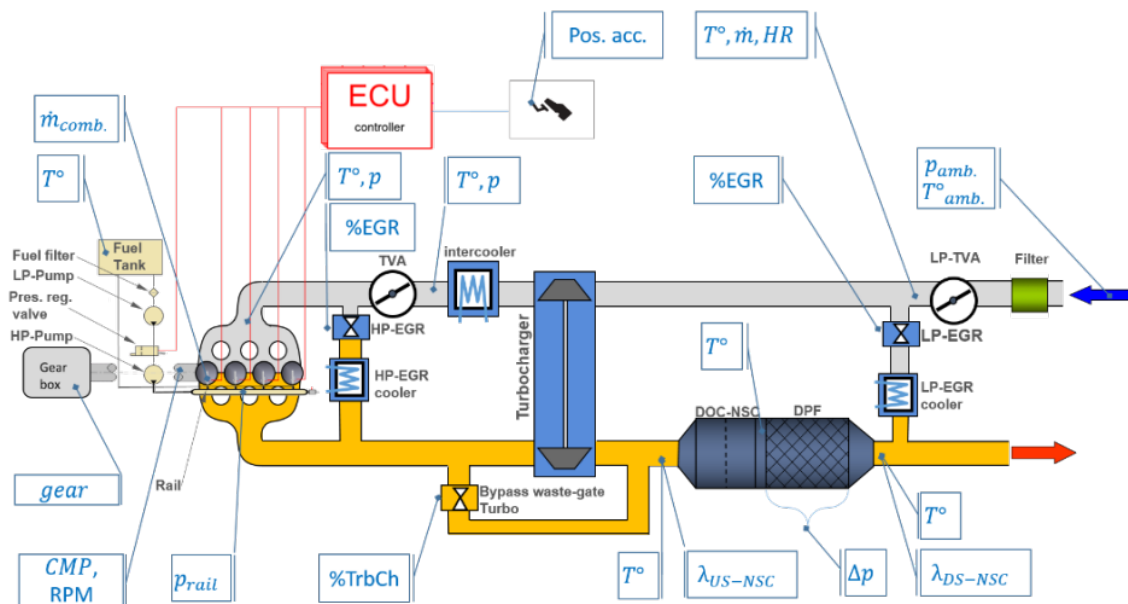


Figure 1. Layout schema of an internal combustion engine [1,2].

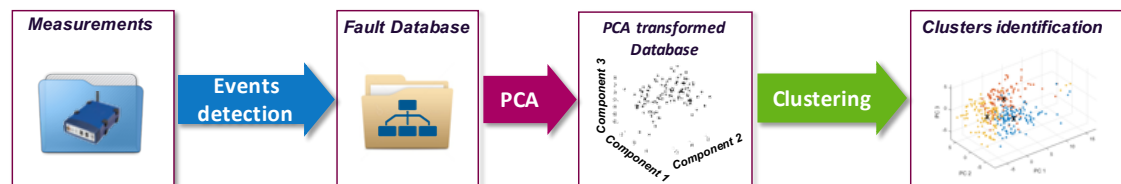


Figure 2. A frame of fault diagnosis based on multisensor data.

Several studies presented in the literature for fault diagnosis are based on classification methods (support vector machine, naive Bayes, and logistic regression). These approaches are basically supervised because labeled training data with known fault classes are employed to train the classifier first. Subsequently, the classifier processes new data to diagnose potential faults by matching the patterns against the measurement data [4]. However, reliable measurements under a specific label (known fault condition) may not be available in actual applications. When labeled data are not available for training, the unsupervised classification of measurement data provided by several sensors during a fault event should be considered to support fault diagnosis. The unsupervised classification is based on partitioning of profile data to isolate the fault events in a restricted number of scenarios, each one described by a reference pattern. Then, this pattern could be examined by an expert for decision-making, in other words, to find root causes.

In this paper, a semi-supervised data-driven approach is discussed, in which combined labeled and unlabeled measurement data are used to train the model. Our proposed approach is based on a clustering method in which we assume to have information about pairs of vectors that do not belong to the same cluster (cannot-links) and information about pairs of vectors that belong to the same cluster (must-links) [5]. This information, which may be available from experienced personnel concerning a small subsample of data measurements, may lead to enhanced performance in the clustering process of data.

The most common clustering technique is the  $K$ -means and its variants [6,7]. These methods partition the data into several  $K$  groups with the goal is to minimize a within-cluster dispersion measure. However, the  $K$ -means algorithm performs poorly in case the dataset is not the union of well-separated spherical structures. On the other hand, spectral methods [8,9] are recommended to handle irregularly shaped clusters by using the information of an affinity matrix, which is used for measuring the similarity among data points. Spectral clustering (SC) is an important subject of research in recent years [9]. If the shape of clusters deviates from well-separated spherical structures [10], for which  $K$ -means performs well [11], SC is an effective approach. The SC method has been shown robust concerning the geometry of the clusters, noise, and outliers [12].

The SC approach reduces clustering to a problem of graph partitioning [13–16]. The first step of SC involves forming a positive semi-definite affinity matrix with each entry that refers to the measure of similarity linking each pair of data points. Then, by consulting only a few eigenvalues and eigenvectors of such a matrix, SC maps the data points to  $\mathbb{R}^K$  (where  $K$  is the number of selected eigenvectors of the matrix). This mapping involves the projection of data onto a new space, in which points form tight clusters, and simple clustering methods can be used. SC algorithms are particularly well suited for clustering in a high-dimensional setting. Such is the case of signals acquired by a multisensor monitoring system as emerged with present control systems, which govern the functioning of an internal combustion engine. One of the most significant issues in this application refers to the high number of variables that define the state of the modeled process. To increase clustering accuracy and to reduce the computational cost, it is necessary to reduce the dispersion of raw data to allow a meaningful classification.

The SC approach is analyzed in this study for multisensor data. The effectiveness of the SC approach is illustrated using a real case study concerning a diesel injection control system for fault detection. In a diesel engine, fuel injection into the cylinders is possible thanks to reliable injectors [17]. Nevertheless, a fault may occur due to a flash opening of the injector, which causes an unspecified pressure drop in the fuel rail. The diagnostic system interprets these events as the repeated opening of a worn-out relief valve. This safety component is equipped on a heavy-duty diesel engine to prevent high pressures in the fuel rail, letting fuel to flow back, and so avoiding the system to move into a dangerous condition [18]. In our case study, the diagnosis of the fault cannot be efficiently performed by practitioners due to a large number of sensors, and hence data-driven approaches are required to support the root-cause analysis of a fault.

The contributions of our research are as follows: (i) we propose a new approach for including pairwise constraints information into SC; (ii) we prove that this approach improves the quality of classification using a real case study about a diesel injection control system for fault detection. Despite the specific case study described in this paper, the proposed approach can be exploited in several applications, in which multisensor measurement data are collected on a process. In particular, in the manufacturing field, where the final quality of the manufactured part is more and more often related to the faults of the machining processes [19]. Moreover, the proposed methodology may have a widespread application in other experimental settings of fault diagnosis of interest in the recent literature, such as vibrational signals of induction motors [20] or bearing faults in rotational machinery [21,22].

The outline of the present paper is as follows. Section 2 provides an overview of feature extraction for process data. Section 3 presents an overview of  $K$ -means clustering methods while SC and the proposed SC augmented with pairwise constrain information is described in subsequent Section 4. The effectiveness of the clustering operation is measured by the validation indices described in Section 5. Numerical validation of representative datasets is presented in Section 6, where the case study concerning an injection control system for a diesel engine is considered. The different performances of  $K$ -means and SC are compared in demonstrating the fault scenario. Finally, Section 7 provides conclusions of this study and presents directions for future research.

## 2. Feature Extraction of Process Data with a High Number of Variables

A large number of sensors employed to monitor the state of the process may result in a challenge for a study aimed to define a data-driven approach for fault diagnosis. PCA is a well-known method to reduce the dispersion of the multisensor measurements and their dimensionality. PCA results in the transformation of original variables into a small number of features (principal components, PCs).

The preliminary operation of the PCA approach consists of computation of the sample covariance matrix and its eigendecomposition. The resulting eigenvalues are sorted in decreasing order, where each eigenvalue is related to the fraction of variance explained by the linked PC. Corresponding orthogonal eigenvectors describe a basis of space whose directions are referred to as the maximum variability of the data. The advantage of PCA as a dimensionality reduction algorithm consists of reducing the number of variables while preserving as much variability as possible of the initial raw measurements. In [1], the PCA is used as a feature extraction method for the clustering of multichannel profiles, where analysis concerns the root causes of the fault related to an emission control system in a diesel engine.

Consider the case of a  $P$ -sensor data, of  $M$  samples. A generic sample is stored in a matrix designated as  $\mathbf{X} \in \mathbb{R}^{P \times M}$  and addressed by indexes:  $j, i$  related to rows and columns of  $\mathbf{X}$ , respectively. Let  $\bar{\mathbf{x}} = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_i$  be the average vector of data and let  $\mathbf{x}_i^c = \mathbf{x}_i - \bar{\mathbf{x}}$  be the centered vector obtained from  $\mathbf{x}_i$  by subtracting the average vector. The entire dataset can be represented by matrix  $\mathbf{X}^c \in \mathbb{R}^{P \times M}$  (the rows represent the  $P$  variables and the columns are the  $M$  samples  $\mathbf{x}_i^c$ ).

The aim of PCA is to solve the problem of approximating the data matrix  $\mathbf{X}^c$  with another matrix  $\tilde{\mathbf{X}}^c$  which has a lower rank, where the approximation objective is to minimize the distance between  $\mathbf{X}^c$  and  $\tilde{\mathbf{X}}^c$ .  $N$  is a given upper bound for the rank of matrix  $\tilde{\mathbf{X}}^c$  ( $N < P$ ). Hence, denoting by  $\tilde{\mathbf{U}}$  the matrix formed by the first  $N$  columns of  $\mathbf{U}$ , which correspond to the first  $N$  larger singular values of  $\mathbf{X}^c$ , a data sample vector of  $P$  points  $\mathbf{x}_i$  ( $i = 1, \dots, M$ ) is projected to a feature space as  $\tilde{\mathbf{U}}^T(\mathbf{x}_i - \bar{\mathbf{x}})$ . This is the vector of  $N$  coordinates  $\mathbf{t}_i = (t_{i1}, \dots, t_{iN})$  which represent the so-called *scores* (PC-features) of vector  $\mathbf{x}_i$ . Let  $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^M$  represent the dataset of scores resulting from PCA.

## 3. K-Means Clustering

The  $K$ -means algorithm is a universal technique in clustering due to its simplicity and ease of use, despite it suffers from setting initial conditions and a non-spherical-shape characteristic of the dataset. Let  $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^M$  represent the original dataset. A generic point  $\mathbf{t}_i \in \mathbb{R}^N$ , assigned to a group, has high intra-cluster similarity ( $SSW$ ) with the remaining points belonging to the same cluster while it has low inter-cluster similarity ( $SSB$ ) with the remaining points assigned to different groups.

These parameters are analytically expressed as  $SSW = \sum_{k=1}^K N_k \sum_{\mathbf{t}_i \in \mathcal{C}_k} d(\mathbf{t}_i, \mathbf{c}_k)$  and  $SSB = \sum_{k=1}^K N_k \cdot d(\mathbf{c}_k, \bar{\mathbf{c}})$ . Where  $d: \mathbb{R}^N \times \mathbb{R}^N \rightarrow [0, \infty)$  is a distance metric in  $\mathbb{R}^N$  (in this work, the Euclidean distance is used).  $SSW$  represents the sum of the squared distance between each  $i$ -th data point  $\mathbf{t}_i$  and its closed centroid:  $\mathbf{c}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\mathbf{t}_i \in \mathcal{C}_k} \mathbf{t}_i$ , in other words the barycentre of the  $k$ -th cluster  $\mathcal{C}_k$ , where  $N_k$  is the number of points of the  $k$ -th cluster.  $SSW$  represents the within-cluster variance. The objective of clustering is to find cluster centroids that minimize  $SSW$  (tight clusters).  $SSB$  is the sum of the squared distance between  $\mathbf{c}_k$ , previously introduced, and  $\bar{\mathbf{c}}$ , which is the mean position of all  $K$  centroids.  $SSB$  represents the between-cluster variance. Clustering should maximize  $SSB$  (clusters well separated).

To solve such a problem, minimize  $SSW$  and maximize  $SSB$ , the  $K$ -means algorithm [23] executes two main steps: (i) initialization of  $K$  centroids uniformly distributed between points to be classified; (ii) consequent aggregation of points around centroids, using distance as a criterion of similarity. Once a cluster of points has been settled, the centroid is determined as a weighted average of the points. This step, repeated for each cluster, is followed by a re-calculation of the clusters and related centroids. Iterating the process, when the position of the centroids does not vary significantly, the algorithm reaches convergence.

In [24], several conditions were considered under which the original  $K$ -means algorithm fails or requires a long time before it converges to an adequate solution. As a result, a variant of the original algorithm, named  $K$ -means++, was introduced. This variant produces a better classification along with a reduction of the  $SSW$  parameter and thus compactness of the clusters compared to the initial  $K$ -means.

In particular, in  $K$ -means++, a specific way of choosing centers for the  $K$ -means algorithm is implemented. Let  $d(\mathbf{t}')$  denote the shortest distance from a data point to the closest center chosen. The first centroid is arbitrarily chosen in the overall set of points to be grouped. Remaining  $K - 1$ , are chosen according to the probability distribution  $\frac{d(\mathbf{t}')^2}{\sum_{\mathbf{t} \in \mathcal{T}} d(\mathbf{t})^2}$ . Once the  $K$  centroids are labeled, the  $K$ -means++ algorithm proceeds as the original  $K$ -means algorithm. When compared to the original  $K$ -means algorithm, the  $K$ -means++ shows a better classification accuracy and a faster convergence. We summarize the pseudocode implemented in this study in Algorithm 1.

---

**Algorithm 1**  $K$ -means ++ clustering with distance metric  $d$  (use Euclidean distance)

---

**Input:**  $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^M$  (Data),  $K \in \mathbb{N}_0$

**Output:**  $Y$  (Labels)

- 1: Select first centroid  $\mathbf{c}_1$  uniformly at random from  $\{\mathbf{t}_i\}_{i=1}^M$ .
  - 2: Select a new centroid  $\mathbf{c}_i$  as  $\mathbf{c}_i = \mathbf{t}' \in \mathcal{T}$  with probability  $\frac{d(\mathbf{t}')^2}{\sum_{\mathbf{t} \in \mathcal{T}} d(\mathbf{t})^2}$ ;  $d(\mathbf{t})$  denotes the shortest distance from generic point  $\mathbf{t}$  to the nearest centroid already selected.
  - 3: Repeat step 2 until  $K$  centroids have been collected:  $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}$ .
  - 4: For each  $i \in \{1, \dots, K\}$ , set the cluster  $\mathcal{C}_i$  to be a group of points in  $\mathcal{T}$  that are the nearest to  $\mathbf{c}_i$  then they are to  $\mathbf{c}_j$  for  $j \neq i$ .
  - 5: For each  $i \in \{1, \dots, K\}$ , set  $\mathbf{c}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{t} \in \mathcal{C}_i} \mathbf{t}$ , in other words the barycentre of  $\mathcal{C}_i$ .
  - 6: Loop steps 4 and 5 until set  $\mathcal{C}$  no longer changes.
  - 7: Compute  $Y$  as an labels array containing cluster indexes of each  $\{\mathbf{t}_i\}_{i=1}^M$ .
- 

#### Selecting the Number of Clusters by Elbow Method

The  $K$ -means algorithm requires the preliminary information concerning the number of clusters and so the number of centroids around which to aggregate the nearest points. This feature makes the  $K$ -means algorithm particularly attractive in unsupervised classification problems. Since the data structure is not known, it is convenient to use a single degree of freedom consisting of the number of clusters  $K$ .

The compactness of clusters is one of the criteria used to assess the quality of the clustering. This characteristic is quantified by parameter  $SSW$ . Selecting the optimal number of clusters is usually based on  $SSW(K)$  as a function of variable  $K \in \mathbb{N}_0$ . This function is decreasingly monotonous since as more centroids are introduced, smaller are the clusters, and consequently, the smaller is  $SSW(K)$  (compactness criterion). There is an optimal value of  $K$  above, in which the  $SSW(K)$  parameter does not decrease appreciably. This condition represents an elbow of the curve followed by a plateau for increasing  $K$ -values. Search for optimal  $K$  results in the identification of the maximum curvature point of  $SSW(K)$ . In [25], an algorithm called *Kneedle* is provided to determine the maximum curvature point in a discrete distribution. This approach allows optimization by calibrating appropriate threshold values that influence the sensitivity of the technique to converge to the optimum point. In the present study, *Kneedle* is used on offline data resulting from the experiments. The algorithm is implemented following the  $K$ -means to calibrate, in a closed-loop, the  $K$  input for clustering iteration.



#### 4. Spectral Clustering

The  $K$ -means performs well if data fit a Gaussian model. On the other hand, SC does not pre-assume any model. SC aims to optimize certain criterion that measures the quality of graph partitions. Dissimilarly from the  $K$ -means algorithm that works directly on data points, the SC method starts from an affinity matrix that measures the pairwise similarity of the data points. This corresponds to a graph partition, such that the intra-group edge weights are high and the inter-group edge weights are low. Mathematically, this is the problem of finding eigenvectors of the Laplacian graph from the affinity matrix and then clustering eigenvectors into clusters.

Given the dataset  $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^M$ , whereby there is some degree of similarity between two generic  $\mathbf{t}_i$  and  $\mathbf{t}_j$  points. Consequently, it is possible to create a graph that reflects the properties of  $\mathcal{T}$ , which can be efficiently processed by the clustering algorithm. The first step of SC involves forming a positive semi-definite affinity matrix  $\mathbf{A} \in \mathbb{R}^{M \times M}$  such that each entry  $a_{ij}$  represents the affinity between data  $\mathbf{t}_i$  and  $\mathbf{t}_j$ . Standard SC methods first construct a graph  $G = (\mathcal{T}, \mathbf{A})$ , where  $\mathcal{T}$  denotes the set of vertices and  $a_{ij}$  gives the weight of the edge that connects  $\mathbf{t}_i$  and  $\mathbf{t}_j$ . In these terms, the clustering objective is reduced to identifying that particular partition (a subset of vertexes) of the graph whose edge weights show low values with contiguous partitions. The connections between internal vertexes are associated with high similarity indexes. The partitioning of the graph is then obtained by assigning large edge weights within each cluster and small edge weights between each cluster.

The original formulation of SC uses the traditional Gaussian kernel-based similarity. Let  $d : \mathbb{R}^N \times \mathbb{R}^N \rightarrow [0, \infty)$  be a distance metric in  $\mathbb{R}^N$  (in this work, the Euclidean distance is used). Set  $a_{ij} = \exp\left(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)^2}{2\sigma^2}\right)$ , if  $i \neq j$ , 0 otherwise.  $\sigma$  is a global scaling parameter for every object pair, which always has to be set manually. However, the effect of  $\sigma$  is important, and the optimal  $\sigma$  for different data turned out to be very different. Moreover, there may not be a single value of  $\sigma$  that works well for all the data. In fact, when  $\sigma$  is set small,  $a_{ij}$  cannot effectively capture the high correlation between distant objects in a large sparse cluster. On the contrary, when  $\sigma$  is set large, objects from different but nearby dense clusters will then more likely to be misjudged as similar.

To address the issue, a local scale parameter for each point allows self-tuning of the point-to-point distances according to the local statistics of the neighborhoods surrounding points  $\mathbf{t}_i$  and  $\mathbf{t}_j$ . The local scaling parameter  $\sigma_i$  for each object  $\mathbf{t}_i$  is defined as the distance between  $\mathbf{t}_i$  and its  $l$ -th nearest neighbor ( $l$  can be empirically set). For an object  $\mathbf{t}_i$  in a sparse cluster,  $\sigma_i$  is large. This enlarges the similarity between  $t_i$  and other distant objects in the same cluster of  $\mathbf{t}_i$ . Also, a dense cluster gives a small  $\sigma_i$ , which effectively decreases the similarity between  $t_i$  and objects from nearby clusters. Thus, the affinity between the points  $\mathbf{t}_i$  and  $\mathbf{t}_j$  can be written as:  $a_{ij} = \exp\left(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)^2}{\sigma_i \sigma_j}\right)$ .

Scale parameters  $\sigma_i$  calibrate the similarity index according to the dispersion of points around generic  $t_i$ . For each index  $i$ ,  $\sigma_i$  was computed as a self-calibrating parameter based on point distribution [26]. The selection of the local scale  $\sigma_i$  can be done by considering the local statistics of the neighborhood of point  $\mathbf{t}_i$ . A simple approach, which is used for the experiments in this paper, is:  $\sigma_i = d(\mathbf{t}_i, \mathbf{t}_{knn})$ , where  $\mathbf{t}_{knn}$  is the  $k$  nearest neighbor of point  $\mathbf{t}_i$  [27].

In our experiments, we employed the value of  $knn = 7$ , which showed good results even for high-dimensional data. However, compared to methods that use a global scale, this approach comes with a slightly higher computational cost considering it calls for a  $knn$  search for each data point in the process of forming the affinity matrix. Other approaches discussed in the literature for defining the affinity matrix are the Dominant Neighborhoods [28] and the Consensus of  $knn$  [29].

##### 4.1. Laplacian Graph

Let  $\mathbf{A} \in \mathbb{R}^{M \times M}$  with the  $(i, j)$  entry  $a_{ij} = \exp\left(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)^2}{\sigma_i \sigma_j}\right)$  be the affinity matrix. The degree matrix  $\mathbf{D}$  is a diagonal matrix associated with  $\mathbf{A}$  with  $d_{ii} = \sum_{j=1}^M a_{ij}$  be the sum of  $\mathbf{A}$ 's  $i$ -th row.

The combinatorial, degree-normalized and symmetric normalized Laplacian graph are defined as follows.

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad , \quad \mathbf{L}_{\text{norm}} = \mathbf{D}^{-1}\mathbf{L} \quad , \quad \mathbf{L}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \quad (1)$$

Different types of normalization can be considered for SC. For example, the *normalized cuts* (NCuts) [30] method employs random walk-based normalization  $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$  while the *Ng-Jordan-Weiss* (NJW) [8] method uses symmetric normalization  $\mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ , where  $\mathbf{I}$  is the identity sparse matrix. In this study, the normalized SC is implemented as it maximizes within-cluster similarity and minimizes between-cluster similarity, while unnormalized SC only minimizes between-cluster similarity [9]. The matrix  $\mathbf{L}$  is semi-positive definite and its eigenvalues are in the interval  $[0, 2]$ . The eigenvalues of  $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$  range in the interval  $[-1, 1]$ .

Therefore, SC with  $\mathbf{L}_{\text{sym}}$  combined to NJW algorithm [8] is implemented. We denote the eigenvalues of  $\mathbf{L}_{\text{sym}}$  (identical to those of  $\mathbf{L}_{\text{norm}}$ ) by  $\lambda_1 \leq \dots \leq \lambda_n$ , and the corresponding eigenvectors by  $\phi_1, \dots, \phi_n$ . To cluster the data into  $K$  groups according to [8], the first step consists of computing an  $M \times K$  matrix  $\Phi$  whose columns are given by  $\{\phi_j\}_{j=1}^K$ . The rows of  $\Phi$  are then normalized to obtain the matrix  $\mathbf{V}$ , that is  $v_{ij} = \Phi_{ij} / (\sum_j \Phi_{ij}^2)^{1/2}$ . Let  $\{\mathbf{v}_i\}_{i=1}^M \in \mathbb{R}^K$  denote the rows of  $\mathbf{V}$ . Eventually,  $K$ -means is applied to cluster the  $\{\mathbf{v}_i\}_{i=1}^M$  into  $K$  groups, which defines a partition of data  $\{\mathbf{t}_i\}_{i=1}^M$ .  $\mathbf{L}_{\text{norm}}$  can be used instead [30].

Choosing  $K$  is a significant aspect of the SC method, and in fact, various approaches have been proposed in the literature [31,32]. The eigenvalues of  $\mathbf{L}_{\text{sym}}$  can be used to estimate the number of clusters by considering the largest empirical eigengap  $\hat{K} = \arg \max_i \lambda_{i+1} - \lambda_i$ . This heuristic estimate is called the *eigengap* statistic [9].

Basically, we use multiple eigenvectors to embed each data point into a low-dimensional space to preserve the significant difference in normalized similarity. Then, the  $K$ -means algorithm can be used to group the points in the embedding space.  $K$ -means is applied to cluster the  $\{\mathbf{v}_i\}_{i=1}^M \in \mathbb{R}^K$  into  $K$  groups.

The SC approach reflects the key objective of non-hierarchical clustering, consisting of clusters of points with high similarity (intra-clusters) and low similarity between points belonging to different clusters (inter-clusters). In the  $K$ -means algorithm, this target is reached through iterative optimization. SC, using the graph representation, learns the structure of the set of points intrinsically [9]. The choice of  $K$  in SC has been analyzed by many authors in the literature [26,31,32]. We summarize the pseudocode implemented in this study in Algorithm 2.

---

**Algorithm 2** Spectral Clustering with metric  $d$  (Euclidean distance)

---

**Input:**  $\mathcal{T} = \{\mathbf{t}_i\}_{i=1}^M$  (Data)

**Output:**  $Y$  (Labels)

- 1: Select the local scale  $\sigma_i = d(\mathbf{t}_i, \mathbf{t}_{knn})$  where  $\mathbf{t}_{knn}$  is the 7th nearest neighbor of point  $\mathbf{t}_i$ .
  - 2: Compute the affinity matrix  $\mathbf{A} \in \mathbb{R}^{M \times M}$  with  $a_{ij} = \exp\left(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)^2}{\sigma_i \sigma_j}\right)$ .
  - 3: Compute the diagonal degree matrix  $\mathbf{D} \in \mathbb{R}^{M \times M}$  with  $d_{ii} = \sum_{j=1}^M a_{ij}$ .
  - 4: Form the symmetric normalized Laplacian graph  $\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ .
  - 5: Compute the eigendecomposition  $\{(\phi_i, \lambda_i)\}_{i=1}^M$ , sorted so that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$ .
  - 6: Estimate the number of clusters  $K$  as  $\hat{K} = \arg \max_k \lambda_{k+1} - \lambda_k$ .
  - 7: For  $1 \leq i \leq M$ , let  $\mathbf{v}_i = (\phi_1(\mathbf{t}_i), \phi_2(\mathbf{t}_i), \dots, \phi_K(\mathbf{t}_i)) / \|(\phi_1(\mathbf{t}_i), \phi_2(\mathbf{t}_i), \dots, \phi_K(\mathbf{t}_i))\|_2$  define the (row normalized) spectral embedding.
  - 8: Compute labels  $Y$  by running  $K$ -means on the data  $\{\mathbf{v}_i\}_{i=1}^M$  using  $\hat{K}$  as the number of clusters.
-

#### 4.2. Spectral Clustering Variants

In this study, we implemented a standard SC method, the NJW. It is worth notice that different SC methods emerged in the literature, which can be classified into the following three categories.

- **Power iteration (PI)-based methods:** PIC (Power Iteration Clustering) [33], DPIC (Deflation-based Power Iteration Clustering) [34], and DPIE (Diverse Power Iteration Embedding) [34] apply PI (Power Iteration) to generate pseudo-eigenvectors as a replacement of eigenvectors.
- **Multi-scale-data-oriented methods:** ZP and FUSE. The first [26] is a self-tuning SC method. It uses eigenvector rotation to estimate the number of clusters. The second [35] is an SC method based on power iteration and Independent Component Analysis.
- **Matrix-reconstruction methods:** This group of methods constructs a new coefficient matrix based on which new similarity matrix as a replacement of the original one. The main representative of the SC algorithm in this group is ROSC (Robust spectral clustering on multi-scale data) [36], which generates a matrix with a grouping effect.

#### 4.3. Pairwise Constraints in Spectral Clustering

The basic idea of an SC method is to obtain the graph partition by the eigendecomposition of Laplacian graph [14]. The algorithm searches the space of possible organizations of the data, preferring those which group similar instances together and keep dissimilar instances apart. Defining pairwise similarity for an effective SC method is fundamentally challenging, given complex data that are often of high dimension and heterogeneous nature.

Moreover, an SC method is based on matching, and it can easily use the pairwise constraint information provided by practitioners. In most cases, experienced personnel have some prior or background knowledge. How to use prior or background knowledge to improve the cluster quality and promote the efficiency of clustering data has become a research topic in recent years. In this study, we aim to insert supplementary pairwise similarity between samples in the original SC algorithm. The goal is to construct more meaningful affinity graphs for enhanced SC.

Two types of pairwise constraints are considered. The must-link constraints show that two sample points should be embedded in the same cluster. The cannot-link constraints show that two sample points should be divided into different classes. The number of distinct constraints ranges from 1 to  $\frac{1}{2}M(M-1)$ , since constraints are by definition symmetric.

In our study, we considered SC aided by the addition of constraints, which serve to restrict the search space and to guide the search through it. We implemented both must-link and cannot-link constraints. The former constraint specifies that two data instances have to be in the same cluster; the latter constraint specifies that two data instances must not be placed in the same cluster.

Let the relation of must-link constraints (two points have to be in the same cluster) be defined as  $\mathcal{ML} = \{(\mathbf{t}_i, \mathbf{t}_j)\}$ , and the relation of cannot-link constraints (prevent two points being from the same cluster) as  $\mathcal{CL} = \{(\mathbf{t}_i, \mathbf{t}_j)\}$ . Thus, the affinity matrix is modified as follows.

$$a_{ij} = \begin{cases} 1, & \text{if } i \neq j \text{ and } (\mathbf{t}_i, \mathbf{t}_j) \in \mathcal{ML}; \\ 0, & \text{if } i = j \text{ or } (\mathbf{t}_i, \mathbf{t}_j) \in \mathcal{CL}; \\ \exp\left(-\frac{d(\mathbf{t}_i, \mathbf{t}_j)^2}{\sigma_i \sigma_j}\right), & \text{otherwise.} \end{cases} \quad (2)$$

### 5. Internal Clustering Validation Measures

In [37], a comprehensive study of 11 internal validation measures was presented by evaluating their performance on a known dataset. In our study, Caliński-Harabasz (CH), Davies-Bouldin (DB), and Silhouette (S) are the internal validation measures used.



The *CH* index evaluates the cluster validity based on the average between- and within-cluster variance. It can be defined as follows.

$$CH = \frac{M - K}{K - 1} \cdot \frac{SSB}{SSW} \quad (3)$$

where  $M$  is the total number of elements, and  $K$  is the number of clusters chosen in the classification.  $SSB$  and  $SSW$  represent inter-cluster and intra-cluster dispersion. A greater *CH* index shows a better clustering result.

The *DB* index can be defined as follows.

$$DB = \frac{1}{K} \sum_{k=1}^K \max_{k' \neq k} \frac{(\bar{d}_k + \bar{d}_{k'})}{d_{k,k'}} \quad (4)$$

where  $K$  is the number of clusters,  $\bar{d}_k$  is the mean distance between the elements of the  $k$ -th cluster and their respective centroid, similarly for  $\bar{d}_{k'}$ .  $d_{k,k'}$  represents the distance between the centroid of the  $k$ -th and the  $k'$ -th cluster. According to the criteria of compactness and separation, the *DB* parameter must be as small as possible.

Another validation index, which quantifies the compactness and separation between clusters, is *S* (*Silhouette*) index. Let function  $s(\mathbf{t}_i)$  be defined as follows:

$$s(\mathbf{t}_i) = \frac{b(\mathbf{t}_i) - a(\mathbf{t}_i)}{\max(a(\mathbf{t}_i), b(\mathbf{t}_i))} \quad (5)$$

where  $a(\mathbf{t}_i)$  represents the mean distance between the generic point  $\mathbf{t}_i$  and the remaining points assigned to the same cluster;  $a(\mathbf{t}_i)$  measures compactness. The  $b(\mathbf{t}_i)$  element is the smallest mean distance between point  $\mathbf{t}_i$  and residual points assigned to the remaining clusters;  $b(\mathbf{t}_i)$  is an index of separation between clusters. Parameter  $s(\mathbf{t}_i)$  is representative of how much a point  $\mathbf{t}_i$  belongs to the assigned cluster. From the definition of  $s(\mathbf{t}_i)$ , valid for the single point, it is possible to define the global *Silhouette* index  $S$ :

$$S = \frac{1}{K} \sum_{k=1}^K \frac{1}{N_k} \sum_{\mathbf{t}_i \in C_k} s(\mathbf{t}_i) \quad (6)$$

where  $K$  is the number of clusters,  $N_k$  is the number of elements assigned to the  $k$ -th cluster. Higher is  $S$  (at most tending to 1), better is the corresponding clustering solution.

## 6. Fault Diagnosis of an Injection Control System

In the development of a modern diesel engine, numerous technologies are employed to reduce fuel consumption and the emission of pollutants into the environment. Two examples are the selective catalytic reduction system and the high-pressure common-rail (HPCR) system. In particular, the HPCR is a fuel-injection system equipped with a storage chamber, in which fuel is stored under pressure, and a rail pipe, which provides fuel to the injectors. By the HPCR, the Engine Control Unit (ECU) regulates and optimizes the combustion process in a very accurate manner.

The adequate operation of the HPCR is guaranteed by electronically controlling most of its subcomponents through triggers modulated by the ECU. These signals are the result of control logic obtained by comparing measurements recorded by sensors and calibrated thresholds. By using electronic regulation, the injection pressure can be adjusted according to both the rotational speed of the engine and the torque demands of the driver (through the accelerator pedal).

Fuel injection into a cylinder is possible thanks to accurate injectors [17]. Nozzle opening occurs indirectly by perturbing the balance of hydraulic forces upstream of the needle. Using a high-pressure gradient, the energizing of a solenoid valve allows refueling of the fuel through calibrated holes, resulting in a dragging effect by lifting the needle. When the energizing of the solenoid valve coil stops, the hydraulic state is restored. Next, the initial equilibrium of forces along the injector valve rod is re-established. The result coincides with the needle falling and the nozzle closing.

Figure 3 draws a general layout schema of an HPCR system. It mainly consists of a pipe with fixing flanges. Internal rail volume is accessible through a tube to the high-pressure pump and pressure lines connecting the injectors in parallel. To obtain the desired injection pressure, both injection starting time and its duration must be electrically actuated by triggers released by the ECU. An example of a solenoid injector is reported in Figure 4. It can be observed the electric contacts for the solenoid coil (top), which receive triggers from the ECU, and the high-pressure connector (middle), which joins the injector to the HPCR system.

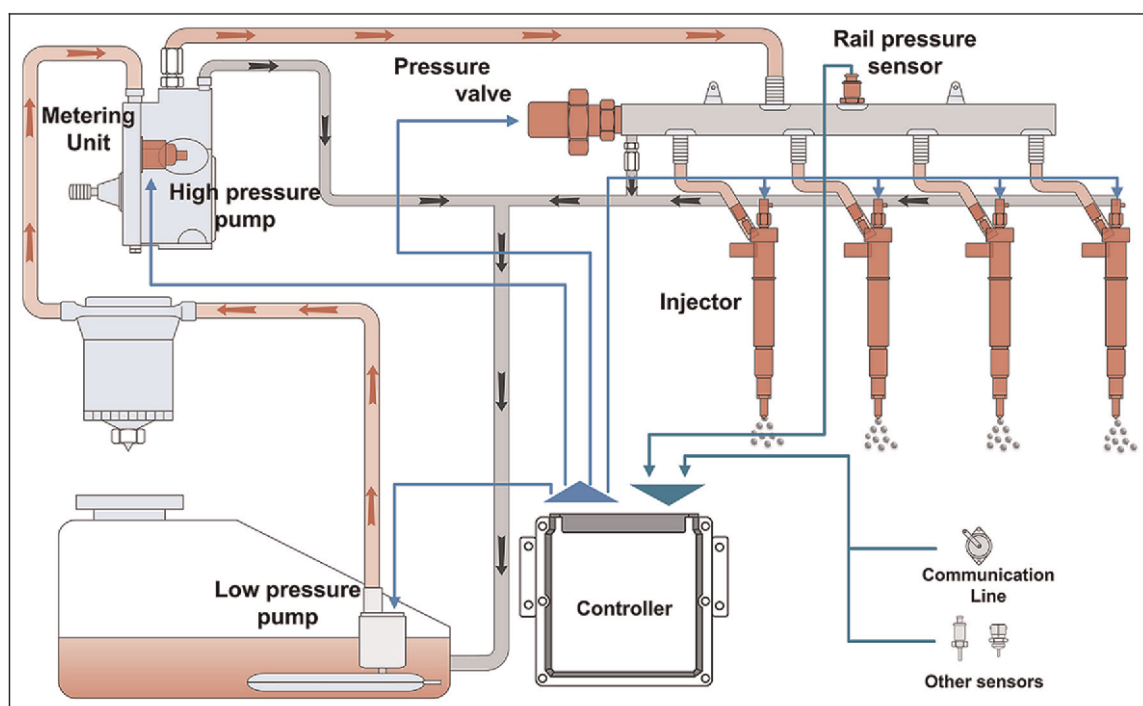
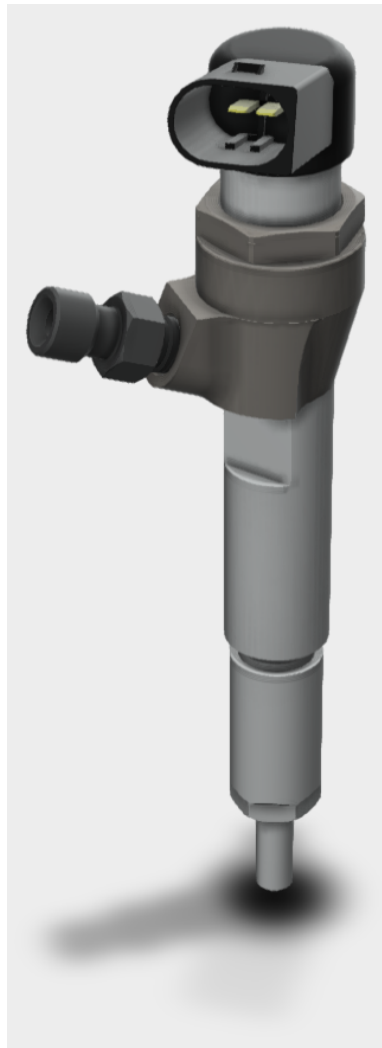
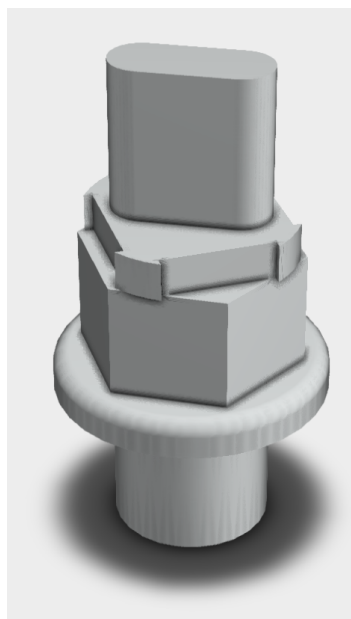


Figure 3. Layout schema of high-pressure common-rail (HPCR) system.

In an HPCR system, it is essential to maintain the stability of the injection pressure and reduce the difference in the fuel-injection amount caused by different injectors in the system. Compression and rarefaction waves inside the rail may be caused by suddenly fuel acceleration and deceleration with the result of degrading the injection precision [38]. Under certain load conditions, a fault may also occur through a flash opening of the injector causes an unspecified pressure drop in the rail. The diagnostic system reads these events as the repeated opening of a worn-out pressure relief valve (Pressure Relief Valve—PRV). A PRV for an HPCR system incorporates a ceramic spherical valve element that moves into and out of contact with a conical valve seat of a metallic valve body. This safety component is equipped on some HPCR systems, particularly heavy-duty diesel engines, to prevent high pressures in the rail and fuel to flow back. This avoids a potentially dangerous condition for the diesel engine [18]. An example of an HPCR pressure valve system is depicted in Figure 5.



**Figure 4.** HPCR solenoid injector.



**Figure 5.** HPCR pressure valve.

### 6.1. Multisensor Dataset

To get a deeper insight into the targeted fault and to explain the causes of the anomalous injection events, experiments were carried out on a six-cylinder, four-stroke, turbocharged, heavy-duty diesel engine equipped with an HPCR fuel-injection system. The injection system is constituted of one high-pressure fuel pump, one common-rail pipe, and six injectors. The high-pressure fuel is delivered from the pump to the common-rail and finally to the injector in each cylinder of the engine.

The acquisition of process data was made possible by a memory emulator module associated with the ECU employed to increase its storage capacity. The total number of sensors is 34. The list of sensors is reported in Table 1.

**Table 1.** List of channels. For each channel is reported the description, unit of measure and processing label.

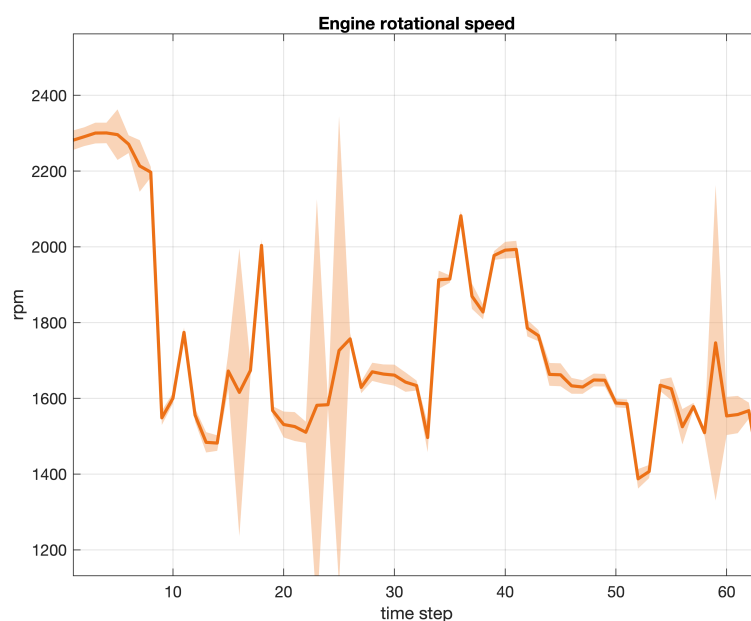
Channel Name	Unit of Measure	Label
Average engine speed	[rpm]	1
Vehicle speed	[km/h]	2
Air mass flow	[kg/h]	3
Current injection quantity	[mg/hub]	4
Metering Unit actual current	[mA]	5
Battery voltage	[mV]	6
Downstream engine coolant temperature	[°C]	7
Measured rail pressure gradients (new)	[hPa]	8
Measured rail pressure gradients (old)	[hPa]	9
Rail pressure set point	[hPa]	10
Rail pressure filtered	[hPa]	11
Inner torque set value	[Nm]	12
Set point volume flow of RP governing	[mm <sup>3</sup> /s]	13
Volume flow requested overall for all injection system	[mm <sup>3</sup> /s]	14
Volume flow requested for all injections	[mm <sup>3</sup> /s]	15
1st–6th Cylinder individual correction quantity	[mg/hub]	16–21
Injection time (main)	[μs]	22
Injection time (pilot)	[μs]	23
Desired main injection quantity	[mg/hub]	24
Desired pilot injection quantity	[mg/hub]	25
Quantity correction for limiting pressure variations	[mg/hub]	26
Actual fuel supply low pressure	[bar]	27
Value of fuel temperature	[°C]	28
Sum of requested injection quantity per cyl.	[mg/hub]	29
Sum of all estimated injection control quantities	[mg/hub]	30
Torque generating engine fuel-injection quantity	[mg/hub]	31
Status of fuel balancing control loop	[-]	32
State variable of PRV	[-]	33
PRV opening detection	[-]	34

Collected measurements were recorded by sensors placed on the vehicle, while a few of them were related to actuator signals generated by the ECU during the injection process. Only two channels were related to the on-board diagnostic system of the PRV (labels 33 and 34). Channels were related to all principal variables related to the HPCR injection process. In our study, data measured in the post-treatment system were not considered because, given the latency of the exhaust flow, these variables resulted shifted in time from the instant of the detected fault event.

The targeted fault is an unusual opening of the injector, which leaks a quantity of fuel that causes pressure drops in the rail. When these fluctuations are significant, the diagnostic system interprets the phenomenon inaccurately. On-board diagnostic releases an alarm concerning the opening of the PRV. Although it allows the fuel backflow only in situations of pressure overshoot, PRV is a passive safety component, implying that it is neither equipped with sensors nor can be controlled for opening. Any wear resulting in leakage is evaluated indirectly through a deviation of the rail

pressure signal. When the magnitude of the fault is not sufficient to trigger the warning, it is difficult to discriminate this event from regular injections. Both scenarios could lead to a similar rate of rail pressure reduction. Therefore, evaluating only rail pressure profile variability is ineffective. To understand the progression of the injection process before and after fault events, it is important to evaluate all the sensor measurements.

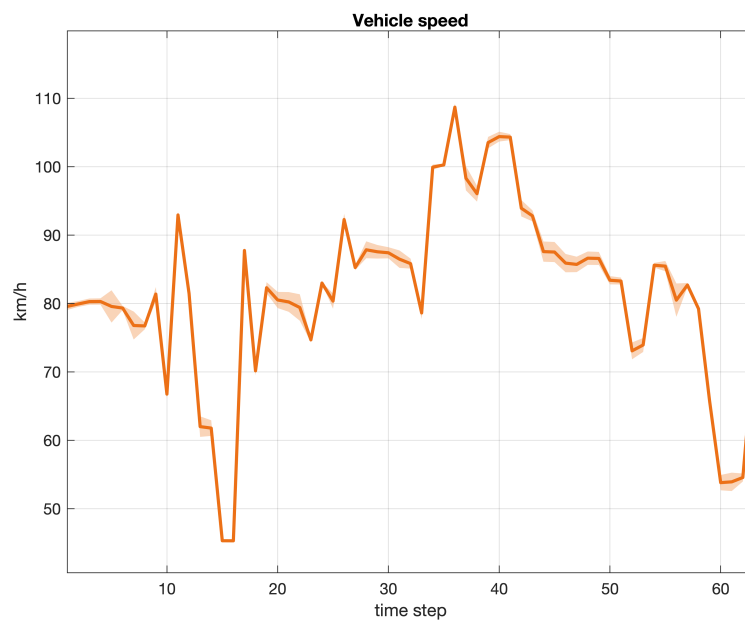
Three examples of signals collected during our experiments are depicted in subsequent Figures 6–8. In particular, each figure describes the average value of the signal, with a continued bold line, including the area of variability computed as the 3-sigma interval from the mean value for each time step. The total number of samples considered in each graph is equal to 203. Figure 6 refers to the engine rotational speed (label 1 in Table 1), Figure 7 to the vehicle speed (label 2 in Table 1), while Figure 8 to the inner torque set value (label 12 in Table 1). Every signal recorded by a sensor in a given time window was linearly scaled in the range  $[0, 1]$  and centered by subtracting the average profile of the relative variable.



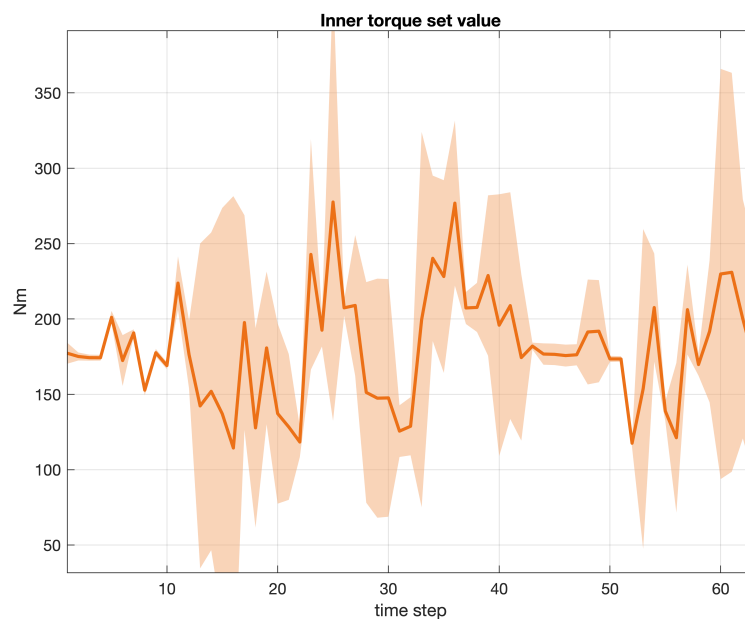
**Figure 6.** Engine rotational speed (label 1). Average value (continued bold line) and 3-sigma area plot for 203 samples.

The final dataset for fault diagnosis was obtained by considering the series of Diagnostic Trouble Codes (DTCs) triggered by the ECU corresponding to the opening of the PRV. For each scaled and centered signal associated with a specific sensor, the values related to instants of DTCs produced by the ECU were collected. The total number of such events in the monitored window (number of DTCs released by the ECU) was equal to 1101. Therefore, these operations produced a dataset on the form of a matrix  $\mathbf{X}_{c,[0,1]} \in \mathbb{R}^{P \times M}$  with  $M = 1101$  observations and  $P = 34$  process measurements, subsequently processed by PCA to extract relevant features and to reduce the dimensionality of multisensor process data. Specifically, 7 PCs were derived from the original dataset of data from 34 sensors. To extract the feature and to choose the number of PCs, conventional cross-validation statistical techniques were implemented on data [39]. Such  $N = 7$  PCs correspond to about 90% of explained variability in the data, while the first 4 PCs only correspond to about 85% of explained variability in the data.





**Figure 7.** Vehicle speed (label 2). Average value (continued bold line) and 3-sigma area plot for 203 samples.



**Figure 8.** Inner torque set value (label 12). Average value (continued bold line) and 3-sigma area plot for 203 samples.

## 6.2. Clustering

Cluster analysis aims to arrange observations considered similar to reveal patterns that support the investigation on the targeted fault (the pressure drop in the injection process). This assists the practitioner in the root-cause analysis by highlighting the set of components to which the fault can be ascribed.

Three clustering methods in our case study: (i) the *K*-means++, (ii) the original NJW SC, and (iii) the proposed NJW SC with pairwise constraints. The three methods were applied to the dataset of  $N = 7$  scores obtained from PCA.

In *K*-means++ clustering, defining the optimal number of clusters was handled by *Kneedle*. It consists of minimizing the intra-cluster variance of *SSW*. The results of the *Kneedle* algorithm applied to the case study dataset are depicted in Figure 9. The resulting number of optimal clusters is equal to  $K = 3$ . To validate the classification results, the Caliński-Harabasz (*CH* in Equation (3)), Davies-Bouldin (*DB* in Equation (4)), and Silhouette (*S* in Equation (5)) indices were computed. Figure 10 shows that the value of  $K = 3$  appears to be optimal according to *DB* and *S* indexes, although the *CH* exhibits a slightly greater value for  $K = 5$  clusters.

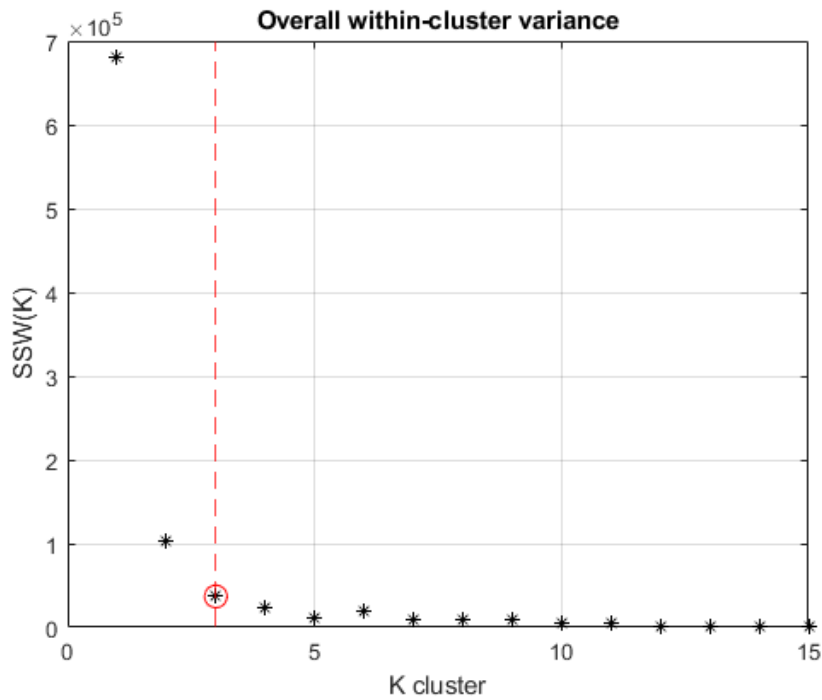


Figure 9. Kneedle algorithm results applied to *K*-means++ method.

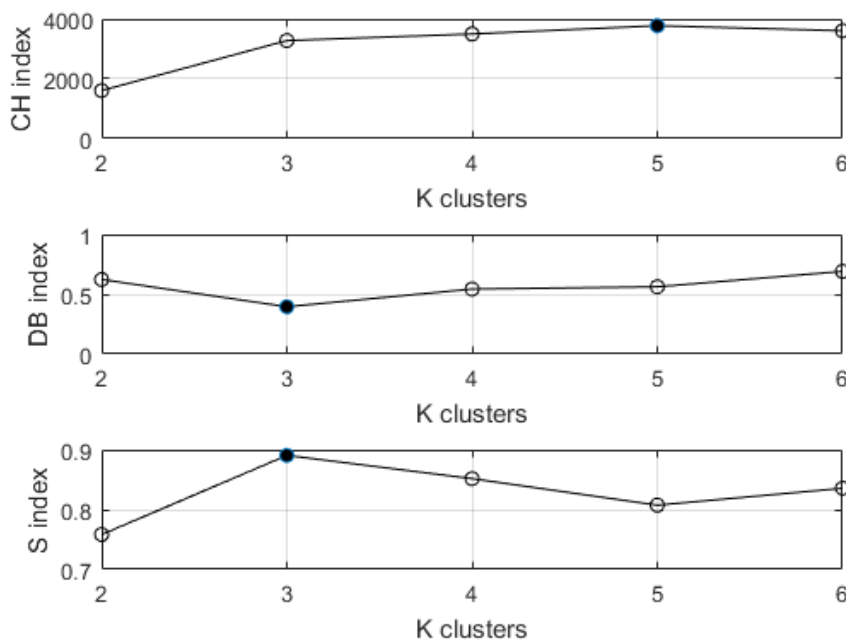
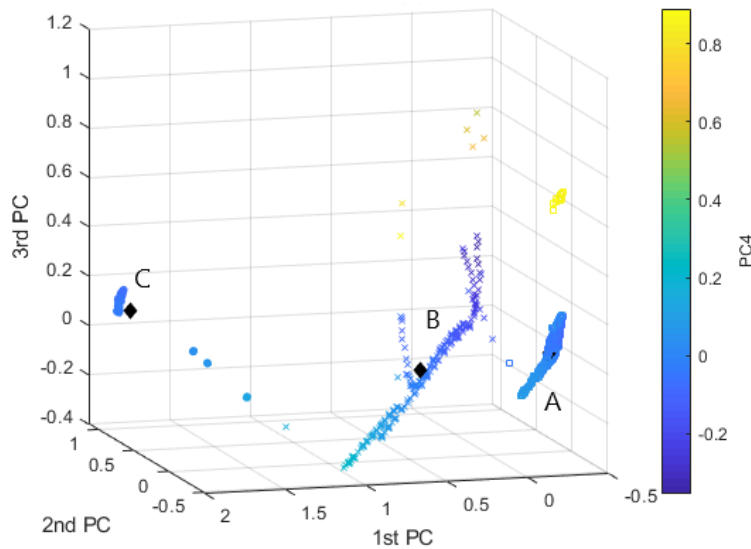


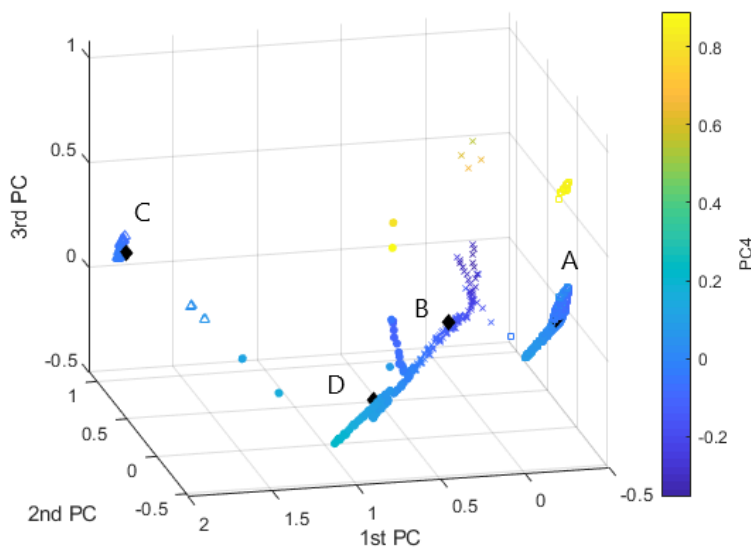
Figure 10. Internal validation indexes of the performed clustering.

Figure 11 displays the  $K = 3$  clusters obtained by  $K$ -means++, in a three-axis diagram, where each axis is related to the first 3 PC. A color scale is used to express the value of the 4th PC (the first 4 PCs describe 85% of variability). From Figure 11 it can be noticed that several outliers (points distant from the relative centroid of the cluster) are assigned to clusters A (square graphical symbol) and B (x graphical symbol) and are characterized by a high value of the 4th PC.



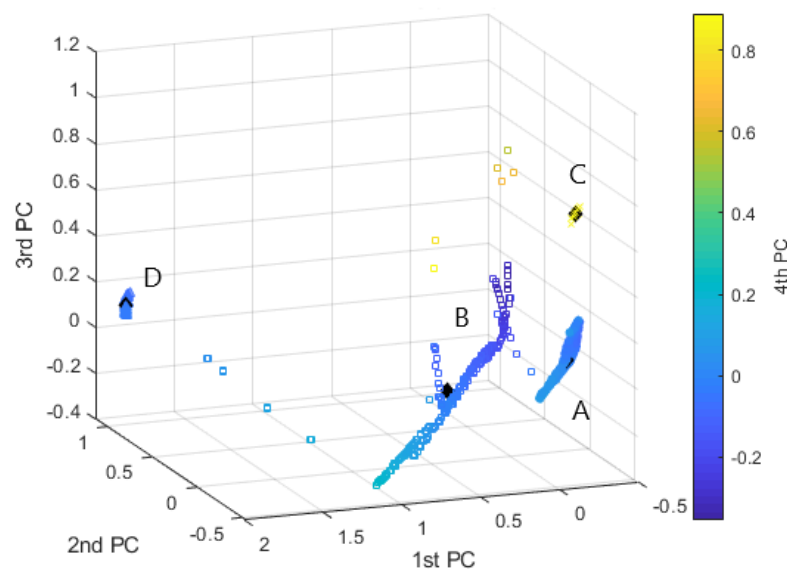
**Figure 11.**  $K$ -means,  $K = 3$ . Clustering representation in the principal component scatterplot, Clust.A—"□", Clust.B—"x", Clust.C—"o".

We implemented the SC method on the same multisensor dataset with  $L_{\text{sym}}$  and constructed the spectral embedding according to the NJW algorithm [8]. The eigenvalues of  $L_{\text{sym}}$  were estimated as the largest empirical eigengap  $\hat{K} = \arg \max_i \lambda_{i+1} - \lambda_i$ , which resulted in the value  $\hat{K} = 4$ . By applying SC to all scores, it is possible to determine the classification as in Figure 12. It can be observed that SC results in splitting one of the clusters into two different partitions.



**Figure 12.** SC method. Clustering representation in the principal components scatterplot: Clust.A—"□", Clust.B—"x", Clust.C—"△", Clust.D—"o".

Ultimately, we implemented the SC method by adding 7 must-link constraints and 8 cannot-link constraints, which were obtained from experienced personnel for 15 specific events out of the 1101 faults collected, which represents a small fraction of the dataset. From the results graphically depicted in Figure 13 it can be observed that the SC approach augmented with pairwise constraints can accurately partition the set of data by isolating the points characterized by a high value of the 4th PC and that is distantly positioned from the centroid of the dense clusters.



**Figure 13.** SC method with pairwise constraints. Clustering representation in the principal components scatterplot: Clust.A—"o", Clust.B—"□", Clust.C—"x", Clust.D—"△" (SC,  $K = 4$ ).

To confirm the results obtained by the original SC NJW algorithm and the SC NJW augmented with pairwise constraints, the validation indexes were computed and are reported in Table 2. Bold font represents outperforming results. It can be noted that while *CH* presents a decreasing performance level of around 2%, the *DB* and *S* indexes show an improved performance of the SC NJW augmented with pairwise constraints of about 16% and 4% when compared to the original SC.

**Table 2.** Internal validation measures for spectral clustering.

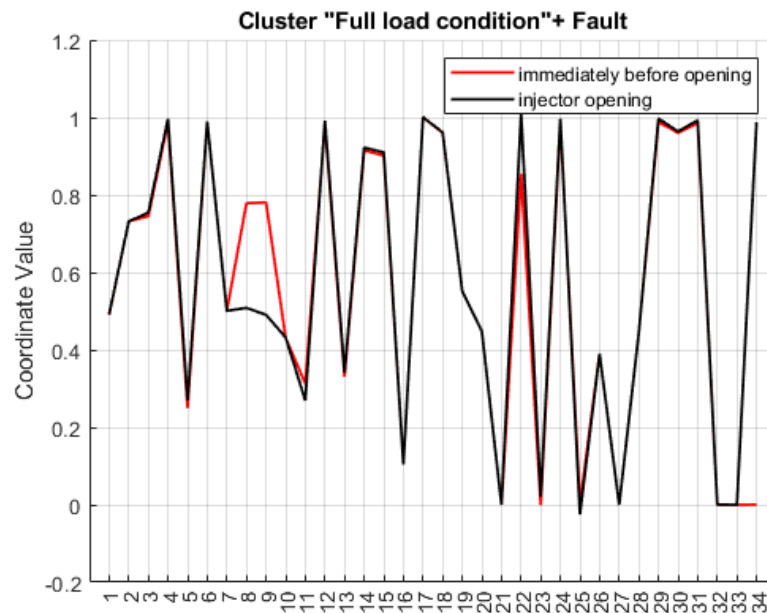
Index	Original SC	SC with Pairwise Constraints
CH	2495	2438.3
DB	0.4162	<b>0.3481</b>
S	0.8528	<b>0.8840</b>

### 6.3. Cluster Evaluation of Fault Scenario

To evaluate the content of a cluster and to label the classes they represent, the parallel coordinate plot [40] of cluster centroid of interest was examined. The cluster centroid is computed as the barycentre of a discrete points distribution and hence of the cluster. In this regard, a centroid can be considered the most representative point of a cluster if that is compact and dense, as for clusters *A* and *C* obtained by SC (Figure 13).

Both clusters *A* and *C* describe conditions under which the torque (label 4 in Table 1) of the engine is maximum, as well as all related profiles (injected quantity of fuel); for this reason, clusters *A* and *C* are labeled "Full load". The comparison, obtained by superimposing the combination of centroid coordinates, allows us to emphasize the fault, the operating conditions related to this fault event, and above all, the progression of the process.

The analysis of Figure 14 reveals that the coordinates of centroids are similar except for the variables with labels No. 8 and No. 9 related to the rail pressure gradient (Table 1). The pressure drop, and therefore the fault, is imputable to a malfunction of the injector.



**Figure 14.** Comparison between cluster A (red) and cluster C (black) centroid coordinates. SC with pairwise constrains.

From Figure 14, it is clear that the scenario demonstrated by the two clusters is the same and matches the condition of maximum torque demand. Activation of the last variable No. 34 (in cluster C) is associated with a reduction in the pressure gradient monitored by variables No. 7 and No. 8. Furthermore, differences are present in channel No. 22, the energizing time of the injector: this is attributable to the dependence of this variable on the rail pressure, which is not capable of following the set point. This gap is also shown by the slight differences between the values of channel No. 10 and No. 11.

As a result, cluster C presents the specific problem of the common rail system under investigation, namely the pressure loss of the fuel that is not attributable to normal operation. In the extracted pattern, the filtered pressure is lower than the set pressure at a consistent rate. Since the rail pressure regulation is in a closed loop, the pressure drop event connected to the injector fault occurs so rapidly that the system does not compensate for the deviation immediately. Variable No. 22 represents the energizing time during the main injection, and it can be noticed that it increases in the fault scenario. The pressure drop, and therefore the fault, is clearly due to a malfunction of the injector.

## 7. Conclusions

Clustering algorithms, which group similar features into the same cluster and separate dissimilar features into different ones, are common analysis methods for unlabeled data. The clustering phase is an essential aspect of the analysis of multisensor data. In this study, to have a better insight of the faults in the injection process, clustering has been applied to multisensor measurements obtained by experiments on a real diesel engine, equipped with an HPCR system and electronically controlled by an ECU. Clustering, exploiting the compactness of the space constituted, has contributed to identifying different scenarios allowing us to diagnose the root causes of the targeted fault in different operating areas of the engine.



The most widely used clustering algorithm, *K*-means, although distinguishing such zones, has failed to identify a fault scenario. Using this classification, clusters are misconstrued because the *K*-means is sensitive to non-spherical structures of data, altering the cluster centroid computation, which supports fault diagnosis. In the case study presented in this paper, SC provides the advantage of an aggregation criterion more robust to non-spherical structures of data. In this paper, a semi-supervised approach has also been discussed to combine labeled and unlabeled measurement data in SC modeling. A class of fault has been identified within the resulting groups of clusters, contributing to a comprehensive understanding of the phenomenon.

In this study, the PCA was implemented as a dimensional reduction procedure to improve clustering by decreasing the dimension of the measurement dataset while capturing the linear correlation structure of the data. The PCA replaces the measurements with a smaller number of points that are a linear combination of original data and considers these new points as the scalar variables. However, this approach may mask the effect of each sensor by merging them into new ones and fails to exploit the ordering structure of a variable. In the situations in which many of the inputs are not informative, the extracted features may become diluted. A direction of future research includes replacing the PCA step with a variable screening phase for fault diagnosis. This method should perform variable selection to distinguish which inputs are most informative in the original measurement domain, i.e., a method capable of sensor screening by selecting the most informative variable inputs. A recent example in the literature of a statistical method to perform sensor screening, and to generate predictions, is reported in [41] concerning a case study related to the monitoring of an internal combustion engine through a large number of sensor signals.

Considering the experimental approach of this work, clustering techniques are showed to be the main tool for assisting fault diagnosis in modern applications. Facilitating fault examination procedures would contribute to improving root-cause analysis, bypassing the need for a deep knowledge of the process, and so for the support of experienced personnel.

**Author Contributions:** M.P.: conceptualization, methodology, software, writing. G.P.: formal analysis, validation, data curation, visualization. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## References

1. Pacella, M. Unsupervised classification of multichannel profile data using PCA: An application to an emission control system. *Comput. Ind. Eng.* **2018**, *122*, 161–169. [[CrossRef](#)]
2. Gomez, A.; Paynabar, K.; Pacella, M. Functional directed graphical models and applications in root-cause analysis and diagnosis. *J. Qual. Technol.* **2020**. [[CrossRef](#)]
3. Isermann, R. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
4. Alshorman, O.; Irfan, M.; Saad, N.; Zhen, D.; Haider, N.; Glowacz, A.; Alshorman, A. A Review of Artificial Intelligence Methods for Condition Monitoring and Fault Diagnosis of Rolling Element Bearings for Induction Motor. *Shock Vib.* **2020**, *2020*, 8843759. [[CrossRef](#)]
5. Qian, P.; Jiang, Y.; Wang, S.; Su, K.H.; Wang, J.; Hu, L.; Muzic, R. Affinity and Penalty Jointly Constrained Spectral Clustering with All-Compatibility, Flexibility, and Robustness. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1123–1138. [[CrossRef](#)]
6. Bradley, P.S.; Fayyad, U.M.; Mangasarian, O.L. Mathematical programming for data mining: Formulations and challenges. *INFORMS J. Comput.* **1999**, *11*, 217–238. [[CrossRef](#)]
7. Witten, D.M.; Tibshirani, R. A framework for feature selection in clustering. *J. Am. Stat. Assoc.* **2010**, *105*, 713–726. [[CrossRef](#)]
8. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In Proceedings of the Neural Information Processing Systems: Natural and Synthetic, Vancouver, BC, Canada, 3–8 December 2002.

9. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [[CrossRef](#)]
10. Soltanolkotabi, M.; Elhamifar, E.; Candes, E. Robust subspace clustering. *Ann. Stat.* **2014**, *42*, 669–699. [[CrossRef](#)]
11. Mixon, D.; Villar, S.; Ward, R. Clustering subgaussian mixtures by semidefinite programming. *Inf. Inference* **2017**, *6*, 389–415. [[CrossRef](#)]
12. Arias-Castro, E.; Lerman, G.; Zhang, T. Spectral clustering based on local PCA. *J. Mach. Learn. Res.* **2017**, *18*, 1–57.
13. Zhang, X.; You, Q. An improved spectral clustering algorithm based on random walk. *Front. Comput. Sci. China* **2011**, *5*, 268–278. [[CrossRef](#)]
14. Jia, H.; Ding, S.; Du, M. Self-Tuning p-Spectral Clustering Based on Shared Nearest Neighbors. *Cogn. Comput.* **2015**, *7*, 622–632. [[CrossRef](#)]
15. Langone, R.; Suykens, J. Fast kernel spectral clustering. *Neurocomputing* **2017**, *268*, 27–33. [[CrossRef](#)]
16. Affeldt, S.; Labiod, L.; Nadif, M. Spectral clustering via ensemble deep autoencoder learning (SC-EDAE). *Pattern Recognit.* **2020**, *108*. [[CrossRef](#)]
17. Reif, K. *Diesel Engine Management*; Springer: Cham, Switzerland, 2014.
18. Wright, G. *Fundamentals of Medium/Heavy Duty Diesel Engines*; Jones & Bartlett Publishers: Burlington, MA, USA, 2015.
19. Grasso, M.; Colosimo, B. Process defects and in situ monitoring methods in metal powder bed fusion: A review. *Meas. Sci. Technol.* **2017**, *28*. [[CrossRef](#)]
20. Glowacz, A.; Glowacz, W.; Kozik, J.; Piech, K.; Gutten, M.; Caesarendra, W.; Liu, H.; Brumercik, F.; Irfan, M.; Faizal Khan, Z. Detection of Deterioration of Three-phase Induction Motor using Vibration Signals. *Meas. Sci. Rev.* **2019**, *19*, 241–249. [[CrossRef](#)]
21. Li, J.; Yao, X.; Wang, X.; Yu, Q.; Zhang, Y. Multiscale local features learning based on BP neural network for rolling bearing intelligent fault diagnosis. *Meas. J. Int. Meas. Confed.* **2020**, *153*, 107419. [[CrossRef](#)]
22. Patel, S.; Upadhyay, S. Euclidean distance based feature ranking and subset selection for bearing fault diagnosis. *Expert Syst. Appl.* **2020**, *154*, 113400. [[CrossRef](#)]
23. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [[CrossRef](#)]
24. Arthur, D.; Vassilvitskii, S. k-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
25. Satopaa, V.; Albrecht, J.; Irwin, D.; Raghavan, B. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, USA, 20–24 June 2011; pp. 166–171.
26. Zelnik-Manor, L.; Perona, P. Self-tuning spectral clustering. In Proceedings of the 17th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 13–18 December 2004; pp. 1601–1608.
27. Wang, J.; Chang, S.F.; Zhou, X.; Wong, S. Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. In Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
28. Pavan, M.; Pelillo, M. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 167–172. [[CrossRef](#)]
29. Premachandran, V.; Kakarala, R. Consensus of k-NNs for robust neighborhood selection on graph-based manifolds. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1594–1601.
30. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
31. Azran, A.; Ghahramani, Z. Spectral methods for automatic multiscale data clustering. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 1, pp. 190–197.
32. Sanguinetti, G.; Laidler, J.; Lawrence, N.D. Automatic determination of the number of clusters using spectral algorithms. In Proceedings of the 2005 IEEE Workshop on Machine Learning for Signal Processing, Mystic, CT, USA, 28–30 September 2005; pp. 55–60.

33. Lin, F.; Cohen, W. Power iteration clustering. In Proceedings of the ICML 2010 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 655–662.
34. The, A.; Thang, N.; Vinh, L.; Lee, Y.K.; Lee, S. Deflation-based power iteration clustering. *Appl. Intell.* **2013**, *39*, 367–385. [[CrossRef](#)]
35. Ye, W.; Goebel, S.; Plant, C.; Bohm, C. FUSE: Full spectral clustering. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1985–1994.
36. Li, X.; Kao, B.; Luo, S.; Ester, M. ROSC: Robust spectral clustering on multi-scale data. In Proceedings of the WWW '18: Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 157–166.
37. Liu, Y.; Li, Z.; Xiong, H.; Gao, X.; Wu, J. Understanding of internal clustering validation measures. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 13–17 December 2010; pp. 911–916.
38. Ferrari, A.; Paolicelli, F. An indirect method for the real-time evaluation of the fuel mass injected in small injections in Common Rail diesel engines. *Fuel* **2017**, *191*, 322–329. [[CrossRef](#)]
39. Eastment, H.; Krzanowski, W. Cross-validators choice of the number of components from a principal component analysis. *Technometrics* **1982**, *24*, 73–77. [[CrossRef](#)]
40. Heinrich, J.; Weiskopf, D. State of the Art of Parallel Coordinates. In Proceedings of the Eurographics (STARs), Girona, Spain, 6–10 May 2013; pp. 95–116.
41. Gahrooei, M.; Paynabar, K.; Pacella, M.; Shi, J. Process Modeling and Prediction with Large Number of High-Dimensional Variables Using Functional Regression. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 684–696. [[CrossRef](#)]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).