





Article

Performance Analysis and Architecture of a Clustering Hybrid Algorithm Called FA+GA-DBSCAN Using Artificial Datasets

Juan Carlos Perafan-Lopez ^{1,*} , Valeria Lucía Ferrer-Gregory ² , César Nieto-Londoño ³ 
and Julián Sierra-Pérez ¹ 

- ¹ Grupo de Investigación en Ingeniería Aeroespacial, Universidad Pontificia Bolivariana, Medellín 050031, Colombia; julian.sierra@upb.edu.co
² Semillero de Investigación en Ingeniería Aeroespacial, Universidad Pontificia Bolivariana, Medellín 050031, Colombia; valeria.ferrer@upb.edu.co
³ Grupo de Investigación en Energía y Termodinámica, Universidad Pontificia Bolivariana, Medellín 050031, Colombia; cesar.nieto@upb.edu.co
* Correspondence: juan.perafan@upb.edu.co

Abstract: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a widely used algorithm for exploratory clustering applications. Despite the DBSCAN algorithm being considered an unsupervised pattern recognition method, it has two parameters that must be tuned prior to the clustering process in order to reduce uncertainties, the minimum number of points in a clustering segmentation MinPts, and the radii around selected points from a specific dataset Eps. This article presents the performance of a clustering hybrid algorithm for automatically grouping datasets into a two-dimensional space using the well-known algorithm DBSCAN. Here, the function nearest neighbor and a genetic algorithm were used for the automation of parameters MinPts and Eps. Furthermore, the Factor Analysis (FA) method was defined for pre-processing through a dimensionality reduction of high-dimensional datasets with dimensions greater than two. Finally, the performance of the clustering algorithm called FA+GA-DBSCAN was evaluated using artificial datasets. In addition, the precision and Entropy of the clustering hybrid algorithm were measured, which showed there was less probability of error in clustering the most condensed datasets.

Keywords: clustering; DBSCAN; factor analysis; genetic algorithm; pattern recognition; entropy



Citation: Perafan-Lopez, J.C.; Ferrer-Gregory, V.L.; Nieto-Londoño, C.; Sierra-Pérez, J. Performance Analysis and Architecture of a Clustering Hybrid Algorithm Called FA+GA-DBSCAN Using Artificial Datasets. *Entropy* **2022**, *24*, 875. <https://doi.org/10.3390/e24070875>

Academic Editors: Diego Oliva and Salvador Miguel Hinojosa Cervantes

Received: 20 March 2022

Accepted: 12 June 2022

Published: 25 June 2022

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



Copyright: © 2022 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

Data classification has recently become an essential activity in solving and handling problems in which large datasets are involved. When used as a tool for data classification, pattern recognition algorithms may have a fundamental implication in the decision-making process. There are two prominent methods defined for data classification: supervised classification, where the number of groups has to be defined before classification, and unsupervised classification, in which it is expected that the algorithm performs the clustering analysis by itself without requiring a previous setup of the parameters. Density clustering techniques are a subgroup of unsupervised pattern recognition algorithms. These algorithms involve methodologies to identify a particular density in the space of points from a specific dataset. Therefore, density interrelated elements will be included in a particular cluster. As a result, density-based clustering algorithms can determine clusters with a significant diversity of shapes and discriminate meaningful information from outliers [1,2].

Density-Based Spatial Clustering of Application with Noise (DBSCAN) is a density-based unsupervised classification method developed by Ester et al. and presented in 1996 [3]; however, it is still recognized as a helpful method due to its simplicity and good overall performance [4]. Density-based algorithms such as DBSCAN, Clustering Large Applications based on Randomized Search (CLARANS) [5], and Ordering Points To Identify Clustering Structure (OPTICS) [6], among others [7,8], are used for exploratory analysis,

where classes belonging to a specific dataset D are not completely defined or the set is by nature randomized. Furthermore, unlike popular clustering algorithms, such as K-means, DBSCAN does not require limiting the number of clusters or classes previously.

The DBSCAN algorithm is not entirely automatized, and defining two input parameters that depend on the analyzed dataset D is thus needed to perform the clustering process. These input parameters are Eps and MinPts, which depend on the density and magnitude of the specific dataset D being examined. Eps is the radius of an imaginary circumference where a minimum set of points is reachable when using the Euclidean distance defined by the parameter MinPts. Although DBSCAN was first introduced in 1996, the algorithm is still widely used today, being awarded the SIGKDD Test-of-Time Award in 2014. Its relevance is associated with its ease of implementation and reasonable computational cost, $O(n^3)$, when used in large datasets [4]. Nevertheless, the algorithm may achieve an acceptable computational cost, as well as its precision will vastly depend on the selection of parameters Eps and MinPts, since these parameters must be adjusted according to the specific dataset being analyzed.

The DBSCAN algorithm measures distance from point to point using the well-known Euclidean distance, i.e., DBSCAN could perform a clustering analysis in dimensional spaces greater than two. Even though the Euclidean distance can be measured in an n -dimensional space, the clustering process entails a lower computational complexity cost when performed in a two-dimensional space. Consequently, if the dataset being analyzed has a dimension higher than two, it is deemed necessary to evaluate the use of a strategy for dimensional reduction as part of pre-processing, such as Principal Component Analysis (PCA) or the Factor Analysis (FA) method. Datasets can be handled before clustering using a dimensionality reduction method. Although a fraction of information is lost, benefits can be expected in the overall clustering process by reducing computational costs.

Many variations of the DBSCAN algorithm have been developed to obtain a highly autonomous and precise algorithm with the lowest possible computational cost. In addition, different performance metrics can be used as algorithm evaluation metrics to improve their performance. BIRCHSCAN is an algorithm presented by de Moura [9], where the BIRCH algorithm was merged with DBSCAN as a strategy for significant dataset clustering. The CF-Three method and a threshold were determined for the Eps parameter selection. The BIRCH algorithm is defined to evaluate the dataset to select a smaller representative biased sub-dataset, which is evaluated using DBSCAN. The evaluation metrics selected for this methodology are the Rand Index and the Adjusted Rand Index.

Lai et al. [10] presented a method based on Multi-Verse Optimization (MVO) to improve the selection of DBSCAN parameters Eps and MinPts using the r rates in the Accuracy of artificial datasets. In the study proposed by Wang et al. [11], a method for automatic estimation of the DBSCAN parameter Eps was defined for LiDAR data segmentation clustering. The estimation of the parameter Eps was based on the average value in the population defined by the nearest neighbor function. The accuracy of the results was estimated using reference data.

The paper presented by Darong and Peng [12] combined a grid partition technique with DBSCAN, calling the methodology GRPDBSCAN. The strategy implies partitioning the information on grids and then finding the suitable DBSCAN parameters considering the information contained in each partition. Although the authors emphasized the algorithm's precision, it was not clear how the clustering performance was measured in this work. Ohadi et al. defined a new DBSCAN algorithm called SW-DBSCAN [13] formulated on the sliding window grid-based model [14]. Nevertheless, in this paper, the evaluation of the algorithm was measured using the Accuracy metric. The algorithm BDE-DBSCAN proposed by Karami and Johansson [15] presents a methodology for automatic DBSCAN parameter definition using a hybrid optimization method called Binary Differential Evolution. An analytical process and the Tournament Selection (TS) technique were selected for Eps estimation. The performance of the algorithm was defined using the effectiveness metric.

The work presented by Kumar and Reddy [16] adopted a methodology based on structures associated with specific groups that accelerate the neighborhood search queries. As a result, the clustering technique increased DBSCAN's clustering performance by 2.2. This method is called G-DBSCAN, an accelerated DBSCAN algorithm that aims to find the nearest neighbor with the help of group methods. In short, the algorithm works by applying grouping partition methods to identify subgroups with similar patterns in a specific dataset D , which is followed by a dimensional reduction method and the definition of the parameter Eps for each group.

Zhu et al. [17] defined a methodology for an adaptive Eps parameter estimation implementing a Gauss kernel density method considering the clustering of unbalanced artificial datasets. Clustering performance was evaluated using the Rand Index and V-measure. A novel algorithm was presented in [18]; this algorithm, called K-DBSCAN, is considered as an optimization algorithm called Harmony Search (HS), which designates the proper value of the clustering parameters. Here, the cluster number K is predefined by a partition clustering approach. The HS algorithm defines the optimal value for the DBSCAN parameters. The Rand Index and Jaccard coefficient are the evaluation metrics selected to measure the algorithm's effectiveness.

A parameter-free method called Dsets-DBSCAN was reported by Hou et al. [19]. A histogram equalization transformation of similarity matrices was executed in this work to create a dominant set of independent parameters. The quality of the results was estimated using the F-measure metric. The results showed a remarkable performance of the parameter-free algorithm. The methodology presented in [20,21] also used a parameter-free clustering process for DBSCAN using the nearest neighbor function commonly denoted as k -dist. The evaluation of the algorithm was performed by visual inspection of the results. Ozkok and Celik [22] presented a novel algorithm called AE-DBSCAN, which included a method for the automatic definition of parameters Eps and MinPts. They also considered the k -dist by using the nearest neighbor function. Soni and Ganatra [23] proposed a new algorithm called AGED. The methodology defines a group of densities extracted from the dataset clustered using the well-known nearest neighbor function, specifically the k -dist plot. This work evaluated a variety of performance metrics, including the Dunn Index, the Pearson Gamma coefficient, and the Entropy. Other methodologies based on the DBSCAN algorithm were presented in [24–28].

As shown above, a large variety of metrics have been employed. However, Entropy as a metric has not been quite used for the performance evaluation of the DBSCAN algorithm and its variants. Here lies the intention of using Entropy as an evaluation metric considering the information given by DBSCAN after performing the clustering analysis. Entropy will manifest the orderly clustering in which results with values close to 0 are considered and grouped into datasets. In this work, the performance of a clustering hybrid algorithm called FA+GA-DBSCAN is presented, taking into account the DBSCAN algorithm as its core. This unsupervised pattern recognition algorithm was at first developed to identify the operational conditions in a structure under a variety of loads [29]. Moreover, in order to define adequate values for Eps and MinPts, a Genetic Algorithm (GA) was implemented. The GA was based on a randomized population extracted from a particular dataset D using distances selected by the nearest neighbor function and also included a set of points (x,y) belonging to the dataset D being examined. Later, a radius that represents the parameter Eps was found. In this work, the data preprocessing, including normalization and data reduction, is shown in Section 2. The definition of the DBSCAN parameters is specified in Section 3. The evaluation of FA+GA-DBSCAN is performed in Section 4. Two case studies using FA+GA-DBSCAN are presented in Section 5. Conclusions are made in Section 6.

2. Data Preprocessing

A large amount of information is collected from experiments related to knowledge discovery problems. Therefore, it is expected that under a non-trivial process, novel and potentially useful information is extracted using a data preprocessing technique.

Preprocessing techniques include strategies to quantify the reduction of the computational cost related to pattern recognition algorithms, such as cleansing data by removing noise and inconsistent or redundant information. In addition, when considering the nature of the DBSCAN algorithm, it is noted that the computational cost will decrease if the input information is represented by a two-dimensional dataset, losing a small amount of the original information. This dataset representation can be carried out using the factor analysis dimensionality reduction algorithm [30]. The steps of dimensionality reduction using FA are presented below.

2.1. Data Collection Method

The data collection method for a dataset D considers the operation of the DBSCAN algorithm, which aims to create clusters in a two-dimensional space from said dataset D . As presented by Mujica et al. [31], this dataset D is a matrix of size $m \times n$, where m is the number of row vectors x_i , i.e., experimental trials defined by a set of variables of interest in a time instant, and n is the number of column vectors v_j of one variable of interest such as the one extracted by a network of strain sensors or accelerometers.

The number of column vectors can also be assumed as the number of dimensions of the dataset D , which is represented in matrix form as

$$D_{n \times m} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1j} & d_{1m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{i1} & d_{i2} & \cdots & d_{ij} & d_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & d_{nj} & d_{mn} \end{bmatrix}, \quad (1)$$

2.2. Data Normalization

Normalization of the original database $D_{m \times n}$ allows the algorithm to process the information using compatible magnitudes. This facilitates the correlation among variables, which improves the precision of the clustering process. Normalization was carried out using auto-scaling, which transformed each variable into an element with zero mean and unity variance as

$$\bar{d}_{ij} = \frac{d_{ij} - \mu_{vj}}{\sigma_{vj}}, \quad (2)$$

where σ_{vj}^2 is defined as the variance of v_j , defined by:

$$\sigma_{vj}^2 = \frac{1}{n-1} \sum_{i=1}^n (d_{ij} - \mu_{vj})^2, \quad (3)$$

where μ_{vj} is the mean of the variable of interest v_j .

2.3. Dimensionality Reduction Technique

The process of dimensionality reduction was performed using the linear Factor Analysis (FA) method, which has similar characteristics to principal component analysis; as FA, PCA is also a linear technique based on orthogonal projections [32]. PCA is a widely known dimensionality reduction technique that reduces the size of the dataset based on the co-variance of the original information. Nevertheless, FA reveals underlying information hidden in the original dataset using a combination of linear variables m , with $m < p$, except for an error term with a length size equal to the original dataset. The general form of the FA method is presented using the notation presented by I. T. Jolliffe [33]:

$$\begin{aligned}
x_1 &= \lambda_{11}f_1 + \lambda_{12}f_2 + \cdots + \lambda_{1m}f_m + \epsilon_1 \\
x_2 &= \lambda_{21}f_1 + \lambda_{22}f_2 + \cdots + \lambda_{2m}f_m + \epsilon_2 \\
&\vdots \\
x_p &= \lambda_{p1}f_1 + \lambda_{p2}f_2 + \cdots + \lambda_{pm}f_m + \epsilon_p,
\end{aligned}$$

where x represents the attributes or original variables x_1, x_2, \dots, x_p , λ_{jk} is the factor loadings: $j = 1, 2, \dots, p, k = 1, 2, \dots, m$, while f_1, f_2, \dots, f_m represent the common factors and ϵ_p is defined as a residual error vector of specific factors. In general, a matrix representation of FA is given by:

$$X = \Lambda F + \epsilon. \quad (4)$$

In this case, the factor loading Λ and the common factors F remain unknown; thus, in contrast with a standard regression model, the FA technique can lead to different solutions, which means there will not be a single solution. FA may be represented in terms of co-variances as follows:

$$\Sigma = \Lambda \Lambda' + \Psi, \quad (5)$$

where Σ represents the covariance or correlation matrix and Ψ the covariance of the specific factors ϵ_p using the maximum likelihood estimation, allowing finding more precise values of Λ and Ψ .

In addition, a rotation matrix T can be included in order to define different solutions for FA: $\Lambda^* = \Lambda T$, which, after a mathematical process T , is included in the rotation as $\Lambda \Lambda'$. Varimax [34], Quartimax [35], and Promax [36] are commonly used as factorial rotation methods. The pseudo-code of FA is presented in Algorithm 1.

Algorithm 1: Dimensionality reduction using Factor Analysis.

```

Data:  $D_{m \times n}$ , with  $m$  rows (time instants) and  $n$  columns (number of sensors)
Result:  $D_{m \times 2}$ , dimensionality reduced matrix
/* Determine the number of factors                                     */
if Eigenvalues  $\geq 1$  then
    | Select the number of factors with eigenvalues greater than one
    |   (number of factors  $< n$ );
else
    | Select the number of factors desired by user criteria;
end
/* Choose the ideal rotation matrix: varimax, promax, quartimax, or
   none                                                                */
/* Select the number of common factors desired                        */
while  $D_{m \times n} \neq \Lambda \times F + \epsilon$  do
    | Estimate factor loadings  $\lambda$ ;
    | Generate the common factor predictions  $f$ ;
    | Define the specific variances or specific factors  $e$ ;
    | Rotate factors until the equality is achieved;
end

```

The desired rotating factors will reduce the original dimension of a dataset correlated to several specific eigenvalues that can describe the retained information. Several techniques are used to quantify the information retained after the dimensionality reduction. These techniques include the “eigenvalues greater than one” rule, the definition of the cumulative variance over 80%, and the scree-plot rule, which is a graphic method where the breaking point of a curve of factors against eigenvalues is identified as the point related to the number of appropriate eigenvalues.

In short, as common factors, f_m will preserve hidden relationships among variables, and the DBSCAN algorithm performs its clustering process in a two-dimensional space; a strategic approach is to preserve the first two common factors from the original dataset, which can be represented and plotted in a two-dimensional space. It is also expected that different magnitudes related to specific values inside the dataset remain preserved after the dimensionality reduction process, allowing DBSCAN to identify well-defined clusters. In other words, the original dataset of dimension $D_{m \times n}$ of m variables and n sensors or dimensions can be represented in a lower dimension using the FA's first two common factors as a $D_{m \times 2}$ dataset. Moreover, the new reduced dataset can now be graphically represented in a Cartesian coordinate system, in which every row from dataset $D_{m \times 2}$ represents an x, y point in space. An example of the dimensionality reduction of a dataset of six-column vectors containing two classes is presented in Figure 1, in which the first three dimensions are plotted. After performing a dimensionality reduction using the first two common factors, the projection of these two classes is observable. Figure 1a presents a graphic representation of an artificial dataset with $D_{1000 \times 6}$. Figure 1b illustrates the projection of the first two common factors of the artificial dataset with $D_{1000 \times 2}$.

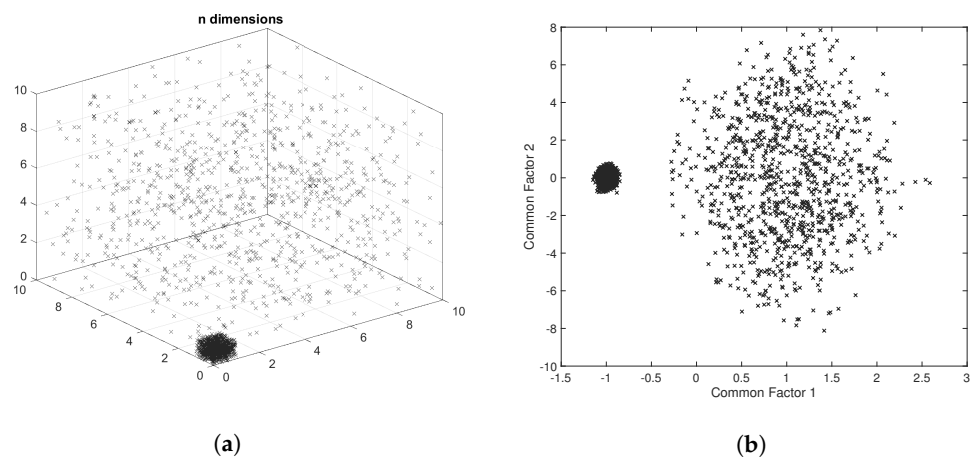


Figure 1. Example of a dimensionality reduction of a dataset D . (a) Three-dimensional scatter-plot of dataset $D_{1000 \times 6}$. (b) Scatter-plot of artificial dataset projection $D_{1000 \times 2}$.

3. Dbscan Algorithm

As previously mentioned, DBSCAN is an unsupervised density-based algorithm. It was designed to define specific clusters from a particular dataset, usually in a two-dimensional space $D_{m \times 2}$, without the need for predefined class labels. However, the algorithm is not fully automatized, thus the necessity to define two entry parameters Eps and MinPts. Nonetheless, the DBSCAN algorithm is still relevant due to its exploratory characteristics and its acceptable computational cost $O(n \log n)$, for large datasets [37].

These initial parameters allow the algorithm to define a specific group of correlated points depending on their Euclidean distance defined by a circle with radii Eps and a minimum specified number of correlated elements MinPts. Moreover, these entry parameters define a group of points that may have no correlation with any of the discovered clusters and will be treated as noise. The pseudo-code of the DBSCAN algorithm is presented in Algorithm 2.

The DBSCAN algorithm follows a series of rules to define clusters from a specific dataset D , considering two arbitrary points p and q from D . These rules are defined as follows:

- Eps-neighborhood of point: The Eps-neighborhood of a point p , denoted by $N_{\text{Eps}}(p)$, is defined by $N_{\text{Eps}}(p) = \{q \in D | \text{dist}(p, q) \leq \text{Eps}\}$.
- Directly density-reachable: A point p is directly density-reachable from a point q if:
 - $p \in N_{\text{Eps}}(q)$.

- The core point condition is reached, i.e., $N_{\text{Eps}}(q) \geq \text{MinPts}$.
- Density-reachable: A point p is density-reachable from a point q if there is a set of points p_1, \dots, p_n , with $p_1 = q$ and $p_n = p$, such that p_{i+1} is directly density-reachable from p_i .
- Density-connected: A point p is density-connected to a point q if there is a point o such that p and q are density-reachable from o .
- Cluster: Let D be a specific dataset. A cluster is a non-empty subgroup from dataset D that meets the following criteria:
 - Maximality $\forall p, q$: if $p \in C$ and q is density-reachable from p , then $q \in C$.
 - Connectivity $\forall p, q \in C$, then p is density-connected to q .
- Noise: Let C_1, \dots, C_k be the clusters of dataset D . Noise is defined as the set of points in the dataset D not belonging to any cluster C_i , that is $p \in D | \forall i : p \notin C_i$.

Algorithm 2: Clustering using DBSCAN algorithm.

```

Data:  $D_{m,2}$  and Eps
Result: A number  $N$  of clusters  $C_N$  and noise
/* Determine the value of MinPts                                     */
MinPts =  $\frac{1}{n} \sum d_i$ ;
/* Let  $X_{un}$  be a set of unvisited points from  $D_{m,2}$                  */
Set  $C = 0$ ;
Set  $\emptyset = \text{Empty Set}$ ;
while  $X_{un} \neq \emptyset$  do
  Randomly select a point  $p_{i,j} \in X_{un}$ ;
  if  $p_{i,j}$  is a noncore point then
    Mark  $p_{i,j}$  as noise;
     $X_{un} = X_{un} - p_{i,j}$ ;
  else
     $N = N + 1$ ;
    Determine all density-reachable points from  $p_{i,j}$ ;
    Assign  $p_{i,j}$  and previous points to a cluster  $C_N$ ;
     $X_{un} = X_{un} - C_N$ ;
  end
  Points marked as noise are also assigned to a special cluster  $C_N$ ;
end
/* Tag each generated cluster  $C_N$  with a natural number; tag noise
   with 0                                                         */
/* plot the clusters with a specific color                         */

```

As mentioned before, the selection of the initial parameters will impact the algorithm's overall clustering precision and computational complexity. Therefore, a strategy is needed to define the parameters MinPts and Eps seeking to remove human handling and improve the precision in the overall process. Another important characteristic of the DBSCAN algorithm is its ability to automatically define outliers as noise, excluding undesired or redundant information.

3.1. Definition of Parameter MinPts

The function nearest neighbor is considered to define the parameter MinPts. The function nearest neighbor defines specific distances in the proximity of an element belonging to a dataset $D_{m \times n}$ using the Euclidean distance among variables. These distances can be assumed as particular densities in the cloud of points. As presented by Gaonkar and Sawant Gaonkar and Sawant [38], the parameter MinPts can be defined using the function sample mean of the particular densities from a specific dataset $D_{m \times n}$:

$$\text{MinPts} = \frac{1}{n} \sum_{i=1}^n d_i, \quad (6)$$

where d_i is every value of density assessed by the function nearest neighbor in a specific dataset $D_{m \times 2}$ with m number of samples.

3.2. Definition of Parameter Eps Using a Fitness Proportionate Selection

As presented in the previous section, the parameter Eps can be interpreted as an imaginary radius of a circumference inscribed around an arbitrary point included in a specific dataset of dimension two $D_{m \times 2}$. It is necessary to find an adequate Eps value since this parameter affects the overall clustering process of the hybrid algorithm in terms of computational cost and precision. According to the previous statement, the algorithm may lose its clustering capacity if the parameter Eps is selected arbitrarily as the user disregards the magnitude and density of the specific dataset being analyzed. As a solution, a Genetic Algorithm (GA) based on a fitness function is considered to define a particular Eps for a dataset $D_{m \times 2}$. The GA is defined using the model presented by [39].

Typical distances or densities are determined using the function nearest neighbor for a specific dataset $D_{m \times 2}$ to define the initial population of the selected GA model. These distances are defined as the standard radii from the specific dataset. The initial density population is defined in 50 elements with magnitudes between the average radius r_{avg} and the maximum radius r_{max} . Points with coordinates belonging to the dimensionality-reduced dataset $p_{x,y} \in D_{m \times 2}$ are selected as additional alleles in the chromosome associated with the initial population to be optimized via the GA. These points are considered the center of the possible radii. Therefore, the chromosome could have the following structure presented in Table 1.

Table 1. A scheme of a chromosome belonging to the initial population with two alleles; one is the point p , and the other is the radius r .

Allele 1		Allele 2
x	y	Radius r
51.606	12.783	1.036

Furthermore, the fitness function ff to be optimized will have the following outline:

$$ff = \frac{CR \times SD}{DR}, \quad (7)$$

where CR is considered as the coverage ratio and is calculated as follows:

$$CR = \frac{|S_{p_1, r_1} \cup S_{p_2, r_2} \cdots \cup S_{p_n, r_n}|}{|D|}, \quad (8)$$

SD being defined the Sum of Density and evaluated as follows:

$$SD = \sum_{i=1}^n \frac{|S_{p_i, r_i}|}{|r_i^2|}, \quad (9)$$

while DR is established as the Duplicate Ratio and is calculated as follows:

$$DR = \frac{\sum_{i=1}^n |S_{p_i, r_i}|}{|S_{p_1, r_1} \cup S_{p_2, r_2} \cdots \cup S_{p_n, r_n}|}, \quad (10)$$

in general, S_{p_i, r_i} can be defined as the chromosome of center p_i and radius r_i .

The crossover process was performed by selecting radii from the initial population and rearranging their position while points (x,y) remained in the same initial position. These new configurations are the offspring and are defined for a tournament selection by evaluating the fitness function. The definitions of parameters Eps and MinPts are described in Algorithm 3.

Algorithm 3: Selection of DBSCAN parameters using a genetic algorithm.

```

Data:  $D_{m \times 2}$ 
Result: Eps
/* Nearest neighbor densities */
for  $i \leftarrow 1$  to  $m$  do
    | Choose a random point  $p_{i,j}$  belonging to the dataset  $D_{m \times 2}$ ;
end
Make a measure of nearest neighbor densities from  $p_{i,j}$ ;
foreach density measure  $d_i$  do determine the mean density and the maximum density ;
/* Create initial population */
for mean density to max density do
    | Randomly generate an initial population;
end
while number of iterations desired do
    /* Call the fitness function  $ff$  */
     $ff = \text{coverage ratio} / (\text{sum of density} \times \text{duplicate ratio})$ ;
    /* Evaluate the initial population with  $ff$  */
    Preserve the better population  $\rightarrow$  parents;
    /* Crossover */
    for  $i \leftarrow 1$  to number of parents do
        | Vary a desired number of density measures in random positions;
    end
    /* Mutation */
    for  $i \leftarrow 1$  to number of parents do
        | replace a desired number of new random density measures in random
        | positions;
    end
    /* Evaluate the new population with the  $ff$  */
end
Select the minimum density value  $d_i$  from the new population  $d_{min}$ ;
Set  $d_{min} = \text{Eps}$ ;

```

4. Results

4.1. Performance Evaluation Metrics

4.1.1. Precision

The precision metric of a classifier is an evaluation parameter that is primarily used for classification performance [40]. Precision is a direct measurement of the quality of the information obtained by a clustering algorithm. For example, the precision of a classifier can be measured using the propositions presented by [41]:

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}, \quad (11)$$

where tp is defined as the true positive rate, hit rate, or recall of the clustering algorithm and is defined by:

$$\text{tp} = \frac{\text{positives correctly classified}}{\text{total positives}}, \quad (12)$$

and fp is defined as the false positive rate of the clustering algorithm:

$$fp = \frac{\text{negative incorrectly classified}}{\text{total negatives}}. \quad (13)$$

4.1.2. Entropy

Currently, the evaluation of Entropy has been extended as a popular metric, considering the homogeneity of a pattern recognition algorithm [42,43]. In the machine learning context, Entropy can be measured in the output parameters of the classifier as a way to define the disorder of the information processed by the algorithm. For example, the Entropy, $H(S)$, is measured as follows:

$$H(S) = p_i \log_2 p_i, \quad (14)$$

where p_i is defined as the probability of an i th element belonging to a specific class.

The Entropy indicates the degree of randomness in the data set when used as a metric to estimate the data set's uncertainty. In this regard, when applying a recognition algorithm to a specific data set, it is expected that the classified data presents a reduction in its Entropy. The Entropy difference between unclassified and classified data represents the amount of information gained after applying a classification method. This difference or information gain, $IG(A, S)$, also indicates the uncertainty reduction after splitting the data on a feature (i.e., the more significant the information gained, the greater the decrease in Entropy or uncertainty). The information gained is given as follows:

$$IG(A, S) = H(S) - \sum_{j=1}^m \frac{n_j}{n} \cdot H(S, A), \quad (15)$$

n_j being the number of instances with a j value of an attribute A , n the total number of instances in the dataset, m the set of distinct values of an attribute A , $H(S_j)$ the Entropy of the subset of instances for attribute A , and $H(S, A)$ the Entropy of an attribute A . In the context of the DBSCAN algorithm, these futures or partitions include data in either of the following attributes: correctly classified (tp), negative incorrectly classified (fp), and noise data.

4.1.3. Calinski–Harabasz Clustering Evaluation Method

Calinski and Harabasz [44] presented a clustering evaluation technique that suggests a suitable number of clusters of a specific dataset being analyzed. This exploratory technique, also named the CH Index, evaluates the cohesion or dispersion among elements considering a variance index. Following the notation of the CH Index, the technique is defined by:

$$CH(K) = \frac{B(K)(N - K)}{W(K)(K - 1)}, \quad (16)$$

considering $B(K)$ as the inter-cluster covariance or divergence:

$$B(K) = \sum_{k=1}^K a_k ||\bar{x}_k - \bar{x}||^2; \quad (17)$$

furthermore, $W(K)$ is considered as the intra-cluster covariance and is defined as:

$$W(K) = \sum_{k=1}^K \sum c(j) = k ||d_i - \bar{x}_k||^2 \quad (18)$$

where K represents the number of clusters, d_i is the i th defined cluster, and N represents the number of elements or samples.

4.2. Clustering Performance Analysis

The unsupervised classification methodology included the preliminary processing, processing, and postprocessing of dataset $D_{n \times m}$. The classification was performed using the MATLAB R2021b numerical programming software for Windows 11 with an Intel Core i7, 2.11 GHz processor, 8 GB of RAM, and 1 TB hard drive PC. The hybrid algorithm was created by using a combination of pre-established MATLAB functions, scripts created from the beginning, and a modified script based on the DBSCAN algorithm developed by [45]. The selection of parameters for DBSCAN was defined as proposed in Sections 3.1 and 3.2. Finally, the FA algorithm was executed using the MATLAB built-in function Factoran, which includes the rotation method and the auto-scaling process.

To evaluate the classification performance of the hybrid algorithm FA+GA-DBSCAN, six artificial datasets were selected. The results of the different datasets classified with the FA+GA-DBSCAN algorithm are observed in Figure 2. DBSCAN parameters were defined automatically using a genetic algorithm for each dataset as presented in Section 3; however, the GA method employed a randomized basis; therefore, in order to have control over the selection of parameters Eps and MinPts, each clustering experiment was executed 30 times, and then, the standard deviation was measured for each case. The results of the obtained values for the mentioned parameters are presented in Table 2. Moreover, the well-known pattern recognition algorithm K-means was selected as a comparison benchmark. Although K-means is considered an unsupervised clustering method, it correlates elements taking into account a number of centroids selected a priori by the user. The clustering performance of K-means is illustrated in Figure 3. The results of K-means were also used for a comparative study of FA+GA-DBSCAN's performance employing the Calinski–Harabasz clustering evaluation method. Information related to the comparative study is presented in Table 3.

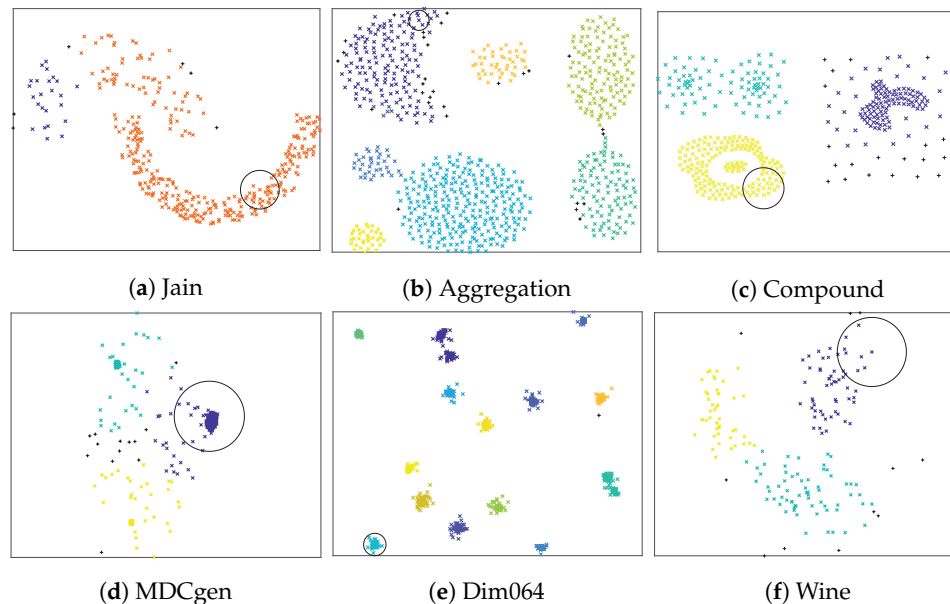


Figure 2. Clustering results using the FA+GA-DBSCAN algorithm with artificial datasets. Noise points are marked by “+”.

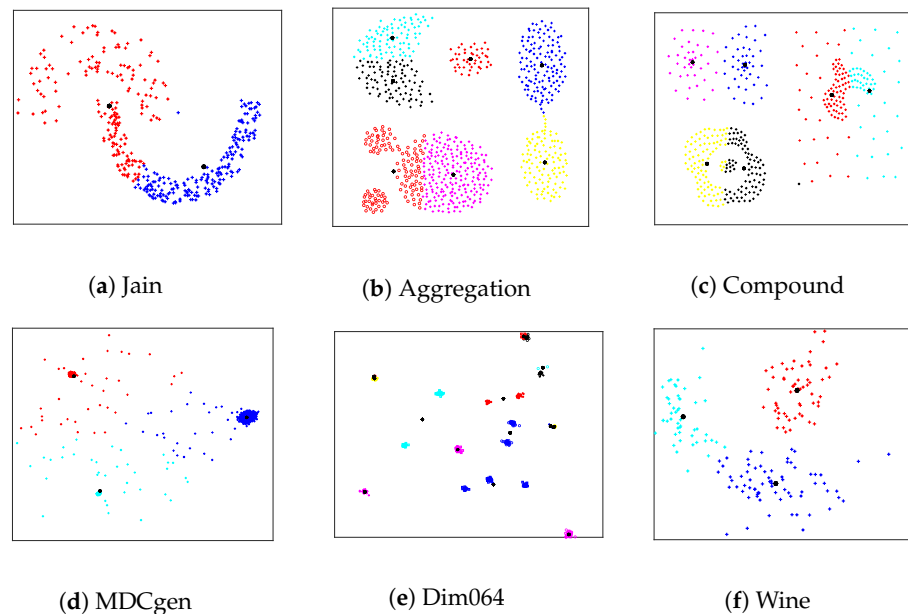
A two-dimensional dataset called Half-ring (Jain) was proposed by Jain and Fred [46]; this dataset is composed of two classes with uneven densities between clusters. Each cluster is well separated, and the top one is made up of 97 elements and the bottom one of 276 elements. As presented in Figure 2a, the algorithm was able to identify two groups; however, certain elements were defined as noise, and the precision was reduced as some elements from the top cluster were placed in the group belonging to the bottom cluster.

Table 2. Automatic definition of FA+GA-DBSCAN's parameters Eps and MinPts; values are represented using their mean and standard deviation after 30 runs of the algorithm.

Dataset Name	Eps	MinPts
Aggregation	$1.130 \pm 2.52 \times 10^{-5}$	5.498 ± 0.00
Compound	$2.413 \pm 2.60 \times 10^{-4}$	7.710 ± 0.00
Jain	$2.550 \pm 3.89 \times 10^{-4}$	5.228 ± 0.00
Dim064	$0.142 \pm 1.95 \times 10^{-6}$	5.898 ± 0.00
Wine	$0.694 \pm 9.90 \times 10^{-6}$	23.042 ± 0.00
MDCgen	$0.872 \pm 3.71 \times 10^{-5}$	26.055 ± 0.00

Table 3. A comparative study of clustering performance using the Calinski–Harabasz clustering evaluation method and FA+GA-DBSCAN; C refers to cluster.

Dataset Name	Classes	Calinski–Harabasz Optimal C	C Defined by FA+GA-DBSCAN
Jain	2	9	2
Aggregation	7	6	7
Compound	2	2	3
MDCgen	3	5	3
dim064	16	16	16
Wine	3	3	3

**Figure 3.** Clustering results of artificial datasets using K-means algorithm with artificial datasets.

Another dataset named Aggregation is a two-dimensional, heterogeneous synthetic distributed dataset of seven classes and 788 elements, proposed by [47]. As a result, the FA+GA-DBSCAN detected eight clusters, and some elements were considered as noise, as presented in Figure 2b; nevertheless, the precision of the algorithm was not quite affected.

The artificial dataset Compound was presented in 1971 by Zahn C. [48]. This two-dimensional dataset presents six groups with different densities and shapes and is one of the most-common datasets for clustering validation. It was evident that the hybrid algorithm was not plausible in terms of a correct grouping in most of the presented clusters, as shown in Figure 2c. Its precision was low considering that the DBSCAN parameters were defined to cover an overall density. In general, three clusters were determined. The evaluation of precision is presented in Figure 4.

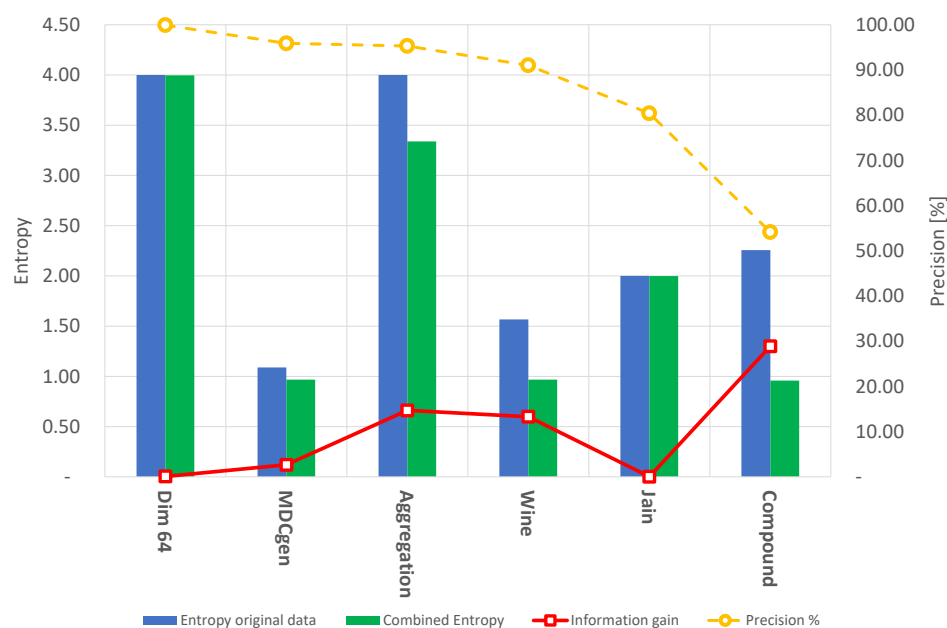


Figure 4. Clustering results, precision, Entropy, and information gain using the hybrid algorithm FA+GA-DBSCAN.

MDCgen is a synthetic multidimensional dataset produced by the algorithm developed by Iglesias et al. [49]. The dataset generator is capable of producing artificial n -dimensional datasets including outliers or noise. For this study, a three-class dataset with six dimensions was generated; furthermore, it possesses 2000 observations and 100 outlier points. The dimensionality reduction process proposes that two common factors are enough to represent 77.810 % of the original variability. Clustering performance is presented in Figure 2d.

The high-dimensional dataset, Dim064, reported by [50], was also considered for a clustering analysis. The relevance of this dataset relies on the need to evaluate the clustering algorithm in a high-dimensional instance. The dimensionality reduction process suggests that 15 common factors are needed to represent 99.890 % of the original variability. Nevertheless, the first two common factors were selected for the clustering process. The synthetic Gaussian clusters are well separated even for this higher-dimensional case. In terms of precision, the clustering results of FA+GA-DBSCAN were satisfactory. As presented in Figure 2e, the algorithm was able to group almost all clusters represented by the common factors from the FA dimensionality reduction process.

Finally, the multivariate dataset called Wine [51] was considered for validation purposes. This dataset presents 13 attributes, which belong to wine characteristics such as color intensity, alcohol, and minerals, among others, and three classes related to three different cultivars. A dimensionality reduction was performed considering the study of cumulative variance. As a result, three common factors are recommended to be retained, as they represent 66.530% of the original variability. Nonetheless, the first two common factors were selected for the clustering process. The algorithm's precision was acceptable, as presented in Figure 4; the evaluated dataset was grouped into three different clusters, as indicated in Figure 2f.

As previously mentioned, the clustering performance was measured using the external clustering evaluation metrics Entropy and precision and a comparative study using the Calinski–Harabasz clustering evaluation method. As shown in Figure 4, the clustering algorithm is capable of grouping well-condensed datasets with significant precision; nevertheless, the Entropy of the resulting clusters on these types of datasets is almost invariant. On the other hand, the information gain is relatively low in condensed datasets. Furthermore, it is evident that the automatic grouping process held by FA+GA-DBSCAN can perform clustering with similar characteristics as those presented by K-means, consid-

ering the Calinski–Harabasz clustering estimation technique. However, FA+GA-DBSCAN presents an advantage when evaluating datasets with an unknown number of classes, taking into consideration that the number of centroids or classes is not previously needed; this ensures a decent level of reliability of the results in an exploratory analysis performed by the presented methodology. The Entropy and precision of K-means are also reported in Figure 5.

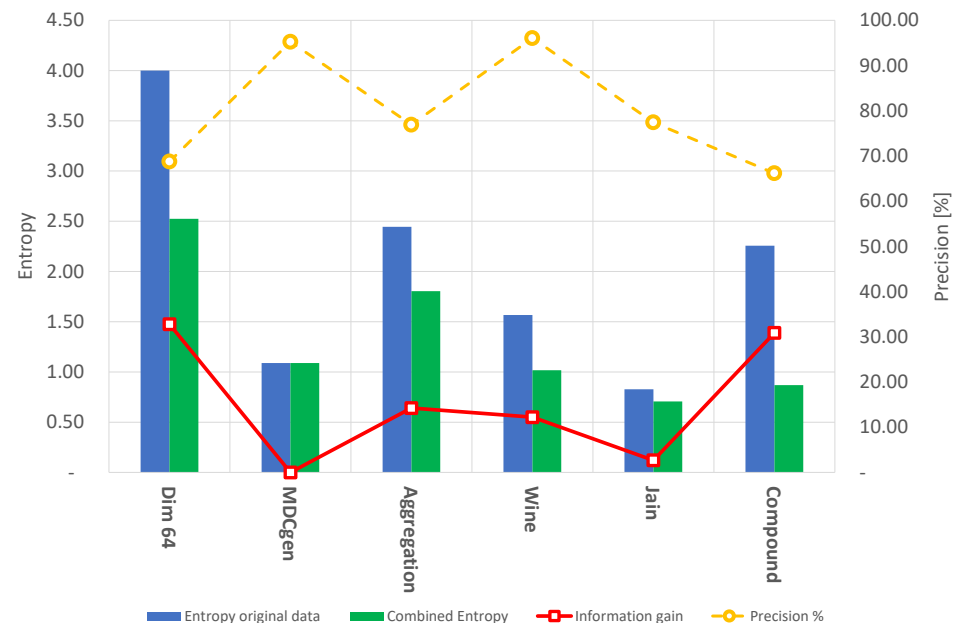


Figure 5. Clustering results, precision, Entropy, and information gain using K-means.

5. Case Studies

5.1. Aircraft Engine Degradation

The work developed by Saxena et al. [52] presented a group of datasets of an aircraft's engine thermodynamic model simulation using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) software. Simulations were carried out under various operational conditions with induced damage propagation. The free access engine dataset is available in the Prognostics Data Repository from the National Aeronautics and Space Administration (NASA) [53]. The study included four different output cases with different altitude, Mach number, and temperature conditions. Each output case behaves differently and includes a specific number of operational conditions and degradation, where outputs are gathered in a dataset matrix.

Considering the analysis proposed in Section 2.1, the dataset collection configuration consists of 21 sensors taking into account the low and high compressor and turbine temperature, pressure, flow speed, and fuel flow, and rows are defined as time instants. Hence, the dataset has a size of $\text{engine}_{33991 \times 21}$. The operational conditions and degradation of the engine are sorted in a way that is not directly quantifiable. The selected dataset includes six operational conditions and one fault mode belonging to the degradation of the high-pressure compressor.

The performed analysis began with a dimensional reduction using FA. Then, the factor selection was analyzed by the “eigenvalues greater than one” method; this suggests that the first two common factors represent 97.550 % of the variability of the original information. Finally, the scree-plot, as presented in Figure 6, represents the eigenvalue of each factor belonging to the considered dataset; it clarifies the FA and illustrates FA's capacity to retain a large amount of information in a lower dimension.

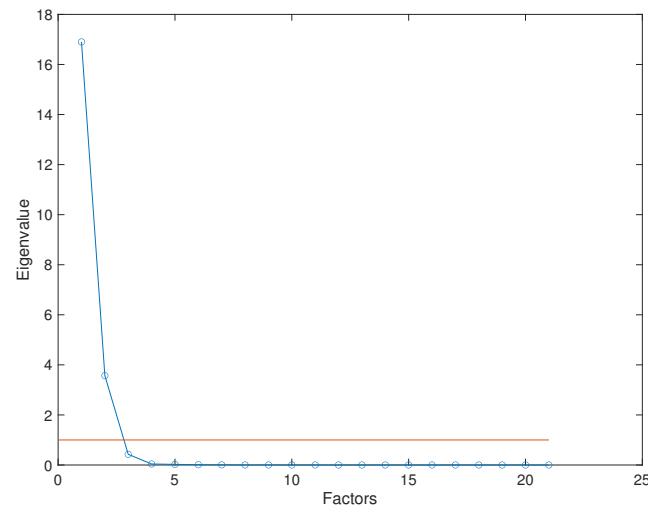


Figure 6. Scree-plot from the aircraft engine operational conditions and degradation dataset. Two common factors are sufficient to represent a large quantity of the original information.

Furthermore, to perform this exploratory analysis, a matrix of engine_{33991×2} was obtained. The two first common factors were considered as points in a two-dimensional space and then automatically grouped by the hybrid algorithm. Parameters $\text{MinPts} = 6.480 \pm 0$ and $\text{Eps} = 0.010 \pm 0$ were defined, and the standard deviation of both parameters was measured after 30 equal runs. As a result, six well-condensed clusters were found. One of them is mainly traced out from the other five as presented in Figure 7. This may indicate a set of parameters related to engine degradation.

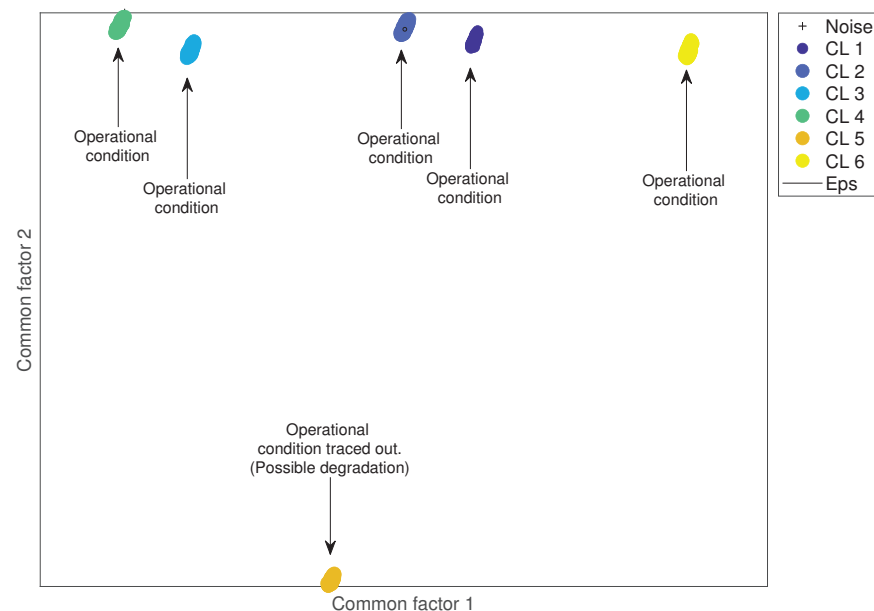


Figure 7. Clustering analysis of an aircraft engine considering different operational conditions and one mode of degradation.

5.2. Lidar Dataset

The Lidar dataset is a free access dataset available in Matlab R2021b [54] for clustering analysis. Lidar is a mapping system that employs laser energy for high-resolution spatial sensing. This laser technology has been widely used for digital cartography, military applications, cellphones, and autonomous mobility. In this study, the linear dataset presents

a spatial overview of a street with a vehicle and various objects such as trees and buildings. This dataset can be assumed as a two-dimensional top base view from a surveillance unmanned aerial vehicle; a contextualization image is presented in Figure 8.



Figure 8. Contextualization picture considering a similar point of view to the Lidar dataset.

The information from the Lidar dataset can be used for an exploratory analysis using clustering in order to identify possible objects in an unsupervised manner. In this two-dimensional spatial dataset, the space is limited to a range of $20\text{ m} \times 20\text{ m}$. The matrix considered for this study is a two-dimensional set of points of size $\text{lidar}_{19070 \times 2}$, and the scatter plot of this dataset is presented in Figure 9a. This set was therefore analyzed automatically by GA-DBSCAN. Parameters $\text{MinPts} = 0.207 \pm 0$ and $\text{Eps} = 3.228 \pm 0.011$ were defined after ten equal runs. The algorithm was capable of determining 13 different clusters, using a mean neighbor circumference value of 3.22 m approximately. This allows for the identification of elements such as the car in the center of the figure and the other one in front of it. Similarly, the algorithm was able to identify other obstacles, such as trees and borders belonging to the sidewalk. The exploratory result is presented in Figure 9b.

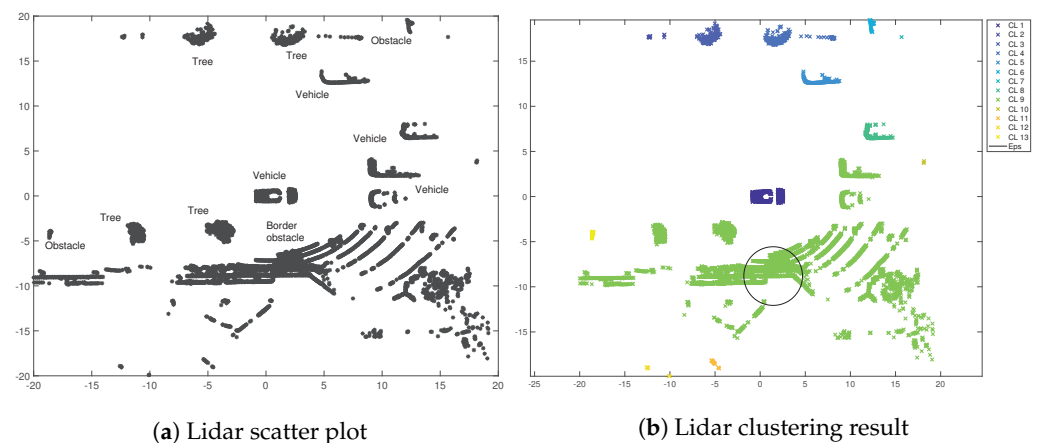


Figure 9. Results of the exploratory analysis using the hybrid algorithm from the Lidar dataset.

6. Conclusions

In summary, the evaluation of a clustering hybrid algorithm called FA+GA-DBSCAN was presented using Entropy and precision performance metrics and a comparative study employing the Calinski–Harabasz clustering evaluation method on different artificial datasets. This unsupervised pattern recognition algorithm was first developed to identify

the operational conditions in a structure under various loads. The dimensionality reduction technique Factor Analysis described the information from the dataset as a combination of specific factors that could be clustered later using DBSCAN. However, DBSCAN on its own cannot automatically define clusters in a particular dataset as the parameters Eps and MinPts need to be selected before the recognition of patterns. A large number of variations of DBSCAN are still being proposed since the clustering algorithm operates according to the parameters Eps and MinPts. These parameters can be defined using many deterministic techniques, including density studies, genetic algorithms, and evaluations made by “hand” iterations, among others. As the algorithm is implemented using various parameter definition techniques, many variations of the clustering results are presented. The performance of FA+GA-DBSCAN clustering was defined using Entropy, precision, and a comparative study, which included the well-known clustering algorithm K-means. The hybrid algorithm automatically clustered datasets with condensed and scattered groups with notable precision; however, the information gained in this type of dataset was almost null as the variation of the Entropy did not change significantly.

Author Contributions: Conceptualization, J.C.P.-L. and J.S.-P.; formal analysis, J.C.P.-L. and C.N.-L.; investigation, J.C.P.-L. and J.S.-P.; writing-original draft preparation, J.C.P.-L. and V.L.F.-G.; writing-review and editing, J.C.P.-L. and V.L.F.-G. All authors have read and agreed to the published version of the manuscript.

Funding: The present work was funded by the Centro de Investigación para el Desarrollo y la Innovación CIDI from Universidad Pontificia Bolivariana Sede Central (No. 636B-06/16–57).

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. Kriegel, H.P.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 231–240. [\[CrossRef\]](#)
2. Bhattacharjee, P.; Mitra, P. A survey of density based clustering algorithms. *Front. Comput. Sci.* **2021**, *15*, 1–27. [\[CrossRef\]](#)
3. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Knowledge Discovery and Data Mining KDD, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
4. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst. (TODS)* **2017**, *42*, 1–21. [\[CrossRef\]](#)
5. Ng, R.T.; Han, J. CLARANS: A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 1003–1016. [\[CrossRef\]](#)
6. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [\[CrossRef\]](#)
7. Hinneburg, A.; Gabriel, H.H. Denclue 2.0: Fast clustering based on kernel density estimation. In *International Symposium on Intelligent Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 70–80.
8. Pei, T.; Jasra, A.; Hand, D.J.; Zhu, A.X.; Zhou, C. DECODE: A new method for discovering clusters of different densities in spatial data. *Data Min. Knowl. Discov.* **2009**, *18*, 337–369. [\[CrossRef\]](#)
9. de Moura Ventorim, I.; Luchi, D.; Rodrigues, A.L.; Varejão, F.M. BIRCHSCAN: A sampling method for applying DBSCAN to large datasets. *Expert Syst. Appl.* **2021**, *184*, 115518. [\[CrossRef\]](#)
10. Lai, W.; Zhou, M.; Hu, F.; Bian, K.; Song, Q. A new DBSCAN parameters determination method based on improved MVO. *IEEE Access* **2019**, *7*, 104085–104095. [\[CrossRef\]](#)
11. Wang, C.; Ji, M.; Wang, J.; Wen, W.; Li, T.; Sun, Y. An improved DBSCAN method for LiDAR data segmentation with automatic Eps estimation. *Sensors* **2019**, *19*, 172. [\[CrossRef\]](#)
12. Darong, H.; Peng, W. Grid-based DBSCAN algorithm with referential parameters. *Phys. Procedia* **2012**, *24*, 1166–1170. [\[CrossRef\]](#)
13. Ohadi, N.; Kamandi, A.; Shabankhah, M.; Fatemi, S.M.; Hosseini, S.M.; Mahmoudi, A. Sw-dbscan: A grid-based dbscan algorithm for large datasets. In Proceedings of the 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 22–23 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 139–145.

14. Shamisa, A.; Majidi, B.; Patra, J.C. Sliding-window-based real-time model order reduction for stability prediction in smart grid. *IEEE Trans. Power Syst.* **2018**, *34*, 326–337. [\[CrossRef\]](#)
15. Karami, A.; Johansson, R. Choosing DBSCAN parameters automatically using differential evolution. *Int. J. Comput. Appl.* **2014**, *91*, 1–11. [\[CrossRef\]](#)
16. Kumar, K.M.; Reddy, A.R.M. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Pattern Recognit.* **2016**, *58*, 39–48. [\[CrossRef\]](#)
17. Zhu, L.; Zhu, J.; Bao, C.; Zhou, L.; Wang, C.; Kong, B. Improvement of DBSCAN Algorithm Based on Adaptive Eps Parameter Estimation. In Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 21–23 December 2018; pp. 1–7.
18. Zhu, Q.; Tang, X.; Elahi, A. Application of the novel harmony search optimization algorithm for DBSCAN clustering. *Expert Syst. Appl.* **2021**, *178*, 115054. [\[CrossRef\]](#)
19. Hou, J.; Gao, H.; Li, X. DSets-DBSCAN: A parameter-free clustering algorithm. *IEEE Trans. Image Process.* **2016**, *25*, 3182–3193. [\[CrossRef\]](#)
20. Starczewski, A.; Goetzen, P.; Er, M.J. A new method for automatic determining of the DBSCAN parameters. *J. Artif. Intell. Soft Comput. Res.* **2020**, *10*, 209–221. [\[CrossRef\]](#)
21. Starczewski, A.; Cader, A. Determining the EPS parameter of the DBSCAN algorithm. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 16–20 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 420–430.
22. Ozkok, F.O.; Celik, M. A new approach to determine Eps parameter of DBSCAN algorithm. *Int. J. Intell. Syst. Appl. Eng.* **2017**, *5*, 247–251. [\[CrossRef\]](#)
23. Soni, N.; Ganatra, A. Aged (automatic generation of eps for dbscan). *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 536.
24. Birant, D.; Kut, A. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data Knowl. Eng.* **2007**, *60*, 208–221. [\[CrossRef\]](#)
25. Li, M.; Bi, X.; Wang, L.; Han, X. A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm. *Comput. Commun.* **2021**, *167*, 75–84. [\[CrossRef\]](#)
26. He, Y.; Tan, H.; Luo, W.; Mao, H.; Ma, D.; Feng, S.; Fan, J. Mr-dbscan: An efficient parallel density-based clustering algorithm using mapreduce. In Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems, Tainan, Taiwan, 7–9 December 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 473–480.
27. Chen, Y.; Zhou, L.; Bouguila, N.; Wang, C.; Chen, Y.; Du, J. BLOCK-DBSCAN: Fast clustering for large scale data. *Pattern Recognit.* **2021**, *109*, 107624. [\[CrossRef\]](#)
28. Gholizadeh, N.; Saadatfar, H.; Hanafi, N. K-DBSCAN: An improved DBSCAN algorithm for big data. *J. Supercomput.* **2021**, *77*, 6214–6235. [\[CrossRef\]](#)
29. Perafán-López, J.C.; Sierra-Pérez, J. An unsupervised pattern recognition methodology based on factor analysis and a genetic-DBSCAN algorithm to infer operational conditions from strain measurements in structural applications. *Chin. J. Aeronaut.* **2021**, *34*, 165–181. [\[CrossRef\]](#)
30. Lawley, D.N.; Maxwell, A.E. Factor analysis as a statistical method. *J. R. Stat. Soc. Ser. Stat.* **1962**, *12*, 209–229. [\[CrossRef\]](#)
31. Mujica, L.; Rodellar, J.; Fernandez, A.; Güemes, A. Q-statistic and T2-statistic PCA-based measures for damage assessment in structures. *Struct. Health Monit.* **2011**, *10*, 539–553. [\[CrossRef\]](#)
32. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [\[CrossRef\]](#)
33. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer: New York, NY, USA, 2002.
34. Kaiser, H.F. The varimax criterion for analytic rotation in factor analysis. *Psychometrika* **1958**, *23*, 187–200. [\[CrossRef\]](#)
35. Neuhaus, J.O.; Wrigley, C. The quartimax method: An analytic approach to orthogonal simple structure 1. *Br. J. Stat. Psychol.* **1954**, *7*, 81–91. [\[CrossRef\]](#)
36. Hendrickson, A.E.; White, P.O. Promax: A quick method for rotation to oblique simple structure. *Br. J. Stat. Psychol.* **1964**, *17*, 65–70. [\[CrossRef\]](#)
37. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Chennai, India, 17–19 February 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 232–238.
38. Gaonkar, M.N.; Sawant, K. AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset. *Int. J. Adv. Comput. Theory Eng.* **2013**, *2*, 11–16.
39. Lin, C.Y.; Chang, C.C.; Lin, C.C. A new density-based scheme for clustering based on genetic algorithm. *Fundam. Inform.* **2005**, *68*, 315–331.
40. Davis, J.; Goadrich, M. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 233–240.
41. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [\[CrossRef\]](#)
42. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Makhadmeh, S.N. A novel hybrid multi-verse optimizer with K-means for text documents clustering. *Neural Comput. Appl.* **2020**, *32*, 17703–17729. [\[CrossRef\]](#)
43. Zhang, T.; Wang, H.; Chen, J.; He, E. Detecting unfavorable driving states in electroencephalography based on a PCA sample Entropy feature and multiple classification algorithms. *Entropy* **2020**, *22*, 1248. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. Theory Methods* **1974**, *3*, 1–27. [\[CrossRef\]](#)

45. Heris, M.K. DBSCAN Clustering in MATLAB. 2015. Available online: <https://yarpiz.com/255/ypml110-dbscan-clustering> (accessed on 8 February 2021).
46. Jain, A.K.; Law, M.H. Data clustering: A user's dilemma. In Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, 20–22 December 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1–10.
47. Gionis, A.; Mannila, H.; Tsaparas, P. Clustering aggregation. *ACM Trans. Knowl. Discov. Data (TKDD)* **2007**, *1*, 4-es. [[CrossRef](#)]
48. Zahn, C.T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.* **1971**, *100*, 68–86. [[CrossRef](#)]
49. Iglesias, F.; Zseby, T.; Ferreira, D.; Zimek, A. MDCGen: Multidimensional dataset generator for clustering. *J. Classif.* **2019**, *36*, 599–618. [[CrossRef](#)]
50. Franti, P.; Virtajoki, O.; Hautamaki, V. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1875–1881. [[CrossRef](#)]
51. Fränti, P.; Sieranoja, S. K-means properties on six clustering benchmark datasets. *Appl. Intell.* **2018**, *48*, 4743–4759. [[CrossRef](#)]
52. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–9.
53. Saxena, A.; Goebel, K. “Turbofan Engine Degradation Simulation Data Set”, NASA Ames Prognostics Data Repository. 2008. Available online: <http://ti.arc.nasa.gov/project/prognostic-data-repository> (accessed on 20 May 2022).
54. The Math Works, Inc. *MATLAB and Statistics Toolbox Release R2021b*; The Math Works, Inc: Natick, MA, USA, 2022.