Measurement-based Coherency Detection through Monte Carlo Consensus Clustering

Fabrizio De Caro[1a], Antonio Pepiciello[2a], Federico Milano[3b], Alfredo Vaccaro[4a]

[a]*Department of Engineering, University of Sannio, Piazza Roma 21, Benevento, 82100, Italy*
[b]*School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04 V1W8, Ireland*

Abstract

This paper proposes a novel data-driven algorithm for measurement-based coherency detection in power systems, which is based on Monte Carlo Consensus Clustering (M3C). Unlike the clustering techniques conventionally adopted for coherency detection, M3C automatically identifies the optimal and stable number of coherent groups, despite the time-varying phenomena affecting power system operation. The proposed methodology is tested and validated with the IEEE 39-bus system. Results are compared with other existing clustering techniques, where the Friedman test with post-hoc analysis is performed on returned clustering scores to assess the statistically significant difference between the clustering techniques. This comparison highlights the advantages of the proposed approach, which does not require time-consuming analysis aimed at preliminary tuning and adjourning the algorithm parameters.

Keywords: Coherency, Network partitioning, Power system stability, Monte Carlo Consensus Clustering (M3C).

1. Introduction

Coherency detection consists in finding groups of generators oscillating, ideally, with the same angular speed, given a disturbance in the system. Power system engineers employ coherency detection in the task of solving many fundamental operation and planning problems, which include disturbance analysis, system separation [1], identification of grid equivalent models [2], [3], optimal placement of Phasor Measurement Units (PMU), and data-driven-based situational awareness [4].

Traditionally, power system coherency has been assessed through model-based methods, which rely on the eigenanalysis of power system's linearized model. The slow-coherency identification method [5] and the method based on eigenvectors [6] are well-known examples of model-based

[1] Corresponding Author, e-mail: *fdecaro@unisannio.it*
[2] e-mail: *apepiciello@unisannio.it*
[3] e-mail: *federico.milano@ucd.ie*
[4] e-mail: *vaccaro@unisannio.it*

approaches. These methods do not consider system configuration changes or disturbance severity and location [7]. The large-scale employment of Wide Area Monitoring Systems (WAMS) has led the transition from model to data-driven based coherency. Indeed, the proliferation of advanced sensing infrastructures improves the stability assessment and control capabilities of power systems. An interesting review of the opportunities and challenges associated to WAMS worldwide can be found in [8].

Despite the large number of methods being proposed in the literature for measurement-based coherency detection, their deployment in realistic operation scenarios is hindered by the need of identifying the number of coherent groups and properly tuning the parameters of the classification algorithm. To solve this challenging issue, in this paper a consensus clustering-based algorithm with a hypothesis testing framework based on Monte Carlo consensus clustering (M3C) is proposed.

1.1. Literature Review

Data-driven methods use real-time measurements obtained from PMUs, characterized by a high sampling frequency.

These methods do not rely on accurate models and detailed parameters to assess coherency, only on the analysis of measured signals [9]. Grouping power system buses into coherent clusters through data requires unsupervised machine learning techniques [10].

The simplest approach to cluster power system buses is to observe the correlation between phase angle or frequency signals measured by PMUs. The authors of [11] use a combination of ten correlation indices in the time domain, whereas authors in [12] and [13] apply the Fourier transform before assessing correlations. In [14], again based on the Fourier transform, the correlation is evaluated considering not only the amplitude, but also the phase angle. Furthermore, the coherency analysis is extended to non generating buses, to define the exact boundaries of the coherent areas. More advanced techniques, suitable for nonlinear and non-stationary time series, are Hilbert-Huang transform [15], wavelet phase difference [16] and Prony analysis [17]. Alternatively, PMU data can be employed to partition coherent areas by defining relevant energy functions [18].

Due to the large dimensionality of the input data, data reduction techniques have been widely applied to data-driven coherency analysis. Principal Component Analysis (PCA) [19], Independent component analysis [20] or Singular Value Decomposition [21] are examples of data reduction techniques used in data-driven coherency analysis. After pre-processing, clustering techniques provide a means to group the measured time series. A wide range of clustering techniques has been proposed in the literature for data-driven coherency detection. Fuzzy c-means [22], k-means [23] and hierarchical clustering [24] are the most common clustering algorithms employed.

The main advantage of the aforementioned works is that they are based on well-known algorithms, applied in a large variety of contexts. They are available and optimized in many packages for several programming languages.

The disadvantages are as follows. First, no data-driven technique has emerged to be the most effective with each data set. Hence, in some scenarios, these algorithms can fail. Second, many of them have no online capabilities. However, this problem is solved in [25]. It proposes a combination of singular value decomposition and k-means for the online identification and the dynamic visualization of coherent areas. Third, traditional clustering techniques require to define *a priori* the number of expected coherent groups. This number is defined based on the knowledge of the system or on a trial-and-error approach. However, fault location and operation conditions may affect the optimal number of clusters.

To solve the last issue, a coherency detection method based on Affinity Propagation clustering is proposed in [4]. This advanced technique does not require the number of clusters to be defined, but its performance is greatly affected by the hyper-parameters setting, which is a demanding issue to address [26].

Another interesting data-driven approach for reliable cluster identification is based on the typicality-based analysis, which is an alternative to the frequentist representation of data distribution [27]. Although this approach shows promising results and has the advantage of not using clustering algorithms, it does not integrate any methods or metrics to validate the effectiveness of the identification of the coherent groups over long time-horizons and variable operating conditions.

To deal with these challenging issues, the quality of clusters is often judged through visual inspection or expert knowledge. In some cases, the results of the clustering techniques are evaluated through similarity metrics, such as Average Silhouette [28], Elbow method or Gap statistics [29]. A description of these and additional internal metrics is provided by [30]. However, these metrics do not consider the strong dependence of clusters on contingencies, network operating conditions and cluster initialization. Finally, existing works do not consider many fault locations for a considered grid. This may lead to an incomplete or even inconsistent identification of the clusters and/or of the clustering technique assessment. In light of these reasons, this manuscript proposes the application of M3C for the coherency detection of power system clusters. The features of this methodology that constitutes an added value for power system practitioners with respect to the state-of-the-art data-driven-clustering algorithms are as follows:

- Internal best cluster number estimation without relying on external metrics.

- The ability to provide a self-consistent validation of the returned clusters based on a Relative

Cluster Stability Index (RCSI) and on statistical evaluations.

Besides the aforementioned features, the proposed M3C-based coherence detection shows a variety of advantages for power system operators, as follows:

- Robustness: the ability to return "more robust" clusters than the traditional clustering algorithms. This is a consequence of the resampling mechanism included in M3C that mitigates the effect of anomalous data on the final clustering.

- Performance: the M3C results are independent of the initial guess on the returned clustering. The initial guess, on the other hand, affects most existing algorithms and, hence, several random reinitializations are often required to obtain good results, which affects performance.

- Consistency: as shown in the case studies of this paper, the M3C returns more consistent (from the dynamic point of view, i.e., "more coherent") clusters than the concurrent methods. This feature can help better understand the dynamic behavior of the system under study, and develop advanced planning and operation-oriented applications based on coherency detection.

Furthermore, the paper presents a comprehensive comparison of the proposed technique with the most commonly utilized data-driven-clustering algorithms. The following parameters and tests are considered:

- Faults across each bus of the grid, given several grid sizes.

- A Friedman-based significance statistical test on the returned clustering scores to establish a performance rank of the clustering algorithms, and to assess their performance under a statistical framework.

For its depth and completeness, the case study allows not only for a fair comparison of the state-of-the-art data-driven clustering algorithms and illustrates the benefits of the proposed M3C technique, but also serves as a benchmark for future works on this topic.

1.2. Organization

The remainder of the paper is organized as follows. Section II provides some background of power system coherency and clustering. Section III outlines the main concepts of Consensus Clustering, required to understand the M3C, introduced in Section IV. Section V describes the case studies, based on the coherency analysis of the IEEE 39-bus system through the M3C and compares it to existing data-driven methods. Finally, Section VI draws the main conclusions and outlines possible future research directions on the topic.

2. Background on Coherency and Clustering

The proposed M3C method uses frequency and phase angle signals, provided by PMUs installed over the network, for coherency identification. In this section, first the definition of coherency and the dissimilarity matrix used in clustering techniques are introduced. Then, the most relevant concepts of Consensus Clustering (CC) and its shortcuts are discussed.

2.1. Coherency

Power system coherency is defined based on phase angle and frequency measurements over the network. Two generators i and j are considered coherent if the following relation (1) holds:

$$\Delta\delta_i(t) - \Delta\delta_j(t) \leq \varepsilon \tag{1}$$

where $\Delta\delta_i(t)$ and $\Delta\delta_j(t)$ are the phase angle deviations of generator i and j after a disturbance. The parameter ε represents the tolerance set to declare two generators coherent. A larger value leads to loose coherent groups, a smaller value to tight groups, whereas $\varepsilon = 0$ corresponds to the ideal coherency. Since the frequency f is obtained as the time derivative of phase angle, assuming a constant numeric derivation step in its calculation, it is interchangeable with phase angle δ to check coherency. Note also that, if PMUs are installed not only at generator buses, but wherever on the network, the coherency concept can be extended to generic buses.

2.2. Time Series Clustering

Clustering is an unsupervised learning process aimed at partitioning a set of N items into k homogeneous groups, according to a chosen similarity metric. Time-series clustering performs the partitioning of signals measured over a time interval, where each signal has T samples, depending on its sampling frequency. In coherency detection, the measured data can be represented as a matrix $\mathbf{M}$, whose dimensions are $[N \times T]$. Each row represents the frequency measurements of a PMU installed on a bus, for a total number of N buses in the network. Each column represents a measurement sample at time t. Hence, the dimension of the matrix is highly dependent on the number of PMUs in the network, the sampling frequency and the observation interval.

The majority of deployed clustering algorithms in coherency detection, such as hierarchical clustering, requires computing a dissimilarity matrix $\mathbf{D}$, which is a symmetric matrix of order N. Each element of $\mathbf{D}$ represents the distance between a pair of items of matrix $\mathbf{M}$. Different distance metrics can be employed, such as correlation, dynamic time warping, or Euclidean distance. The latter is the most common choice. Other clustering algorithms, such as k-means, require as input the matrix $\mathbf{M}$ in the original / novel space instead of the distance matrix $\mathbf{D}$. These aspects will be analyzed more in depth in the rest of the manuscript.

3. Consensus Clustering

High dimensional data, with a low ratio between the number of items and the number of features per item, results in clusters sensitive to noise and susceptible to over-fitting. Furthermore, many algorithms require random initialization, which may lead to different results over many trials.

For this reason, the authors of [31] defined the concept of cluster stability. It is linked to the robustness of clusters to resampling. Larger robustness in resampling is associated with high confidence in considering the clusters as representative of the real data structure. The estimation of the best number of stable clusters, given a span $\{1, \ldots, K\}$, is conducted by considering R resamplings per k.

3.1. Subset resampling

For each $k \in \{1, K\}$, CC runs R clustering trials on bootstrapped subsets of the original dataset. Given the r-th subset, the CC performs a clustering algorithm to split the elements. Hence, "consensus matrix" $\mathbf{C}_{(r)}$, which is a square matrix of order N, is computed from the knowledge of the returned clusters as follows:

$$\mathbf{C}_{(r)}(i,j) = \begin{cases} 1 & \text{if items } i \text{ and } j \text{ are in the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Similarly, "connectivity matrix" $\mathbf{I}_{(r)}$, which is a square matrix of order N, is computed as follows:

$$\mathbf{I}_{(r)}(i,j) = \begin{cases} 1 & \text{if items } i \text{ and } j \text{ are bootstrapped together} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Particularly, $\mathbf{I}_{(r)}$ encloses information about the bootstrapped items for each r-th trial.

3.2. Information fusion

After the R trials, the consensus matrices are processed to compute the "Overall consensus matrix" $\mathbf{\Gamma}_{(k)}$:

$$\mathbf{\Gamma}_{(k)}(i,j) = \frac{\sum_{r=1}^{R} \mathbf{C}_{(r)}(i,j)}{\sum_{r=1}^{R} \mathbf{I}_{(r)}(i,j)} \tag{4}$$

Given the r-th trial, the connectivity matrix shows which elements are clustered together, whereas the Overall Consensus matrix represents the tendency of elements i and j to be clustered together, given different trials with changing subpopulations. (4) normalizes the membership grade between two items considering the resampling. For this reason, the elements of $\mathbf{\Gamma}_{(k)}$ are continuous values between 0 and 1.

3.3. Estimation of the Best Number of Clusters

The best number of clusters is assessed by analyzing the constructed overall consensus matrices for each k. The first step requires the computation of the Cumulative Distribution Function (CDF) score for each k, $\mathrm{CDF}_{(k)}$. A robust cluster corresponds to the elements of $\mathbf{\Gamma}_{(k)}$ being equal to either 0 or 1. On the other hand, if the clusters are sensitive to resampling, the elements of $\mathbf{\Gamma}_{(k)}$ are equal to the intermediate values between 0 and 1. The computation of the empirical $\mathrm{CDF}_{(k)}$ for each k is a graphical indicator of the best number of clusters. The empirical $\mathrm{CDF}_{(k)}$ is computed as in (5):

$$\mathrm{CDF}_{(k)}(q) = \frac{\sum_{i<j} 1\{\mathbf{\Gamma}_{(k)}(i,j) \leq q\}}{N(N-1)/2} \tag{5}$$

where $1\{\dots\}$ is the indicator function, which is 1 if the condition in the braces holds, and 0 otherwise, and q is an auxiliary variable, whose range is $[0,1]$. A flatter $\mathrm{CDF}_{(k)}$ results in more robust clusters, for a given k. The indicator function is computed for increasing discrete steps of q.

The search for the best number of clusters relies on the analysis of the differences between different $\mathrm{CDF}_{(k)}$. The area subtended to $\mathrm{CDF}_{(k)}$, $A_{(k)}$, is equal to (6):

$$A_{(k)} = \sum_{i=2}^{m} [x_i - x_{i-1}] \mathrm{CDF}_{(k)}(x_i) \tag{6}$$

where $x = \{x_1, \dots, x_m\}$ are the sorted entries of $\mathbf{\Gamma}_{(k)}$. The difference between the subtended area of different $\mathrm{CDF}_{(k)}$, i.e. the $\Delta_{(k)}$ score, is computed as follows:

$$\Delta_{(k)} = \begin{cases} A_{(k)} & \text{if } k = 2 \\ \dfrac{A_{(k+1)} - A_{(k)}}{A_{(k)}} & \text{if } k > 2 \end{cases} \tag{7}$$

The optimal number of clusters results in the maximum value of $\Delta_{(k)}$. Once the best number of clusters k^* is returned, the corresponding $\mathbf{\Gamma}_{(k^*)}$ plays the role of distance matrix in an agglomerative Hierarchical Clustering, producing the final clustering, where the dendrogram is cut at the height corresponding to k^* clusters.

3.3.1. Limitations of Consensus Clustering

The conventional CC presents some critical issues. In particular, the $\Delta_{(k)}$ score is uninformative in some circumstances, even in the presence of well-defined structures. For this reason, reference [32] proposed an alternative to $A_{(k)}$ and $\Delta_{(k)}$ scores, called "Proportion of Ambiguous Clustering" (PAC), as follows:

$$\mathrm{PAC}_{(k)}(v_1, v_2) = \mathrm{CDF}_{(k)}(v_2) - \mathrm{CDF}_{(k)}(v_1) \tag{8}$$

where $v_{1,2} \in]0,1[$ are the sub-interval bounds. The literature suggests to set $v_1 = 0.05$ and $v_2 = 0.95$. Hence, the lower is the $\mathrm{PAC}_{(k)}$ the more evident is the structure found in the data, meaning that

Algorithm 1 CC Algorithm

```
▷ Input: frequency oscillation time series for each bus, algorithm hyper-parameters.
for $k \in \{1, \ldots, K\}$ do
    for $r \in \{1, \ldots, R\}$ do
        Run the internal clustering algorithm on a subpopulation of $\mathbf{M}$;
        Compute the "consensus matrix" $\mathbf{C}_{(r)}$ and the "connectivity matrix" $\mathbf{I}_{(r)}$;
    end for
    Compute the "Overall consensus matrix" $\mathbf{\Gamma}_{(k)}$;
    Compute the CDF score for $\mathbf{\Gamma}_{(k)}$;
    Compute the $\Delta_{(k)}$ score.
end for
Find $k^*$, which is the best number of clusters maximizing $\Delta_{(k)}$.
Perform Hierarchical Clustering over $\mathbf{\Gamma}_{(k^*)}$.
```

the elements of $\mathbf{\Gamma}_{(k)}$ are distributed close to 0 and 1, respectively. The best number of clusters k returns the lowest $\mathrm{PAC}_{(k)}$.

The $\mathrm{CDF}_{(k)}$ of the overall consensus matrix shows the distribution of its elements after n trials, given a number of clusters. Lower values of such elements mean that items are often clustered separately, higher ones mean that items are often clustered together. Given k, a perfect cluster separation, which returns always the same groups for all resamplings, corresponds to the case where the element distribution of the overall consensus matrix is binary (case A) as shown in Fig. 1. Lesser robust clusters to resampling produce intermediate elements in the range $]0, 1[$ (case B). $\mathrm{PAC}_{(k)}$ is a metric analyzing the consensus membership values falling within this intermediate range, which corresponds to items considered "ambiguous" since they are not always clustered together in B trials. Although the introduction of this novel score improves the original consensus cluster scores, statistical tests to validate the PAC score and the corresponding best number of clusters are required.

4. Monte Carlo Consensus Clustering

This paper proposes the application of M3C to the coherency detection analysis in power systems. M3C was proposed by the authors of [33], which showed to outperform the original CC by employing statistical tests to validate the best number of clusters. Similarly to CC, the idea behind M3C is to measure the strength of clusters assessing their tendency to form stable groups in the presence of elements resampling. Furthermore, M3C introduces a statistical-based procedure to suggest the optimal number of clusters, based on the work of [34], to overcome the CC limitations in estimating

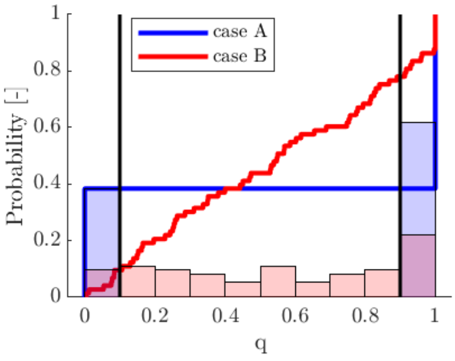


Figure 1: Examples of the experimental distribution, and corresponding empirical CDF for a perfect cluster separation (case A), and imperfect cluster separation (case B). Data processed are the elements of the overall consensus matrix. The vertical black lines are $v_1 = 0.10$ and $v_2 = 0.9$, which are used to compute the PAC score in this example. To improve the readability of the figure, the values of v_1 and v_2 differ from those in [33].

the best number of clusters as shown by [32]. The conducted statistical analysis compares the expected and actual score. The former is computed on a set of fictional data having statistical properties to the original ones, whereas the latter on the real clustered elements. In particular, M3C integrates the original CC with PAC scores according to the algorithm 2. The details of the steps are explained below.

4.1. Principal Component Analysis

The PCA is the most popular method to perform cardinality reduction. It consists in the linear combination of the original variables to produce new ones. The original variables are combined producing a new space with new axes (called Principal Components) orthogonal to the original ones, and oriented in the direction which maximizes the variance.

The algorithm requires the following steps: (i) transform the original matrix $\mathbf{M}$ in $\bar{\mathbf{M}}$, which has a null mean and unitary variance for each variable; (ii) compute the Singular Value Decomposition through $\bar{\mathbf{M}} = \mathbf{ULA}^T$, where $\mathbf{U}$, whose dimensions are $[N \times N]$, is the left eigenvector matrix, $\mathbf{L}$, whose dimensions are $[N \times T]$, is the rectangular diagonal matrix, and $\mathbf{A}$, whose dimensions are $[T \times T]$, is the right eigenvector matrix, and where the columns of $\mathbf{A}$ are the eigenvectors of matrix $\mathbf{M}^T\mathbf{M}$; iii) Compute the novel variables in the PCA domain through the relation $\mathbf{S} = \mathbf{MA}$.

4.2. Generation of the random dataset in M3C

The application of PCA on matrix $\mathbf{M}$ allows obtaining the orthogonal eigenvector matrix $\mathbf{A}$ according to $\bar{\mathbf{M}} = \mathbf{ULA}^T$. The latter is linked to the principal component matrix $\mathbf{S}$ through the relation $\mathbf{S} = \mathbf{MA}$. Particularly, a normal distribution, with null mean and standard deviation for each column of $\mathbf{S}$, is considered for filling the columns of $\mathbf{S}_{(b,k)}$. The knowledge of this matrix allows computing the estimation of $\bar{\mathbf{M}}_{(b,k)}$, which preserves the same correlation structure as $\bar{\mathbf{M}}$, as shown in (9):

$$\bar{\mathbf{M}}_{(b,k)} = \mathbf{S}_{(b,k)}\mathbf{A}^T \tag{9}$$

where $\mathbf{S}_{(b,k)}$ and $\mathbf{M}_{(b,k)}$ are the b-th generated score and data in the original domain matrix, respectively, for the k-th number of clusters. Then $\bar{\mathbf{M}}_{(b,k)}$ produces $\hat{\mathbf{M}}_{(b,k)}$ by considering variance and mean of $\mathbf{M}$. The generated data are used to compute the expected clustering quality score for each k as proposed by [34], where this strategy is shown to overcome the limits of the heuristic approaches as the elbow method. In this case, the cluster quality index utilised in this work is the PAC score (see Section 4.3).

4.3. RCSI index computation

The Monte Carlo procedure aims at simulating datasets with the same correlation structure as the original ones. The idea is to produce a distribution of $\mathrm{PAC}_{(k)}$ scores, given a number of clusters k. The different reference $\mathrm{PAC}_{(k)}$, for each k, are compared to the real PAC score, $\mathrm{PAC}_{(k)}^{\mathrm{real}}$ to highlight presence of a data structure. The $\mathrm{RCSI}_{(k)}$, (10), is the ratio between the average reference $\overline{\mathrm{PAC}}$, (11), and the real PAC score:

$$\mathrm{RCSI}_{(k)} = \log_{10}\frac{\overline{\mathrm{PAC}}_{(k)}}{\mathrm{PAC}_{(k)}^{\mathrm{real}}} \tag{10}$$

$$\overline{\mathrm{PAC}}_{(k)} = \frac{1}{B}\sum_{b=1}^{B}\mathrm{PAC}_{(b,k)}^{\mathrm{ref}} \tag{11}$$

$\overline{\mathrm{PAC}}_{(k)}$ is the mean value of all the $\mathrm{PAC}_{(b,k)}^{\mathrm{ref}}$ for each k, where $\mathrm{PAC}_{(b,k)}$ defines the PAC scores computed for each generated b-th dataset. In particular, if $\overline{\mathrm{PAC}}_{(k)}/\mathrm{PAC}_{(k)}^{\mathrm{real}} > 1$, the structure found in real data is more evident than the hypothetical structure found considering B similar datasets. For this reason, a greater $\mathrm{RCSI}_{(k)}$ index results in a more evident structure, given k.

4.4. Computation of Monte Carlo p-value

The PAC score of the best number of clusters k^*, identified through the RCSI index, is processed by a statistical test for validation purposes. The Monte Carlo test is a non-parametric test that is useful when there is no knowledge on the shape of the observed test statistic distribution. This test generates independent random datasets under the null hypothesis H_0, where an independent statistical test is conducted for each of the generated datasets. The test statistic is the PAC score in this specific case. Ideally, the test requires an infinite number of B trials. The p-value is computed as $p_{(B=\infty)} = Pr(\mathrm{PAC}^{\mathrm{ref}} \geq \mathrm{PAC}^{\mathrm{real}})$. To provide an unbiased estimation of this probability a large number of trials must be set, where the p-value is computed according to the ratio between the number of times $\mathrm{PAC}^{\mathrm{ref}}_{(b)} \geq \mathrm{PAC}^{\mathrm{real}}$ and B in the original version of the Monte Carlo test [35]. In M3C, the calculation of p-value is adjusted by [36] to avoid zero p-values.

Particularly, the p-value $p_{(k^*)}$, computed according to (12), is employed for the Monte Carlo test:

$$p_{(k^*)} = \frac{\sigma_{(k^*)} + 1}{B + 1} \tag{12}$$

$$\sigma_{(k^*)} = \sum_{b=1}^{B} 1\{\mathrm{PAC}^{\mathrm{ref}}_{(b,k^*)} < \mathrm{PAC}^{\mathrm{real}}_{(k^*)}\} \tag{13}$$

The parameter $\sigma_{(k^*)}$ is the PAC score threshold, in (13) is the number of reference PAC scores lesser than the real one. $p_{(k^*)}$ is the input of the following statistical test,where α is the confidence level threshold. If H_0 holds, the data are a single Gaussian cluster ($k^* = 1$), otherwise the data are not a single Gaussian cluster. According to [33], the statistical test cannot be used to select which k is the best number of clusters, but it is used to test if k^*, returned from the analysis of RCSI score, assures that the null hypothesis (the elements are all similar and a unique cluster exists) is rejected (and the k^* clusters exist). Further discussion about the interpretation of statistical tests in clustering applications can be found in [37].

5. Case Studies

The proposed M3C method for finding a stable optimal number k of coherent groups is now applied to the IEEE 39-bus system. The aim of these case studies is to show the superior performance of the proposed method, the M3C, with respect to the ones in the literature, in performing the clustering while assessing the optimal number of clusters k. The chosen benchmark methods are k-means, Fuzzy-C-Means (FCM), and the original Monti Consensus Clustering (CC). k-means and FCM are supported by the average silhouette score to select the best number of clusters.

The dynamic simulations, which provide the input data for the analysis, were performed through the software Dome [38]. The clustering algorithms were implemented in the scripting language R, version 3.4.1 and were carried out on a workstation equipped with an i7-7900HQ CPU and 16 GB RAM.

5.1. Clustering of the IEEE-39 bus system

The input data consist of frequency measurements from each bus, supposed to be equipped with a PMU. The time step is equal to 10 ms and the observed time window is 10 s. The generators, which are installed from bus 30 to 39, are represented by a fourth-order model. Automatic Voltage Regulators and Power System Stabilizers are also included in the model.

Algorithm 2 M3C Algorithm

```
▷ Input: frequency oscillation time series for each bus, algorithm hyper-parameters.
Perform the PCA to obtain the orthogonal matrix of eigenvectors $\mathbf{A}$, given the data matrix $\mathbf{M}$ (Sec. 4.1);
for $k \in \{1, \ldots, K\}$ do
▷ Iterate over the number of cluster span;
    for $b \in \{1, \ldots, B\}$ do
▷ Perform the Monte Carlo simulation, where $B$ is the total number of Monte Carlo trials and $b$ the individual trial;
        Generate a random score matrix based on the statistical properties of the real one;
        Generate a $\hat{\mathbf{M}}_{(b,k)}$ matrix data, which is used as input for the CC algorithm, given $k$ (Sec. 4.2);
        Return the $\mathrm{PAC}^{\mathrm{ref}}_{(b,k)}$ score;
    end for
    Perform the CC on real data, given $k$;
    Return $\mathrm{PAC}^{\mathrm{real}}_{(b,k)}$;
    Compute the $\mathrm{RCSI}_{(k)}$ index, given $k$ (Sec. 4.3);
end for
Return the best number of clusters $k^*$ and the groups;
Compute the Monte Carlo $p$-value ($P$-score) (Sec. 4.4).
```

The first case study refers to the coherency analysis of the network subject to a three-phase fault at bus 27, with a clearing time equal to $t_{\text{clear}} = 0.2\ s$, which is a realistic value. Table 1 shows the setup of the M3C algorithm, where R_{real} and R_{ref} are the number of bootstrapping for the real and reference datasets, respectively.

Before choosing these parameters, a sensitivity analysis is conducted to assess their impact on the results provided by M3C. The outcomes of this analysis show that M3C results are stable with respect to changes in parameters B, R_{real}, and R_{ref}, even for a low number of Monte Carlo simulations. This is consistent with the results of reference [33].

Ideally, the parameter K, which is the maximum number of clusters considered, should be set equal to N. However, in the presence of a large number of elements, it is preferable to set $K << N$. Furthermore, the case $K = N$ corresponds to the unclustered case, which has no practical utility.

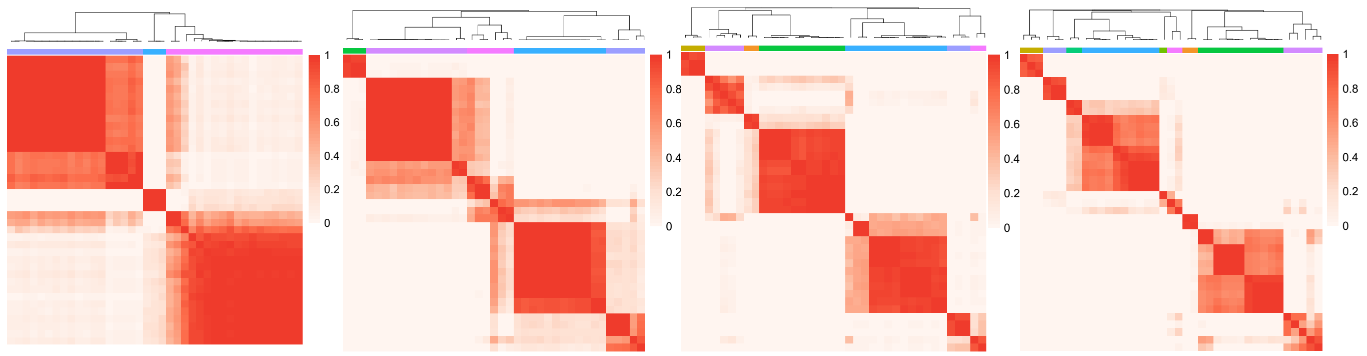
The literature does not suggest a criterion to set the maximum K in the iterative approach; it should be selected depending on the specific application. For example, in this case, the value chosen is $K = 10$ since the number of buses is not very large and a larger number of clusters would be useless for any practical application.

Table 1: Input parameters of M3C

B	R_{real}	R_{ref}	K	v_1	v_2	α
100	100	100	10	0.05	0.95	0.005

5.2. Fault at bus 27

Finding the optimal number of clusters requires building the overall consensus matrix $\mathbf{\Gamma}_{(k)}$ for different numbers of clusters k, as illustrated in Fig. 2. First, a $\text{CDF}_{(k)}$ representing the distribution of matrix elements is calculated for each $\mathbf{\Gamma}_{(k)}$ (Fig. 3a). Then, starting from different $\text{CDF}_{(k)}$, the $\text{PAC}_{(k)}$ scores are computed and the optimal number of clusters is identified by selecting the minimum score (Fig. 3b). Finally, this choice is validated by computing the $\text{RCSI}_{(k)}$ index (Fig. 3c), and performing a statistical test (Fig. 3d). The optimal number of clusters returned by this procedure is $k^* = 7$.

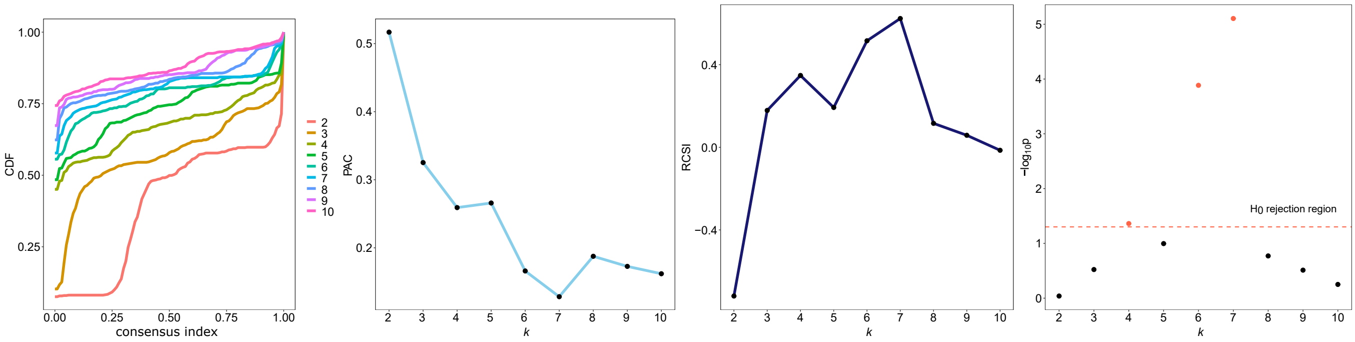


(a) $k = 3$ (b) $k = 5$ (c) $k = 7$ (d) $k = 9$

Figure 2: (2a)-(2d) Overall consensus matrices applied to the coherency analysis of IEEE-39 bus system for different numbers of clusters.

The frequency signals of each bus, grouped by cluster, are shown in Fig. 4a. The partitioned system is represented as a graph in Fig. 4b. The edges are weighted considering the admittances of the network.

The results show that a fault at bus 27 causes large frequency oscillations $\Delta f \approx 0.01$ pu at the set of buses $\{28, 29, 38\}$. These are probably due to the low admittance of the line connecting buses 25 and 26, the only one connecting them to the main network during the fault. Furthermore, buses $\{1, 9, 39\}$ form a cluster with a higher frequency of oscillation than the other coherent groups. Clusters from 3 to 6 seem to be well partitioned from a first visual inspection of frequency signals, although



(a) (b) (c) (d)

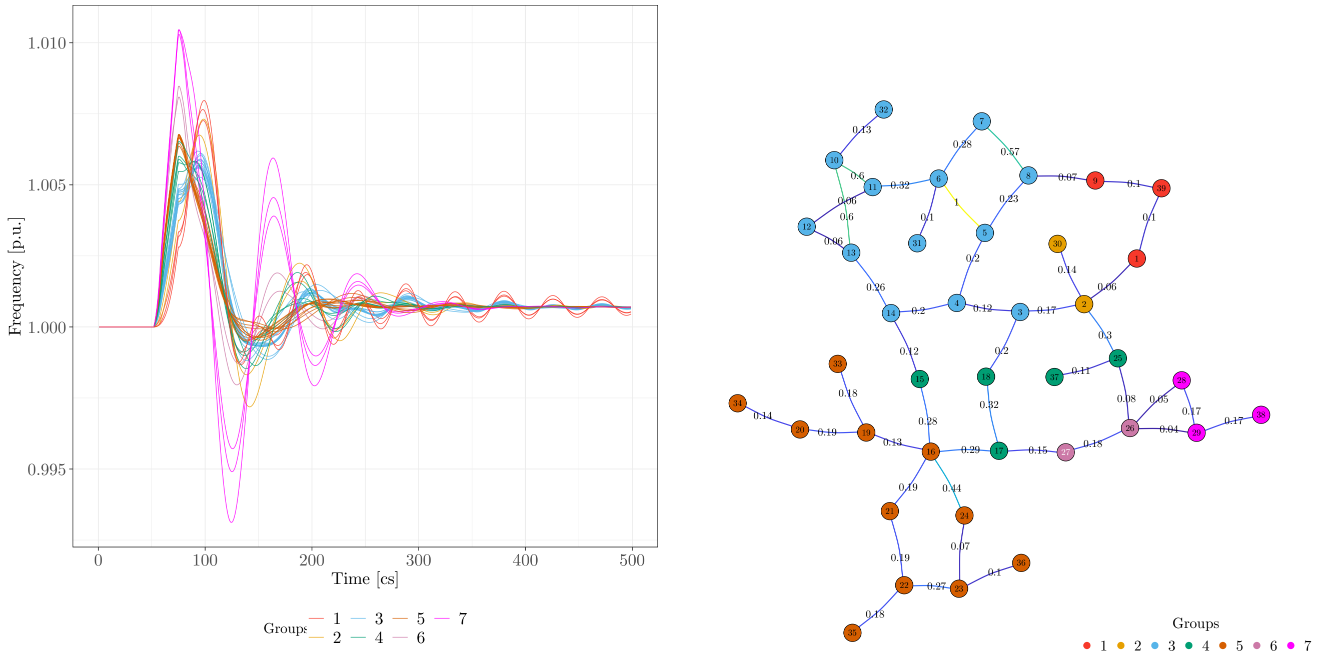
Figure 3: (a) CDFs curve of the elements of $\mathbf{\Gamma}$, (b) PAC score, (c) RCSI index, and (d) P-score applied to the coherency analysis of IEEE-39 bus system, for different k.

they all have similar amplitude and frequency of oscillation. Additional cluster validation can be obtained by reducing the dimension of the input data through a PCA and plotting the resulting data points on a two-dimensional plot. The results of PCA are shown in Fig. 5. In this figure, clusters 1, 2 and 7 have a larger inter-cluster distance from the other clusters. Clusters from 3 to 6 are closer to each other.

Figure 5 also shows the results of the benchmark clustering techniques, in order to compare them to the proposed M3C method. Particularly, Fig. 5 shows the bi-dimensional scatter plot of clusters in two dimensions in the PCA domain. It is possible to observe that CC produces only two clusters, where one is characterized by a large spread over the PC2 axis. FCM splits this large group into two parts. Differently, k-means and M3C produced more complex partitions, which makes it more difficult to interpret which method is better.

The bi-dimensional scatter plot may not be exhaustive when additional principal components are required to correctly represent data. Indeed, the three-dimensional scatter plot shows the separations of buses {39,9,1} from {2,30}. Hence, the closeness of the elements is only apparent and caused by space-flattening. Fig. 6 shows that these groups are on a different plane over the 3-rd PC. However, k-means and FCM do not return a bad clustering for this particular fault scenario, hence additional analysis can lead to suggest that M3C may be the better option. This can be validated through the Visual Assessment of Tendency (VAT), introduced by [39] and first used in the field of coherency grouping by [40].

The VAT method is not a clustering method but a way of sorting the distance matrix, such that similar signals are kept close together and are clearly visible. In this case, the distance matrix between the signals is calculated as the sum of the euclidean norms of each point in time. The result of VAT, using the command *imagesc* in MATLAB® is shown in Figs. 7a-7d. The black



(a)

(b)

Figure 4: Frequency signals (Fig. 4a) and Graph of the network clustered through M3C (Fig. 4b), fault at bus 27 of IEEE-39 bus system. The colors of signals (Fig. 4a) and nodes (Fig. 4b) identify the clusters, the white-colored font identifies the fault location, and the labels of the edges identify the line admittance, which are normalized with respect to the maximum admittance value. For the sake of clarity, the latter are not an input of any clustering model. They are used to improve the physical interpretability of the grid graph.

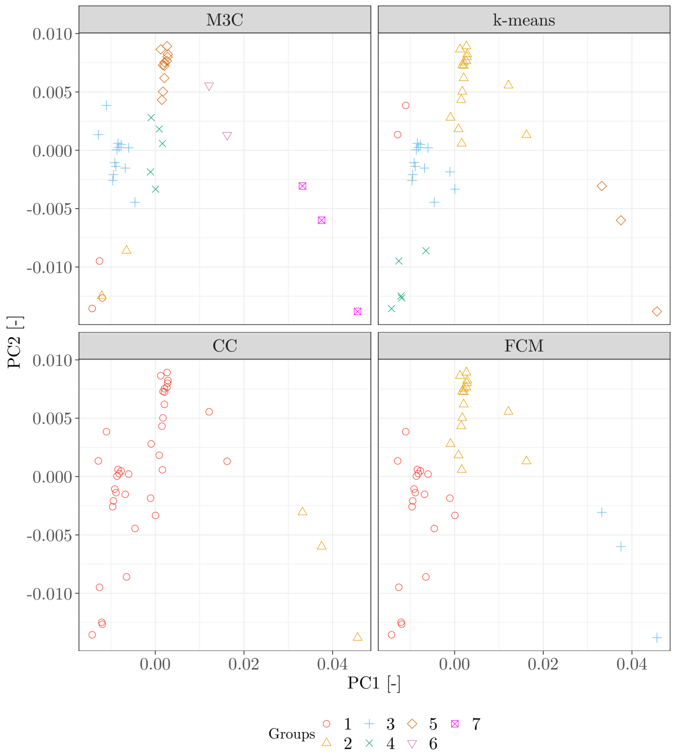


Figure 5: Visualization of the frequency signals reduced through PCA 2D, fault at bus 27 of IEEE-39 bus system.

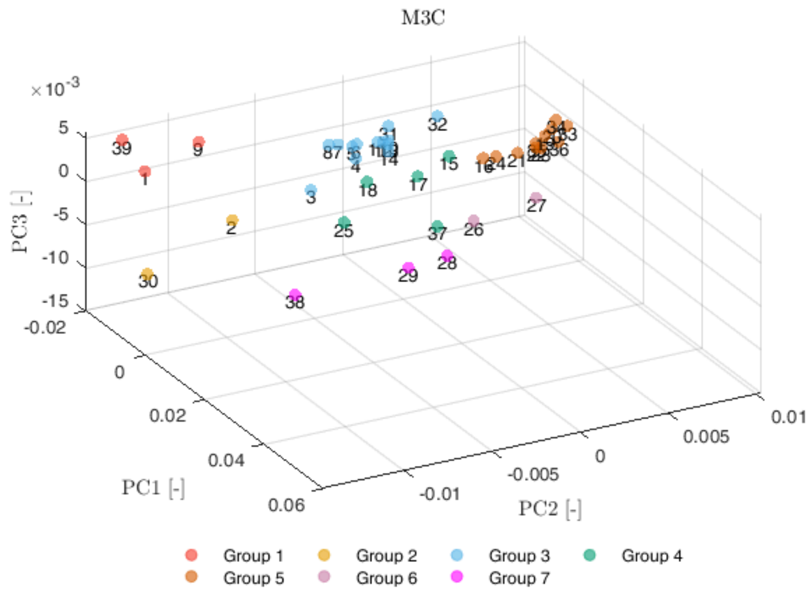


Figure 6: Visualization of the frequency signals reduced through PCA 3D, fault at bus 27 of IEEE-39 bus system.

color indicates a small distance between buses, while the red color denotes maximum distance. For example, the group formed by {28, 29, 38}, which is the one with the larger amplitude of the oscillations, is clearly more distant from the others.

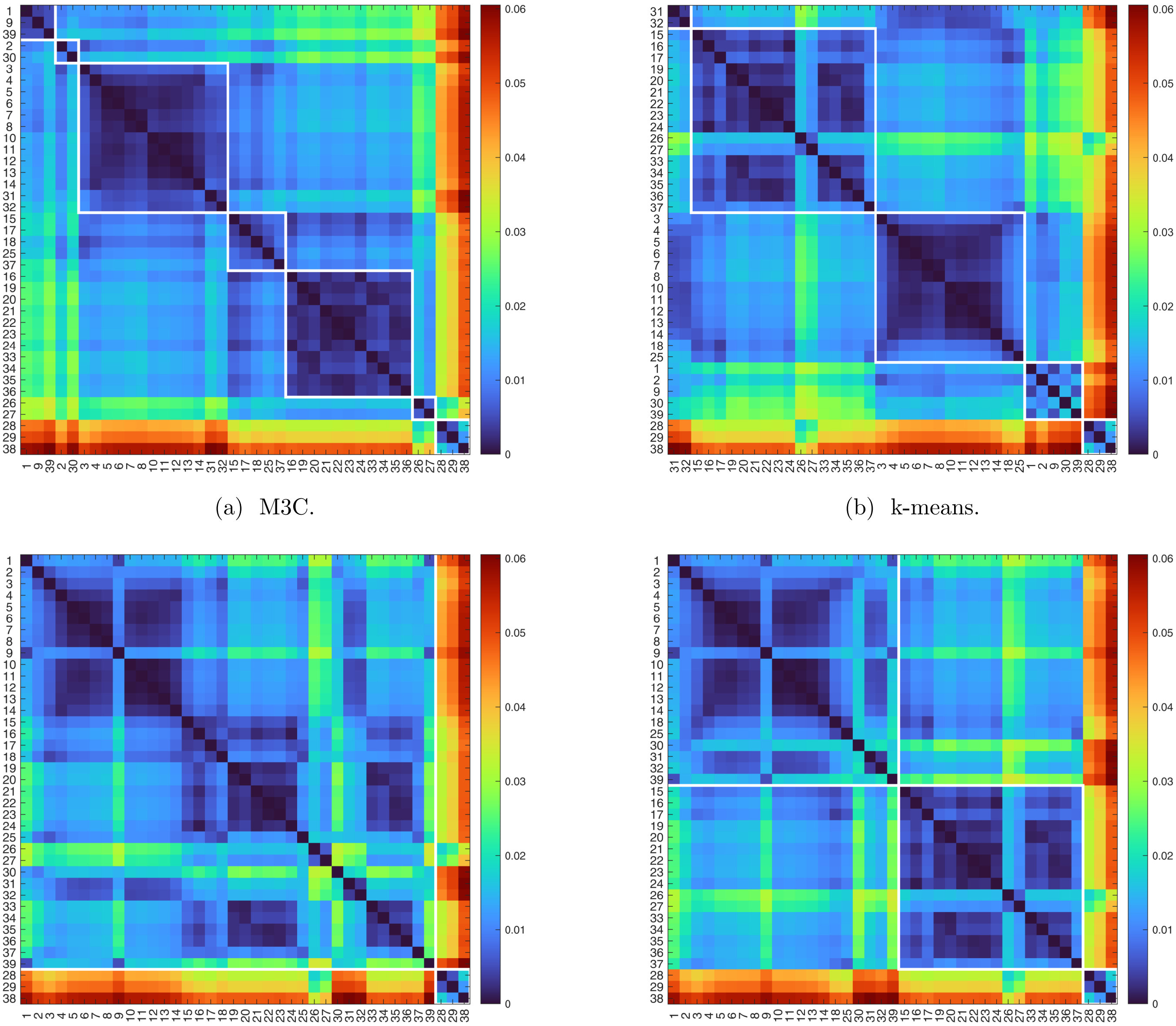
The VAT shows how the M3C partition in 7 clusters is better than the benchmark methods. With k-mean clustering, buses {26, 27} in group 2 are badly matched, as it is clear from Fig. 7b. Also, group 4 containing buses {1, 2, 9, 30, 39} should be split. With CC, the buses are split into two groups, where the former includes 36 buses, and the latter three. The VAT analysis suggests the largest groups include clearly visible separate bus sub-groups {1-14} and {15-25}, {26, 27}, etc., which must be split. Finally, with FCM, buses {26 and 27} are joined in the group 2 {15-37}, whereas VAT suggests splitting them.

5.3. Coherency metrics

Clustering validation is not an easy task, especially in absence of the so-called *ground truth*, which allows external validation. Different metrics, defined in the supplementary material, are tested, each one with its own advantages and flaws. For coherency grouping, a quantitative metric that reports numerically what is shown by the visual inspection is of interest. The first metric tested is the Average Silhouette, the most common internal validation metric used in clustering. Unfortunately, this metric does not perform well on coherency analysis in the time-domain. The main reason is due to its large dependence on the inter-cluster distance. For example, if the distance between one group and all the others is very large, it is likely that the maximum value of the Average Silhouette is reached at $k = 2$ optimal number of clusters.

Since coherent groups in power systems should be characterized by a low intra-cluster distance and low dispersion, a more suitable metric such as the Calinski Harabasz (CH) is calculated. The calculation of this metric for each fault location is reported in Table 2.

The following insights can be extracted from Table 2: M3C outperforms (43% of the total number of cases), or confirms (48%), the results of k-means, which prevails only for the 3% of the total cases. Furthermore, when M3C outperforms k-means, the improvements in CH score are remarkable. The reduction of CH score in the cases in which k-means provided a better cluster than M3C is lower than 20%. Further insights concern the M3C and CC, where the former supplies better CH scores for



(a) M3C.

(b) k-means.

(c) Consensus Clustering (CC).

(d) Fuzzy-C-Means.

Figure 7: Distance matrix plot as obtained by (a) M3C, (b) k-means, (c) Consensus clustering, and (d) Fuzzy-C-Means, fault at bus 27 of IEEE-39 bus system.

67% of cases, compared with 5% where CC is better than M3C. The improvements of MC3 over CC are remarkable, with some cases showing gains greater than 100%. Conversely, CC improvements in M3C are always lower than 11%. Finally, M3C outperforms the CH scores of FCM for 67% of the cases. Conversely, FCM performs better than M3C for only 3% of the cases.

Table 2: Results on Clustering Quality, IEEE-39 Bus System

Fault Location	M3C			k-means				Consensus Clustering (CC)				Fuzzy C-Mean (FCM)			
bus:	Cluster Nr.:	Calinsky Harabasz	CTC	Cluster Nr.:	Calinsky Harabasz	CTC	M3C vs k-means CH change [%]	Cluster Nr.:	Calinsky Harabasz	CTC	M3C vs CC CH change [%]	Cluster Nr.:	Calinsky Harabasz	CTC	M3C vs FCM CH change [%]
1	4	46.12	1.000	4	46.12	1.000	0	2	22.58	1.000	104	9	27.01	0.986	71
2	4	48.52	0.934	2	31.23	1.000	55	2	31.23	1.000	55	5	41.67	0.846	16
3	4	55.95	0.888	5	69.49	0.960	-19	2	48.24	0.875	16	3	50.93	0.794	10
4	3	42.51	0.984	3	11.70	0.833	255	2	16.15	1.000	163	3	39.33	1.000	8
5	3	46.42	1.000	3	46.42	1.000	0	2	10.21	1.000	355	2	34.31	1.000	35
6	3	44.08	1.000	3	44.08	1.000	0	2	9.59	1.000	360	2	35.20	1.000	25
7	3	55.84	1.000	3	30.40	0.833	84	3	55.84	1.000	0	2	47.62	1.000	17
8	3	62.12	1.000	3	31.11	0.833	100	3	62.12	1.000	0	2	53.05	1.000	0
9	2	46.24	1.000	3	29.16	0.833	59	2	16.69	1.000	177	2	46.24	1.000	0
10	2	37.20	1.000	2	37.20	1.000	0	2	10.39	1.000	258	2	37.20	1.000	0
11	2	37.03	1.000	2	37.03	1.000	0	2	7.31	1.000	407	2	37.03	1.000	6
12	2	46.01	1.000	2	46.01	1.000	0	4	35.95	1.000	28	2	43.47	1.000	0
13	2	32.70	1.000	4	32.47	0.988	1	2	8.24	1.000	297	2	32.70	1.000	0
14	9	52.09	0.981	6	42.82	0.976	22	5	31.65	1.000	65	9	52.09	0.981	0
15	2	91.06	1.000	2	91.06	1.000	0	2	91.06	1.000	0	2	91.06	1.000	0
16	2	142.05	1.000	2	142.05	1.000	0	2	142.05	1.000	0	2	142.05	1.000	0
17	2	93.71	1.000	2	93.71	1.000	0	2	93.71	1.000	0	2	93.71	1.000	-10
18	2	71.17	1.000	4	67.22	1.000	6	4	67.22	1.000	6	3	79.24	0.917	0
19	3	131.35	1.000	2	122.19	1.000	8	2	122.19	1.000	8	3	131.13	1.000	181
20	3	85.84	0.889	2	96.02	1.000	-11	2	96.02	1.000	-11	2	30.57	1.000	0
21	2	91.27	1.000	2	91.27	1.000	0	3	82.88	1.000	10	2	91.27	1.000	59
22	9	78.17	0.898	2	54.76	1.000	43	2	68.60	1.000	14	2	49.16	1.000	67
23	3	74.73	1.000	2	49.27	1.000	52	2	62.58	1.000	19	2	44.75	1.000	0
24	2	121.70	1.000	2	121.70	1.000	0	2	121.70	1.000	0	2	121.70	1.000	0
25	5	40.07	1.000	3	35.17	1.000	14	2	10.72	1.000	274	4	30.86	1.000	30
26	2	107.39	1.000	2	107.39	1.000	0	2	107.39	1.000	0	5	69.35	0.875	55
27	7	89.54	0.914	5	58.21	0.872	54	2	50.89	1.000	76	3	74.65	0.979	20
28	2	143.64	1.000	2	143.64	1.000	0	2	143.64	1.000	0	2	113.45	1.000	27
29	2	131.11	1.000	2	131.11	1.000	0	2	131.11	1.000	0	2	34.14	1.000	284
30	4	46.92	1.000	3	39.30	1.000	19	2	13.73	1.000	242	2	23.84	1.000	97
31	2	58.16	1.000	2	14.96	1.000	289	2	14.96	1.000	289	2	58.16	1.000	0
32	2	51.69	1.000	2	51.69	1.000	0	2	12.24	1.000	322	2	51.69	1.000	0
33	3	74.76	1.000	2	79.25	1.000	-6	2	79.25	1.000	-6	2	30.51	1.000	145
34	2	103.11	1.000	2	103.11	1.000	0	2	103.11	1.000	0	3	97.48	1.000	6
35	3	72.75	1.000	2	63.88	1.000	14	3	40.02	1.000	82	2	50.11	1.000	45
36	2	55.84	1.000	2	55.84	1.000	0	2	27.85	1.000	100	2	32.74	1.000	71
37	5	50.22	0.969	5	18.03	0.869	179	2	22.01	1.000	128	2	15.43	1.000	225
38	2	140.27	1.000	2	140.27	1.000	0	2	140.27	1.000	0	2	115.97	1.000	21
39	2	225.28	1.000	2	225.28	1.000	0	2	225.28	1.000	0	3	213.22	0.900	6

There are cases where M3C and the benchmark models return the same output. This means the benchmark models' clusters are robust in those cases. Hence M3C validates the results of the former algorithm. For this reason, when the clusters are equal between M3C and the benchmark models this must not be considered as a negative aspect of M3C.

5.4. Friedman test with post-hoc analysis

The percentage CH score gained/lost by M3C over the benchmark clustering models allows for an easy point comparison for each case study. A limitation of this approach is that is does not return a judgment about the overall performance of the analyzed clustering models.

On the other hand, a fair comparison must consider the performance of all models for each case study given a referring metric, such as the CH score in this case. The Friedman test with a post-hoc analysis returns crucial output to assess the performance of machine learning models [41].

The Friedman test is a rank-based non-parametric multiple comparison between several populations in the presence of repeated measures [42]. Some recent applications in power systems of this test lies in the model performance comparisons for wind power [43], load [44] and irradiance forecasting [45] applications. Moreover, the Friedman test is a randomized block analysis of variance, whose null hypothesis is that the methods' performance has the same distribution. Given a confidence threshold α_{FT}, if the test rejects the null hypothesis H_0, it is possible to assess if the models exhibit different performances. In this case, a *post hoc* analysis reveals which pair of methods is significantly different. In this case, Tukey's test performs the *post hoc* analysis. Particularly, Tukey's test returns an upper diagonal square matrix, whose the column elements are sorted by their rank.

The post hoc analysis can eventually return a graphical output, which is shown in Fig. 8. In this case, $\alpha_{FT} = 0.05$. The final rank is built by considering the models ranks for decreasing values of CH scores by the Friedman test for each fault location. Hence, the number of samples per method is 39. In this case, the test returns that the null hypothesis is rejected, which allows for performing the post hoc analysis. Tukey's test supplies that the most performing clustering model is the M3C, which significantly differs from k-means. The latter is better than CC and FCM, but not enough to be considered significantly different. Hence, under the obtained results it is possible to assess the M3C outperforms the other clustering methods across the 39 fault locations.

5.5. Topological Compactness of clusters

Table 2 also contains the "Cluster Topological Compactness" (CTC) score, which measures the grade of topological compactness of the clusters. CTC can be calculated as follows:

$$\text{CTC}_{(s)} = \frac{B^c_{(s)}}{B_{(s)}} \tag{14}$$

where $B_{(s)}$ and $B^c_{(s)}$ are the total number of buses in the s-th cluster and the number of physically connected buses in that cluster, respectively. This score allows inspecting the degree of connection of the clustered groups. When all the clusters form connected graphs, this index is equal to 1. This index might be particularly important for some applications of coherency detection, such as

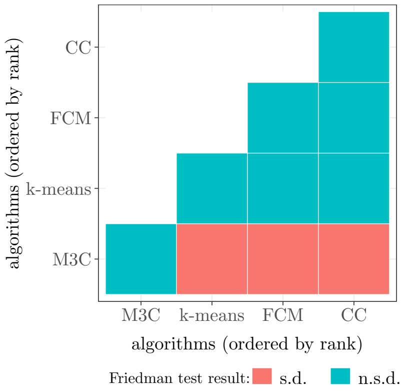


Figure 8: Visualization of Friedman's test with Post-hoc Analysis in terms of CH score. The best models are on the leftmost/lower side. The elements placed on the left side are placed in a specular manner to those on the downside for each insert. A green case means that the models on the respective row/column are not significantly different. (n.s.d), otherwise they are signicantly different (s.d).

controlled islanding, which requires connected groups. It is interesting to notice that M3C returns better results in terms of CTC, when compared to the benchmark method.

6. Conclusions

This paper proposes a Monte Carlo-based consensus clustering method as a tool to choose the optimal number of coherent groups in a network, based on the measured dynamic response of the system to a contingency. The proposed method is compared with the well-assessed clustering technique based on the maximum Average Silhouette score. The case study shows that the proposed method outperforms the benchmark methods both qualitatively, e.g., visual inspection, and quantitatively, considering the metrics. The Friedman test with post-hoc analysis statically confirmed the obtained results over the whole IEEE-39 bus network. Future works on the topic will focus on the application of the M3C method to power systems with a variable penetration of RES, which might affect the behavior of coherent groups due to their characteristic dynamics, and to effectively integrate the the algorithm in online-power system operation tools.

References

[1] E. Saadipour-Hanzaie, T. Amraee, S. Kamali, Minimal controlled islanding with similarity-based coherency identification using phasor measurement data, IEEE Trans. on Industrial Informatics (2021).

[2] J. L. Jardim, A. M. L. da Silva, A methodology for computing robust dynamic equivalents of large power systems, Electric Power Systems Research 143 (2017) 513–521.

[3] M. H. Rezaeian, S. Esmaeili, R. Fadaeinedjad, Generator coherency and network partitioning for dynamic equivalencing using subtractive clustering algorithm, IEEE Systems Journal 12 (4) (2017) 3085–3095.

[4] Z. Lin, F. Wen, Y. Ding, Y. Xue, S. Liu, Y. Zhao, S. Yi, Wams-based coherency detection for situational awareness in power systems with renewables, IEEE Trans. on Power Systems 33 (5) (2018) 5410–5426.

[5] J. H. Chow, Slow coherency and aggregation, in: Power System Coherency and Model Reduction, Springer, 2013, pp. 39–72.

[6] G. Rogers, Power system oscillations, Springer Science & Business Media, 2012.

[7] A. M. Khalil, R. Iravani, A dynamic coherency identification method based on frequency deviation signals, IEEE Trans. on Power Systems 31 (3) (2015) 1779–1787.

[8] F. R. S. Sevilla, Y. Liu, E. Barocio, P. Korba, M. Andrade, F. Bellizio, J. Bos, B. Chaudhuri, H. Chavez, J. Cremer, et al., State-of-the-art of data collection, analytics, and future needs of transmission utilities worldwide to account for the continuous growth of sensing data, International journal of electrical power & energy systems 137 (2022) 107772.

[9] M. H. R. Koochi, S. Esmaeili, G. Ledwich, Taxonomy of coherency detection and coherency-based methods for generators grouping and power system partitioning, IET Generation, Transmission & Distribution 13 (12) (2019) 2597–2610.

[10] S. Aghabozorgi, A. S. Shirkhorshidi, T. Y. Wah, Time-series clustering–a decade review, Information Systems 53 (2015) 16–38.

[11] Z. Lin, F. Wen, Y. Ding, Y. Xue, Data-driven coherency identification for generators based on spectral clustering, IEEE Trans. on Industrial Informatics 14 (3) (2017) 1275–1285.

[12] M. H. R. Koochi, S. Esmaeili, P. Dehghanian, Coherency detection and network partitioning supported by wide area measurement system, in: IEEE Texas Power and Energy Conference (TPEC), IEEE, 2018, pp. 1–6.

[13] M. Jonsson, M. Begovic, J. Daalder, A new method suitable for real-time generator coherency determination, IEEE Trans. on Power Systems 19 (3) (2004) 1473–1482.

[14] M. H. R. Koochi, S. Esmaeili, P. Dehghanian, Coherency detection and network partitioning supported by wide area measurement system, in: 2018 IEEE Texas power and energy conference (TPEC), IEEE, 2018, pp. 1–6.

[15] N. Senroy, Generator coherency using the hilbert–huang transform, IEEE Trans. on Power Systems 23 (4) (2008) 1701–1708.

[16] S. Avdaković, E. Bećirović, A. Nuhanović, M. Kušljugić, Generator coherency using the wavelet phase difference approach, IEEE Trans. on Power Systems 29 (1) (2013) 271–278.

[17] H. R. Chamorro, C. A. Ordonez, J. C. Peng, M. Ghandhari, Non-synchronous generation impact on power systems coherency, IET Generation, Transmission & Distribution 10 (10) (2016) 2443–2453.

[18] M. R. Monteiro, G. F. Alvarenga, Y. R. Rodrigues, A. Z. de Souza, B. Lopes, M. C. Passaro, M. Abdelaziz, Network partitioning in coherent areas of static voltage stability applied to security region enhancement, International Journal of Electrical Power & Energy Systems 117 (2020) 105623.

[19] K. K. Anaparthi, B. Chaudhuri, N. F. Thornhill, B. C. Pal, Coherency identification in power systems through principal component analysis, IEEE Trans. on Power Systems 20 (3) (2005) 1658–1660.

[20] M. Ariff, B. C. Pal, Coherency identification in interconnected power system—an independent component analysis approach, IEEE Trans. on Power Systems 28 (2) (2012) 1747–1755.

[21] Q. Zhu, J. Chen, X. Duan, X. Sun, Y. Li, D. Shi, A method for coherency identification based on singular value decomposition, in: IEEE PES General Meeting, IEEE, 2016, pp. 1–5.

[22] S. Liu, Z. Lin, Y. Zhao, Y. Liu, Y. Ding, B. Zhang, L. Yang, Q. Wang, S. E. White, Robust system separation strategy considering online wide-area coherency identification and uncertainties of renewable energy sources, IEEE Trans. on Power Systems 35 (5) (2020) 3574–3587.

[23] S.-K. Joo, C.-C. Liu, L. E. Jones, J.-W. Choe, Coherency and aggregation techniques incorporating rotor and voltage dynamics, IEEE Trans. on Power Systems 19 (2) (2004) 1068–1075.

[24] M. R. A. Paternina, A. Zamora-Mendez, J. Ortiz-Bejar, J. H. Chow, J. M. Ramirez, Identification of coherent trajectories by modal characteristics and hierarchical agglomerative clustering, Electric Power Systems Research 158 (2018) 170–183.

[25] E. Barocio, P. Korba, W. Sattinger, F. R. Segundo Sevilla, Online coherency identification and stability condition for large interconnected power systems using an unsupervised data mining technique, IET Generation, Transmission & Distribution 13 (15) (2019) 3323–3333.

[26] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Annals of Data Science 2 (2) (2015) 165–193.

[27] L. Lugnani, M. R. A. Paternina, D. Dotta, J. H. Chow, Y. Liu, Power system coherency detection from wide-area measurements by typicality-based data analysis, IEEE Trans. on Power Systems 37 (1) (2021) 388–401.

[28] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[29] G. Brock, V. Pihur, S. Datta, S. Datta, et al., clValid, an R package for cluster validation, Journal of Statistical Software (2011).

[30] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, Understanding of internal clustering validation measures, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 911–916.

[31] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data, Machine Learning 52 (1) (2003) 91–118.

[32] Y. Șenbabaoğlu, G. Michailidis, J. Z. Li, Critical limitations of consensus clustering in class discovery, Scientific Reports 4 (1) (2014) 1–13.

[33] C. R. John, D. Watson, D. Russ, K. Goldmann, M. Ehrenstein, C. Pitzalis, M. Lewis, M. Barnes, M3C: Monte Carlo reference-based consensus clustering, Scientific Reports 10 (1) (2020) 1–14.

[34] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2) (2001) 411–423.

[35] R. A. Fisher, Design of experiments, British Medical Journal 1 (3923) (1936) 554.

[36] B. Phipson, G. K. Smyth, Permutation p-values should never be zero: calculating exact p-values when permutations are randomly drawn, Statistical applications in genetics and molecular biology 9 (1) (2010).

[37] P. K. Kimes, Y. Liu, D. Neil Hayes, J. S. Marron, Statistical significance for hierarchical clustering, Biometrics 73 (3) (2017) 811–821.

[38] F. Milano, A Python-based software tool for power system analysis, in: IEEE PES General Meeting, IEEE, 2013, pp. 1–5.

[39] J. C. Bezdek, R. J. Hathaway, J. M. Huband, Visual assessment of clustering tendency for rectangular dissimilarity matrices, IEEE Trans. on Fuzzy Systems 15 (5) (2007) 890–903.

[40] I. Kamwa, A. K. Pradhan, G. Joos, S. Samantaray, Fuzzy partitioning of a real power system for dynamic vulnerability assessment, IEEE Trans. on Power Systems 24 (3) (2009) 1356–1365.

[41] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research 7 (2006) 1–30.

[42] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (200) (1937) 675–701.

[43] F. De Caro, J. De Stefani, G. Bontempi, A. Vaccaro, D. Villacci, Robust assessment of short-term wind power forecasting models on multiple time horizons, Technology and Economics of Smart Grids and Sustainable Energy 5 (1) (2020) 1–15.

[44] M. Rafiei, T. Niknam, J. Aghaei, M. Shafie-Khah, J. P. Catalão, Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine, IEEE Trans. on Smart Grid 9 (6) (2018) 6961–6971.

[45] R. Bisoi, D. R. Dash, P. Dash, L. Tripathy, An efficient robust optimized functional link broad learning system for solar irradiance prediction, Applied Energy 319 (2022) 119277.