Expert Systems with Applications 42 (2015) 6767–6776

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A robust unsupervised consensus control chart pattern recognition framework

Siavash Haghtalab [a], Petros Xanthopoulos [a,*], Kaveh Madani [b]

[a] Department of Industrial Engineering and Management Systems, University of Central Florida, 4000 Central Florida blvd., Orlando, FL 32816, USA

[b] Centre for Environmental Policy, Imperial College London, 14 Prince's Gardens, London SW7 1NA, UK

ARTICLE INFO

Article history:
Available online 4 May 2015

Keywords:
Consensus clustering
Unsupervised learning
Control chart pattern recognition
k-Means
Spectral clustering
Graph partitioning

ABSTRACT

Early identification and detection of abnormal patterns is vital for a number of applications. In manufacturing for example, slide shifts and alterations of patterns might be indicative of some production process anomaly, such as machinery malfunction. Usually due to the continuous flow of data, monitoring of manufacturing processes and other types of applications requires automated control chart pattern recognition (CCPR) algorithms. Most of the CCPR literature consists of supervised classification algorithms. Fewer studies consider unsupervised versions of the problem. Despite the profound advantage of unsupervised methodology for less manual data labeling their use is limited due to the fact that their performance is not robust enough and might vary significantly from one algorithm to another. In this paper, we propose the use of a consensus clustering framework that takes care of this shortcoming and produces results that are robust with respect to the chosen pool of algorithms. Computational results show that the proposed method achieves not less than 79.10% *G*-mean with most of test instances achieving higher than 90%. This happens even when in the algorithmic pool are included algorithms with performance less than 15%. To our knowledge, this is the first paper proposing an unsupervised consensus learning approach in CCPR. The proposed approach is promising and provides a new research direction in unsupervised CCPR literature.

1. Introduction

Time series analysis is an area of research with numerous application in many fields of science and engineering (Box, Jenkins, & Reinsel, 2013). In manufacturing, for instance, time series pattern recognition is important since slide alterations might be indicative of a malfunction that requires a course of appropriate corrective actions (e.g. maintenance). Manual monitoring is tedious and requires specialized personnel's undistracted attention. For this, machine learning based automated algorithms, also known as control chart pattern recognition (CCPR) algorithms, have been proposed to detect abnormal behaviors. The term was originally coined by Shewhart (1931). An early taxonomy of the patterns was presented in an early publication of the Western Electric Company (1958). Fig. 1 depicts six of the most common abnormal patterns studied in the literature.

These different abnormal patterns are usually related to a specific malfunction and their early detection can provide useful insights for corrective actions and thus improve systems reliability. In the crank case manufacturing operations, up trend and down trend patterns reveal tool wear and malfunction (El-Midany, El-Baz, & Abd-Elwahed, 2010a). Shift patterns might be associated with variation related to operator, material or machine instrument (Davy, Desobry, Gretton, & Doncarli, 2006; El-Midany et al., 2010a). Cyclic patterns are associated with voltage variability (Kawamura, Chuarayapratip, & Haneyoshi, 1988) but they can also appear in manufacturing processes like frozen orange juice packing (Hwarng, 1995). In the car manufacturing industry certain anomalies in the automotive body assembly process appear as up/down trends, cyclic, and systematic patterns (Jang, Yang, & Kang, 2003). Up/down trend patterns can be used in order to detect abnormal stamping tonnage signals (Jin & Shi, 2001). Finally up/down trend signals appear in paper making industry (Chinnam, 2002; Cook & Chiu, 1998) whereas uptrend patterns by itself can be used for detecting fault states in end-milling process (Zorriassatine, Al-Habaibeh, Parkin, Jackson, & Coy, 2005).

During several years, different pattern recognition algorithms have been studied in the literature with the proposed approaches ranging over a broad spectrum of machine learning algorithms. The majority of the proposed schemes follow the supervised

* Corresponding author.
E-mail addresses: s.haghtalab@gmail.com (S. Haghtalab), petrosx@ucf.edu (P. Xanthopoulos), k.madani@imperial.ac.uk (K. Madani).

http://dx.doi.org/10.1016/j.eswa.2015.04.069
0957-4174/

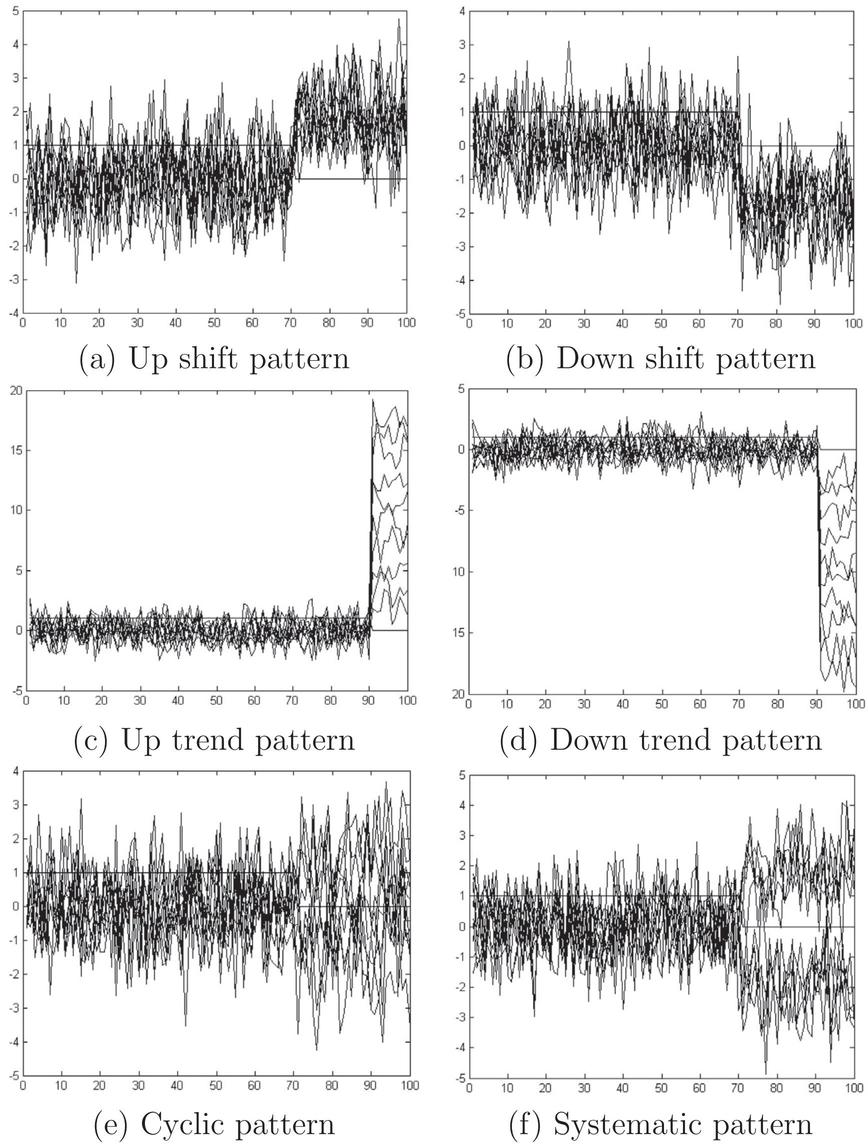


Fig. 1. Examples of six basic abnormal patterns.

learning framework, in which a model is trained with a historical dataset and then the trained model is used for prediction on an unknown testing data set. Some representative algorithms under this category include knowledge-based expert systems and artificial neural networks (El-Midany, El-Baz, & Abd-Elwahed, 2010b; Hwarng, 1995; Hwarng & Hubele, 1992, 1993a, 1993b; Guh & Hsieh, 1999; Kim, Jitpitaklert, Park, & Hwang, 2012; Perry, Spoerre, & Velasco, 2001; Wu & Yu, 2010; Yu & Xi, 2009), Bayes classification (Adam et al., 2011), and support vector Machines (SVM) (Camci, Chinnam, & Ellis, 2008). In more recent literature decomposition techniques are used as a preprocessing step before classification. Some examples include wavelets (Du, Huang, & Lv, 2013), independent component analysis (Cheng & Huang, 2013; Kao, Lee, & Lu, 2014) and extreme-point symmetric mode decomposition (Yang, Zhou, Liao, & Guo, 2015). In another recent study Wu, Liu, and Zhu (2014) proposed the combined approach of classification trees and SVM. For a comprehensive literature review we refer the reader to Hachicha and Ghorbel (2012) and Veiga, Mendes, and Lourenço (2015).

On the other hand, research on unsupervised CCPR algorithms is relatively limited. Unsupervised learning assumes no prior information and aims to categorize the data samples based only on their features (properties) (Warren Liao, 2005). The first unsupervised approach for CCPR was proposed by Al-Ghanim (1997) who developed an unsupervised self-organizing neural paradigm. Al-Ghanim and Kamat (1995) presented a CCPR technique using correlation analysis on trend, systematic and cyclic patterns and presented results with evaluation methods. Wang and Kuo (2007) used three different fuzzy clustering algorithms on CCPR for six patterns and compared their performance.

Unsupervised learning techniques have the profound advantage of not requiring prior labeling knowledge for prediction. On the other side, however, their behavior can be instable and sometimes inconsistent across algorithms or even across different runs of the same algorithm. In the clustering literature this shortcoming is normally addressed through *ensemble* or *consensus* learning schemes. Under this approach a number of clustering with different results is combined to a single clustering that is more robust according to some optimization criteria (Vega-Pons & Ruiz-Shulcloper, 2011; Xanthopoulos, 2014). However this idea has not been implemented yet for the CCPR problem. We anticipate that consensus framework will provide CCPR robust methodologies able to overcome the problems of individual clustering algorithms and thus open new directions for more research in unsupervised CCPR literature. In this paper we propose such a framework for unsupervised CCPR based on the concept of *consensus graph* and we study its behavior for a variety of CCPR problems. The rest of the paper is organized as follows. In Section 2 we provide a theoretical background of the *consensus clustering* problem. In Section 3 we describe the methodology used along with the evaluations metrics for assessment of the results. In Section 4 we present results and in Section 5 we conclude and discuss the future work.

2. Consensus clustering

Clustering algorithms might give different clustering of the same data by several times of running or using different algorithms. For instance, k-means clustering gives different results for the same data by choosing different initial solution. This phenomenon is well studied and is due to local optima convergence of k-means.

The general scheme for consensus clustering is depicted in Fig. 2. Consensus clustering consists of two steps. In the first steps we try to gather different clusterings (Al-Sultana & Khan, 1996). These clusterings can be the result of different data sources, different clustering algorithms, different runs of a nondeterministic clustering algorithm and other factors. In the next step we will use these clusterings to construct the consensus clustering by the help of consensus function (Selim & Ismail, 1984).

One of the earliest approaches for obtaining consensus clustering is solving the k-median problem for a set of clusterings, which can be formulated as an optimization problem (Grötschel & Wakabayashi, 1989). For a set of n different clusterings $\mathcal{P} = \{P_k\}_{k=1}^{n}$ we define $r_{ij}^{(k)} = 1$ if samples s_i and s_j belong to the same cluster in clustering P_k and 0 if they do not. Then we define the decision variable r_{ij} whose value is 1 if points s_i and s_j belong to the same cluster in the consensus clustering and 0 otherwise. The objective function can be the sum of distances between the consensus clustering and the clusterings in $\mathcal{P}$ with respect to some distance measure $d(\cdot,\cdot)$. For a Euclidean distance measure this can be written as:

$$\sum_i d(P, P_i) = \sum_i \sum_{i,j} \left(r_{ij}^{(k)} - r_{ij}\right)^2 \tag{1}$$

Since $r_{ij}^{(k)}$ and $r_{ij} \in \{0,1\}$, the function can be linearized:

$$\sum_k \sum_{i,j} r_{ij}^{(k)} + \sum_k \sum_{i,j} (1 - 2r_{ij}^{(k)}) r_{ij} \tag{2}$$

Since the first term is constant, the objective function can be written as:

$$c + \sum_{i,j} c_{ij} r_{ij} \tag{3}$$

where:

$$c = \sum_k \sum_{i,j} r_{ij}^{(k)}, \quad c_{ij} = \sum_k (1 - 2r_{ij}^{(k)}) \tag{4}$$

Finally, the optimization problem is formulated as follows:

$$\min \quad \sum_{ij} c_{ij} r_{ij} \tag{5a}$$

$$\text{s.t.} \quad r_{ii} = 1, \quad i = 1, \ldots, n \tag{5b}$$

$$r_{ij} = r_{ji}, \quad i,j = 1, \ldots, n \tag{5c}$$

$$r_{ij} + r_{jk} - r_{ik} \leqslant 1, \quad i,j,k = 1, \ldots, n \tag{5d}$$

$$r_{ij} \in \{0,1\}, \quad i,j = 1, \ldots, n \tag{5e}$$

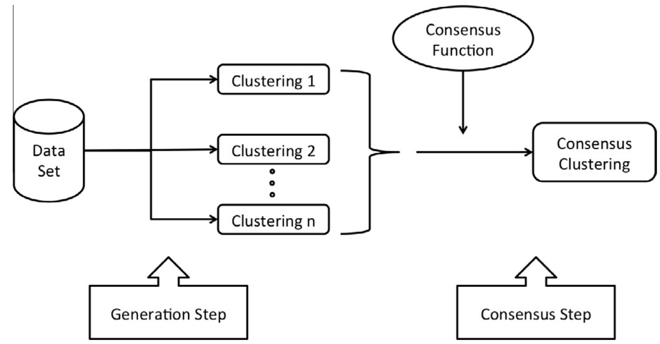


Fig. 2. Consensus clustering scheme.

Since $r_{ij} = r_{ji}$ these variables can be replaced by x_{ij}, and r_{ii} can be dropped since it is a fixed variable. By setting weights $w_{ij} = c_{ij} + c_{ji}$ the above optimization problem is equivalent to the following problem:

$$\min \quad \sum_{1 \leqslant i < j \leqslant n} w_{ij} x_{ij} \tag{6a}$$

$$\text{s.t.} \quad x_{ij} + x_{jk} - x_{ik} \leqslant 1, \quad 1 \leqslant i < j < k \leqslant n \tag{6b}$$

$$x_{ij} - x_{jk} + x_{ik} \leqslant 1, \quad 1 \leqslant i < j < k \leqslant n \tag{6c}$$

$$-x_{ij} + x_{jk} + x_{ik} \leqslant 1, \quad 1 \leqslant i < j < k \leqslant n \tag{6d}$$

$$x_{ij} \in \{0,1\}, \quad 1 \leqslant i < j \leqslant n \tag{6e}$$

The polyhedron of this problem is the same as the one in the clique partition problem (Grötschel & Wakabayashi, 1989). This problem is NP-complete and it is not possible to solve it in large instances (Sukegawa, Yamamoto, & Zhang, 2012). Several metaheuristics have been developed for obtaining "good" solutions in a "reasonable" time. Hansen, Ruiz, and Aloise (2012) proposed a metaheuristic method for normalized cut segmentation. Arya et al. (2004) analyzed local search heuristics for the metric k-median and facility location problems, Chen (2009) also presented a new heuristic method for k-median and k-mean clustering problems.

In addition, the solution of the k-median problem for consensus clustering has been criticized for not producing tight enough clusters. Lancichinetti and Fortunato (2012) mention that explicit optimization of global quality functions usually fail to identify clusters in practical settings. Based on their argument exact optimization approaches have serious limitations and the optimization will not result in good clusters.

Here, we employ a variant of the consensus clustering approach, proposed by Lancichinetti and Fortunato (2012), based on the concept of the consensus graph and meta-clustering approach (clustering the consensus clustering graph). This approach, applied to a citation co-authorship dataset, was found to produce more meaningful results in terms of content and cluster evaluation metrics.

The contribution in this paper is twofold: (1) First we employ the network-based consensus clustering approach as a framework for the quality control CCPR problem; and (2) we examine the algorithm from a robustness point of view, especially for scenarios when individual clustering algorithms are known not to be a "good choice" for the problem under consideration. Also, we study the CCPR problem as an imbalanced clustering problem meaning that the examples for normal patterns greatly outperform the ones for abnormal patterns (Xanthopoulos & Razzaghi, 2014).

2.1. Consensus graph

Under the presented framework, the consensus graph is a construct necessary for obtaining the consensus clustering. Each edge on the consensus graph has the weight equal to $a_{ij} = t_{ij}/n$ where t_{ij} is the number of times where samples i and j were assigned to the *same* cluster. A simple illustration of the consensus graph construction is given in Fig. 3.

For this toy example (Fig. 3) with seven data sample and four clusterings, the consensus graph adjacency matrix is given by:

$$A = \{a_{ij}\}_{i,j=1}^{7} = \begin{bmatrix} 1/4 & 2/4 & 4/4 & 1/4 & 0 & 0 & 0 \\ 2/4 & 1/4 & 2/4 & 3/4 & 1/4 & 0 & 0 \\ 4/4 & 2/4 & 1/4 & 1/4 & 0 & 0 & 0 \\ 1/4 & 3/4 & 1/4 & 1/4 & 1/4 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 1/4 & 3/4 & 3/4 \\ 0 & 0 & 0 & 0 & 3/4 & 1/4 & 4/4 \\ 0 & 0 & 0 & 0 & 3/4 & 4/4 & 1/4 \end{bmatrix}$$

In this graph, by construction, each edge has a weight between 0 and 1. The final solution can be obtained by applying a clustering algorithm to the network.

3. Methodology

3.1. Data description

We consider the CCPR problem from the scope of *time series clustering*. A time series is represented by $D = \{d_1, d_2, \ldots, d_n\}$ where $d_i \in \mathbb{R}^w$ with w being the number of features, also referred to as the *window length*. We are interested in assigning each d_i to a cluster that corresponds to *normal* and *abnormal* pattern. Data have been simulated using the most common models found in the literature (exact mathematical description can be found in Appendix A).

3.2. Clustering algorithms

For constructing our consensus matrix we will use five different clustering algorithms, including: Hierarchical, k-means, fuzzy and two spectral clustering algorithms. Each of these algorithms gives different clusters. Some of these algorithms give the same results after several runs but some of them such as k-means do not do so. Without loss of generality in the present paper we do not deal with multiple run variability of the same algorithm on the same data, but we run these five algorithms on the same data once. In a real scenario, one does not know if a specific algorithm is appropriate or not for a certain dataset. Since the scheme is unsupervised no labeling information is known a priori and this usually makes the choice of clustering algorithm problematic. Here, we examine the robustness of clustering quality in scenarios in which not all of clustering algorithms performs well individually.

Each clustering algorithm labels each d_i as 1 or 2 since we only need two clusters (normal or abnormal). After getting the results of different clusterings we construct the consensus matrix. Consensus matrix A is a $n \times n$ matrix, and each element has a value between 0 to 1. For each a_{ij}, the value shows how many times out of 5, d_i is in the same cluster as d_j.

For decision making and evaluation purposes through voting, we decide if two data are in the same cluster or not. If the value of a_{ij} is greater than 2, then d_i and d_j are in the same cluster; and if the value is smaller or equal than 2 then d_i and d_j are not in the same cluster. In the following sections we give a brief description of the clustering algorithms we used and then evaluate the results.

3.2.1. Clustering algorithms pool

The k-means is a partitioning method first proposed by MacQueen et al. (1967). Given a set of n unlabeled samples, a partitioning method creates k partitions of data, where each partition represents a cluster containing at least one object and $k \leqslant n$. In k-means, partitions are crisp meaning that each object belongs to exactly one cluster, and each cluster is represented by the mean value of the objects in that cluster. The goal of k-means is to minimize the objective function which is the total distance between all objects from their respective cluster centers. The k-means is an iterative algorithm that starts by choosing arbitrarily initial centers for the clusters and then assigning objects to the closest cluster centers and updating the clusters. This process continues until the value of the objective function is minimized.

C-means, the fuzzy variant of k-means, was first proposed by Dunn (1973) and improved by Bezdek (1981). In fuzzy clustering, partitions are not crisp, meaning that one object can belong to more than one cluster to a different degree. Fuzzy clustering associates each pattern to a cluster with a membership function. One can obtain crisp partitions through fuzzy c-means by assigning each sample to the cluster with the highest membership value.

Hierarchical clustering iteratively assigns data to different clusters by forming a tree structure. A hierarchical clustering results in a dendrogram representing the clusterings and similarity levels, and by breaking the dendrogram in different levels, one can obtain different clusterings. There are two types of hierarchical clustering: Agglomerative (bottom up) and Divisive (top down).

Hierarchical clustering has the problem of adjusting once a merging decision is made. Different measures to calculate the distance between clusters results in different variations of hierarchical clustering. Such measures include, but are not limited to, the single link, complete link and minimum variance (Murtagh, 1983).

Different distance measures and linkage criteria can be used for hierarchical clustering. Regarding our data, we used agglomerative because this is the most popular one. We also used euclidean distance measure and single linkage method, because we want to prevent the effects of outliers and usually single link algorithms are more versatile than complete links (Jain, Murty, & Flynn, 1999).

Spectral clustering techniques use the eigenvalues of the Laplacian matrix. Spectral clustering goes back to Donath and Hoffman (1973) when they proposed the use of eigenvectors for graph partitions. In this paper, we used two spectral clustering algorithms, one proposed by Shi and Malik (2000) for solving the perceptual grouping problem in computer vision, and one proposed by Ng, Jordan, and Weiss (2002).

The selected pool demonstrates a diverse algorithmic characteristic and it is known that they each one of them is preferred depending on the structure of the dataset. For example k-means is preferred when clusters form normally distributed clouds, whereas spectral clustering tends to form more balanced clusters. Hierarchical clustering is a greedy iterative approach that has been a popular choice for many applications including medical literature. Although it is known and anticipated that each algorithm by itself might not be the optimal choice for each test problem under consideration it is of interest to study the consensus behavior and its robust characteristic for different clustering problems.

3.3. Evaluation methodology

Different evaluation methods can be used for imbalanced data. The evaluation method we use is based on the confusion matrix. In this data set, data is either in a positive class or negative class, and the algorithms decides on that. So, all in all there are four types of data. If data is in t positive class and algorithm identifies them as positive, then it is true positive. But if the algorithm places them in the negative class then it is false negative and the same thing applies to the negative class data. Based on this rule, we calculate the sensitivity and specificity:

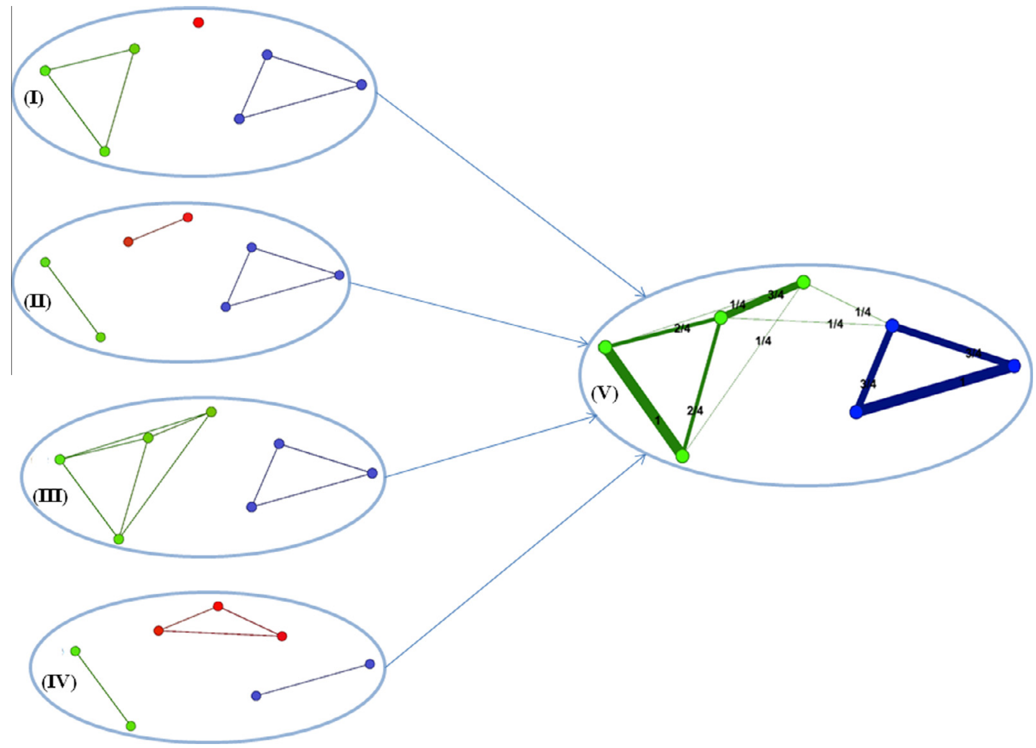


Fig. 3. Consensus clustering graph illustration, (*I*), (*II*), (*III*), (*IV*) are four different clusterings and (*V*) is the consensus created based on them.

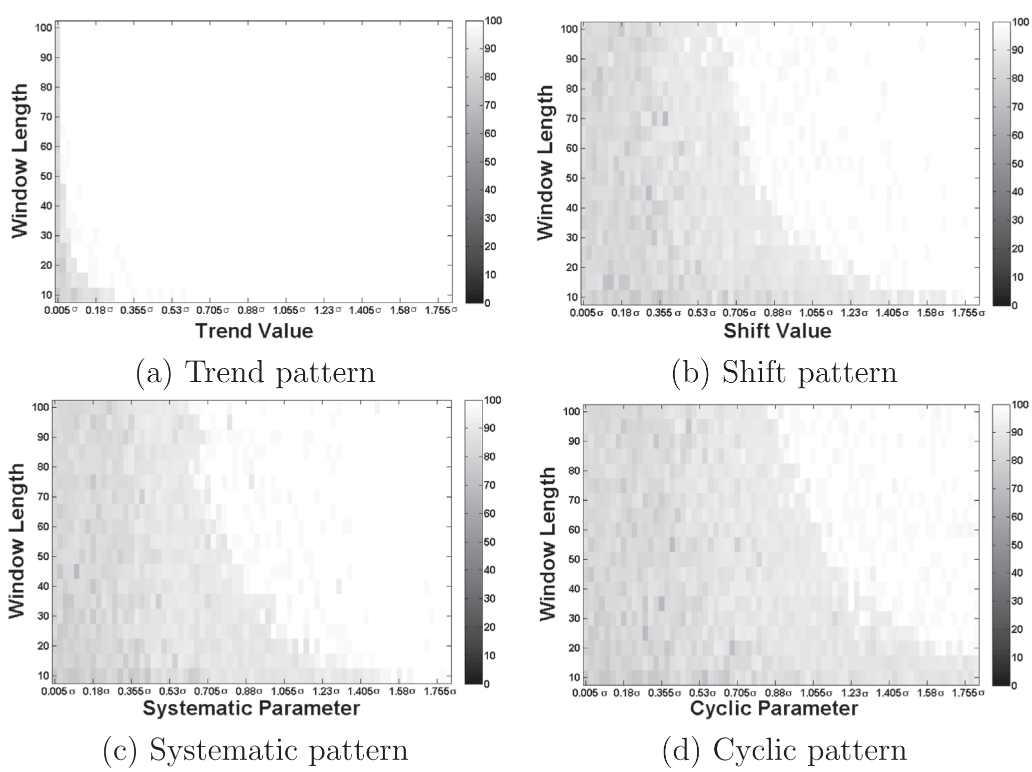


Fig. 4. Consensus clustering results based on *G*-mean calculations.

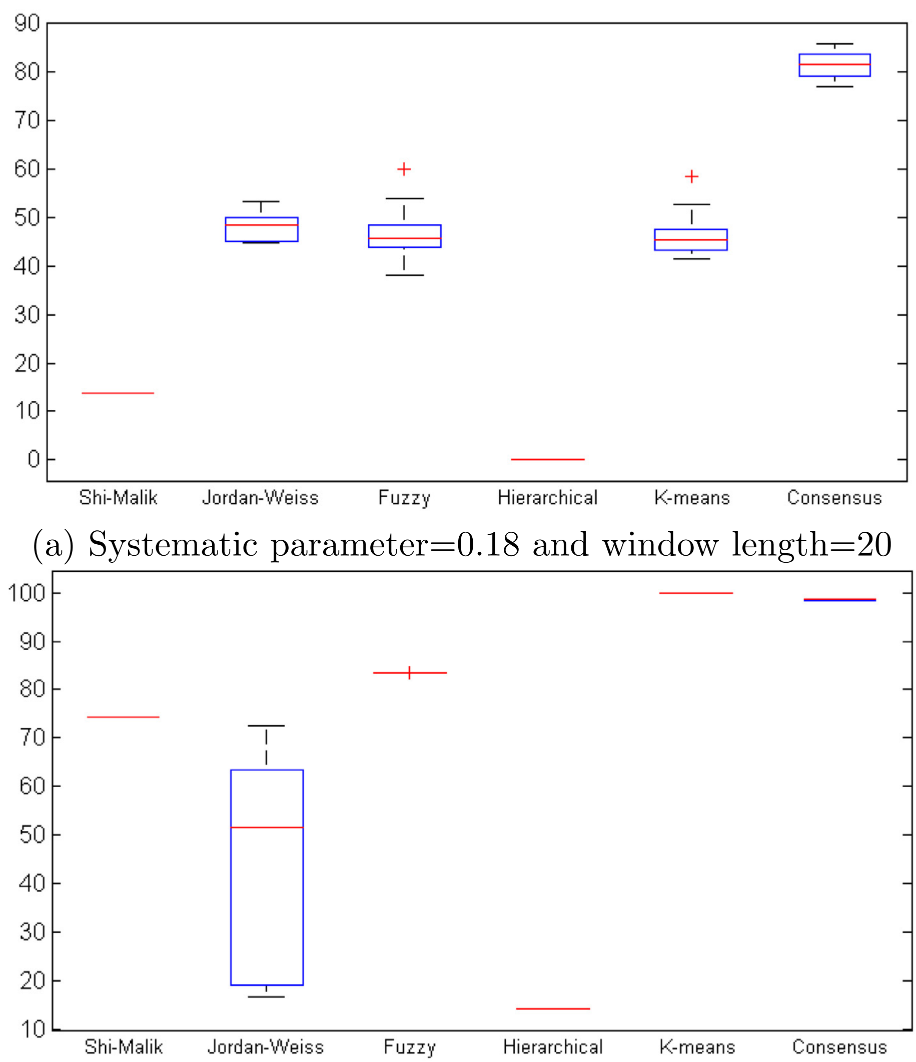
$$Sensitivity = \frac{TP}{TP + FP} \quad Specificity = \frac{TN}{TN + FN} \tag{7}$$

where TP, FN, FP, TN stands for True Positive, False Negative, False Positive and True Negative, respectively. Sensitivity shows how well the algorithm performs on the positive class and specificity shows how well the algorithm performs on the negative class. The geometric mean (*G*-mean) of sensitivity and specificity is also defined as

$$G - mean = \sqrt{Specificity \times Sensitivity} \tag{8}$$

Table 1
Comparison between the methods based on *G*-mean for different patterns and parameters. The detailed breakdown in sensitivity and specificity can be found in Appendix A (Table A.3).

Abnormal Pattern	Shi–Malik	Jordan–Weiss	Fuzzy	Hierarchical	*k*-Means	2-Cons.	3-Cons.	4-Cons.	5-Cons.	Parameters
Uptrend	19.66	54.85	100.00	100.00	100.00	100.00	100.00	100.00	100.00	(60,0.205)
	23.78	44.11	35.40	14.14	69.44	86.96	64.48	90.28	90.88	(85,0.006)
	23.20	53.11	54.77	14.14	54.13	89.07	59.91	84.57	79.10	(60,0.004)
Downtrend	13.71	31.62	100.00	100.00	100.00	100.00	100.00	100.00	100.00	(60,0.205)
	19.56	46.47	45.00	14.14	38.77	94.61	54.44	81.79	86.36	(85,0.006)
	34.72	45.56	56.26	0.00	53.78	88.37	50.44	87.26	82.99	(60,0.004)
Upshift	0.00	42.65	77.26	14.14	99.94	100.00	100.00	100.00	98.90	(60,0.805)
	23.96	39.19	69.10	0.00	33.61	92.32	75.10	99.18	87.07	(80,0.405)
	37.39	46.16	55.62	0.00	58.86	87.52	50.61	90.04	83.14	(40,0.205)
Downshift	50.01	57.45	78.41	14.14	99.84	100.00	100.00	100.00	99.22	(60,0.805)
	19.62	48.11	62.71	14.14	52.78	89.95	56.16	100.00	90.73	(80,0.405)
	28.95	45.50	54.56	0.00	39.61	89.52	49.14	77.95	85.75	(40,0.205)
Cyclic	23.74	58.06	0.00	14.14	99.57	100.00	100.00	100.00	98.08	(50,1.205)
	27.33	62.03	9.59	0.00	75.25	96.93	100.00	99.91	89.70	(50,0.805)
	25.74	44.80	23.15	0.00	44.40	88.23	54.85	84.65	81.24	(40,0.405)
Systematic	24.92	48.52	0.00	14.14	99.73	93.95	100.00	100.00	98.86	(60,0.805)
	30.44	59.40	0.00	0.00	78.13	97.92	100.00	94.20	91.12	(50,0.605)
	40.47	49.01	48.37	10.00	53.95	87.57	47.91	81.71	80.90	(20,0.205)



(b) Systematic parameter=1.405 and window length=50

Fig. 5. Robustness of consensus clustering in comparison to other clustering algorithms for systematic pattern. (a) is the worst state for systematic parameter and (b) is a good state.

Table 2
Running times for different datasets. The first column shows the total running time (individual algorithms plus consensus) and each other column running time that correspond to a specific algorithm.

Data Size	Total time	Cons.	Fuzzy	Hierarchical	Jordan	*k*-Mean	Shi–Malik
100	0.86	0.02	0.03	0.20	0.26	0.05	0.30
300	0.43	0.18	0.01	0.02	0.13	0.00	0.08
500	1.48	0.47	0.01	0.03	0.55	0.00	0.40
700	3.79	0.89	0.01	0.07	1.62	0.01	1.19
900	7.84	1.51	0.02	0.11	3.55	0.01	2.64
1000	10.43	1.77	0.02	0.15	4.92	0.01	3.58
1500	38.91	4.43	0.04	0.52	20.81	0.02	13.10
2000	78.73	7.22	0.04	0.93	41.25	0.02	29.26
3000	258.21	17.09	0.06	2.69	139.33	0.03	99.01
4000	599.49	29.34	0.09	6.51	329.52	0.06	233.97
5000	1233.42	45.81	0.10	11.43	685.83	0.06	490.18
6000	1993.19	66.19	0.13	19.82	1111.99	0.07	794.99
7000	3428.73	115.55	0.41	46.37	1951.26	0.37	1314.78
8000	4829.83	122.89	0.25	50.98	2744.49	0.09	1911.13
9000	7233.65	161.93	0.35	75.33	4086.10	0.23	2909.71
10,000	9084.15	178.99	0.25	90.18	5171.32	0.18	3643.23

which is a composite measure that takes into account both sensitivity and specificity (Kubat, Holte, & Matwin, 1997, 1998).

4. Results

In this section, the results of the given algorithm for imbalanced data is given based on the evaluation methods defined in Section 3.3 and data defined in Section 3.1. This algorithm is implemented in MATLAB Version 7.8 (R2009a). In Fig. 4 the *G*-mean results for consensus matrix of for different patterns are given. For each pattern, *G*-mean is calculated for different window lengths and pattern parameters. For each test problem we created 1000 data samples out of which 95% are normal and only 5% abnormal in order to account for the imbalanced nature of the problem in real settings (Xanthopoulos & Razzaghi, 2014). For clustering the consensus clustering matrix *A* we employed a majority voting scheme that is simple and keeps the last step of the computational effort low (Saeed, Salim, & Abdo, 2012).

As the charts in Fig. 4 show, the clustering problem becomes less challenging as we increase the window length and the abnormal pattern parameter. For each pattern, we can distinguish between three problem categories based on the parameter-window length combination and the corresponding *G*-mean performance: (1) easy problems (white area), (2) medium problems (gray area); and (3) difficult problems (dark area). This is consistent with previous observations (Xanthopoulos & Razzaghi, 2014). For example, as Fig. 4 shows, a window length 15 and trend parameter 0.005σ is a challenging problem. However, for trend pattern, a window length 90 and trend parameter 1.055 is an easy problem. The problem difficulty varies for different patterns, For instance, while (80,1.055) is an easy problem for trend pattern, it is a difficult one for cyclic pattern. In Table 1, we compare consensus clustering results with other algorithms in three different problems of each pattern. In this table, we only included *G*-mean results, specificity and sensitivity are included in the table in appendix. In these points there is a major change in the consensus clustering result and they can be good representatives for comparing the results of the consensus with each of the algorithms. As shown in the table, we have compared the results of the consensus clusterings that are constructed based on two, three, four and the original five algorithms. As the numbers are showing, consensus clustering that is created by five algorithms is producing more stable and reliable results, sometimes other consensuses are giving better *G*-means but those are not reliable and are dependent on the single algorithms, as we increase the number of algorithms, consensus becomes more stable, less relying on single algorithms and produces better results but at the cost of higher run time.

The robustness of the clustering algorithm plays an important role in some problems, a desirable clustering algorithm should cluster data the same way after several times of running. For measuring this characteristic of the consensus and comparing it with other algorithms, we used box plots as shown in Fig. 5. Using box plots gives information about the degree of dispersion and variability of the clustering results. For our problem, we run each clustering algorithm 30 times on the same data and at the end we calculate the *g*-means for each run of the algorithms. Each box plot shows the *G*-means of each of the algorithms for 30 times of run. As the figure is showing, Hierarchical clustering and Shi–Malik algorithms are giving the same result each time, but consensus is more robust than the other algorithms for both cases.

Another important factor in evaluating algorithm performance is their run time. In Table 2 the run time of the algorithm in seconds is given for different data sizes. The run time is calculated on a laptop with Intel core i7 CPU and 8 GB of RAM. This is the running time for systematic error patterns, which was the highest among all patterns. As it is shown in the figure, the highest run time belongs to spectral clustering algorithms and those are the reasons behind exponential increase in run time. In a practical setup, the run time can be decreased significantly through parallelization. We can see that the required time for spectral clustering computational time is disproportionally computationally intensive compared to the other algorithms. This is expected since it requires computation of the eigenvalues and eigenvectors of the Laplacian matrix which is a computationally intensive procedure and highly depends on the size of the dataset.

5. Discussion and conclusion

In this paper, we proposed an unsupervised consensus framework for solving the CCPR problem. To the best of our knowledge this is the first time that the CCPR problem has been formulated as a network clustering problem.

The robust solution seems to be consistently robust with respect to individual algorithms and is not affected much by individual poor clusterings. The use of consensus helps us find the similarities and eliminate weak clusters. This is very useful when dealing with high dimensional datasets where the optimal choice of clustering algorithm is not a priori known. Thus through this proposed framework we wish to motivate more research for unsupervised schemes with robust results. Computationally, implementing consensus clustering can be significantly faster through

Table A.3
Comparison between the methods based on sensitivity and specificity for different patterns and parameters.

Abnormal pattern	Shi–Malik		Jordan–Weiss		Fuzzy		Hierarchical		*k*-Means		2-Cons.		3-Cons.		4-Cons.		5-Cons.		Parameters
	Sen	Spec	Sen	Spec	Sen	Spec	Sen	Spec	Sen	Spec	Sen	Spec	Sen	Spec	Sen	Spec	Sen	Spec	(w, Par)
Uptrend	96.63	4.00	48.53	62.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	(60,0.205)
	94.21	6.00	46.32	42.00	48.21	26.00	100.00	2.00	52.42	92.00	87.16	87.36	56.89	50.19	87.59	84.98	86.10	95.91	(85,0.006)
	89.68	6.00	48.63	58.00	50.00	60.00	100.00	2.00	50.52	58.00	88.23	87.39	54.74	50.02	83.31	82.06	80.02	78.20	(60,0.004)
Downtrend	94.00	2.00	50.00	20.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	(60,0.205)
	95.68	4.00	46.95	46.00	48.21	42.00	100.00	2.00	44.21	34.00	91.05	87.62	52.51	50.64	80.52	79.27	85.15	87.59	(85,0.006)
	86.11	14.00	51.89	40.00	51.05	62.00	99.89	0.00	53.57	54.00	87.90	87.43	50.27	50.10	86.40	85.54	84.13	81.87	(60,0.004)
Upshift	92.63	0.00	50.53	36.00	59.68	100.00	100.00	2.00	99.89	100.00	93.68	87.76	87.01	75.72	99.16	98.33	97.81	100.00	(60,0.805)
	95.68	6.00	48.00	32.00	53.05	90.00	99.89	0.00	51.36	22.00	89.82	87.38	61.31	50.06	91.51	84.44	83.74	90.53	(80,0.405)
	93.22	15.00	49.56	43.00	51.56	60.00	99.89	0.00	55.89	62.00	87.49	87.46	50.26	49.91	87.42	84.87	80.87	85.46	(40,0.205)
Downshift	96.21	26.00	51.58	64.00	61.47	100.00	100.00	2.00	99.68	100.00	93.92	88.21	87.11	75.89	98.73	97.47	99.44	100.00	(60,0.805)
	96.21	4.00	50.32	46.00	50.42	78.00	100.00	2.00	44.94	62.00	88.69	87.45	53.23	50.46	92.25	85.10	84.90	96.97	(80,0.405)
	10.22	82.00	45.00	46.00	51.33	58.00	99.89	0.00	52.31	30.00	88.54	87.57	49.55	49.97	79.66	81.41	82.88	88.73	(40,0.205)
Cyclic	93.89	6.00	49.58	68.00	39.79	0.00	100.00	2.00	99.15	100.00	93.90	88.17	87.08	75.83	98.37	96.77	96.83	99.34	(50,1.205)
	93.37	8.00	52.00	74.00	46.00	2.00	99.89	0.00	57.78	98.00	91.61	86.59	73.06	53.37	89.73	80.59	82.75	97.22	(50,0.805)
	94.67	7.00	46.67	43.00	44.67	12.00	99.89	0.00	51.89	38.00	87.69	87.14	52.36	49.99	83.14	81.66	83.17	79.34	(40,0.405)
Systematic	6.21	100.00	49.05	48.00	37.05	0.00	100.00	2.00	99.47	100.00	90.67	87.51	86.86	75.45	88.99	79.19	97.74	100.00	(60,0.805)
	92.63	10.00	51.89	68.00	44.42	0.00	99.89	0.00	61.05	100.00	92.56	87.50	72.11	52.00	88.81	83.73	83.04	100.00	(50,0.605)
	86.22	19.00	52.22	46.00	49.78	47.00	100.00	1.00	48.52	60.00	87.50	87.42	48.97	50.04	82.42	83.13	81.57	80.24	(20,0.205)

parallelization and thus to become more appealing for real time applications.

Here is must be noted that the two or multiple class approaches such as the one proposed in this manuscript or others explored in CCPR literature can be used only when abnormal data are available. Although this might be the case in the majority of production control problems still there might be exceptions especially in applications where abnormal data are indicative of a rare but catastrophic event (e.g. temperature elevation in a nuclear plant). When this is the case the abnormal pattern detection problem can be studied under the framework of one class classification and/or outlier detection (Brun, Saggese, & Vento, 2014; Gupta, Gao, Aggarwal, & Han, 2014; Kind, Stoecklin, & Dimitropoulos, 2009). Overall, in applications where one wishes to detect outlier points in addition to the anticipated abnormal patterns it is possible to use consensus clustering along with a outlier detection algorithmic scheme.

Future research directions include development of a weighted consensus scheme able to adjust each clusterings contribution to consensus as well as the generalization to the multi class CCPR problem. In addition, applicability to real datasets must be explored. At the moment, as pointed out in the review paper of Hachicha and Ghorbel (2012) 95.59% of the literature about CCPR are based on simulated data evaluations.

Appendix A. Data model description

The simulation method we used for generating data uses different formula for different patterns. An implementation of this data generator can be downloaded from the MATLAB file exchange.[1] These formulas are used in the literature for different CCPR problems (Guh & Hsieh, 1999; Yang & Yang, 2005). For a normal pattern, the pattern has a normal distribution, so let $A(t)$ be a vector such as $A^T = [a_1, a_2, \ldots, a_t]$. Then $A(t)$ is a normal pattern if:

$$A(t) = n(t) \tag{A.1}$$

where, $n(t)$ follows a normal distribution $N(0,1)$.Upshift and downshift patterns are defined as:

$$A(t) = n(t) + ud \tag{A.2}$$

where, $u = 1$ after shift and $u = 0$ before shift. d is the shift parameter which will be chosen later for different datasets. Uptrend and downtrend patterns will be defined as:

$$A(t) = n(t) \pm dt \tag{A.3}$$

where d is the trend slope, which will be chosen later for different data sets. Cyclic pattern is as below:

$$A(t) = n(t) + d \sin\left(\frac{2\pi t}{\omega}\right) \tag{A.4}$$

where ω is the cycle length and d is the cyclic parameter. Finally, systematic patterns will be defined as:

$$A(t) = n(t) + (-1)^t d \tag{A.5}$$

References

Adam, A., Chew, L. C., Shapiai, M. I., Jau, L. W., Ibrahim, Z., & Khalid, M. (2011). A hybrid artificial neural network-Naive Bayes for solving imbalanced dataset problems in semiconductor manufacturing test process. In *Proceedings of the 11th international conference on hybrid intelligent systems (HIS)* (pp. 133–138). IEEE.

Al-Ghanim, A. (1997). An unsupervised learning neural algorithm for identifying process behavior on control charts and a comparison with supervised learning approaches. *Computers & Industrial Engineering, 32*(3), 627–639.

Al-Ghanim, A. M., & Kamat, S. J. (1995). Unnatural pattern recognition on control charts using correlation analysis techniques. *Computers & Industrial Engineering, 29*(1), 43–47.

Al-Sultana, K. S., & Khan, M. M. (1996). Computational experience on four algorithms for the hard clustering problem. *Pattern Recognition Letters, 17*(3), 295–308.

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., & Pandit, V. (2004). Local search heuristics for *k*-median and facility location problems. *SIAM Journal on Computing, 33*(3), 544–562.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers.

Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2013). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Brun, L., Saggese, A., & Vento, M. (2014). Dynamic scene understanding for behavior analysis based on string kernels. *IEEE Transactions on Circuits and Systems for Video Technology, 24*(10), 1669–1681.

Camci, F., Chinnam, R., & Ellis, R. (2008). Robust kernel distance multivariate control chart using support vector principles. *International Journal of Production Research, 46*(18), 5075–5095.

Chen, K. (2009). On coresets for *k*-median and *k*-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing, 39*(3), 923–947.

Cheng, C.-S., & Huang, K.-K. (2013). Identifying process status changes via integration of independent component analysis and support vector machine. In *Proceedings of the institute of industrial engineers Asian conference 2013* (pp. 1009–1016). Springer.

Chinnam, R. B. (2002). Support vector machines for recognizing shifts in correlated and other manufacturing processes. *International Journal of Production Research, 40*(17), 4449–4466.

Cook, D. F., & Chiu, C.-C. (1998). Using radial basis function neural networks to recognize shifts in correlated manufacturing process parameters. *IIE Transactions, 30*(3), 227–234.

Davy, M., Desobry, F., Gretton, A., & Doncarli, C. (2006). An online support vector machine for abnormal events detection. *Signal Processing, 86*(8), 2009–2025.

Donath, W. E., & Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development, 17*(5), 420–425.

Du, S., Huang, D., & Lv, J. (2013). Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines. *Computers & Industrial Engineering, 66*(4), 683–695.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics, 3*, 32–57.

El-Midany, T., El-Baz, M., & Abd-Elwahed, M. (2010a). A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks. *Expert Systems with Applications, 37*(2), 1035–1042.

El-Midany, T. T., El-Baz, M., & Abd-Elwahed, M. (2010b). A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks. *Expert Systems with Applications, 37*(2), 1035–1042.

Grötschel, M., & Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Mathematical Programming, 45*(1-3), 59–96.

Guh, R.-S., & Hsieh, Y.-C. (1999). A neural network based model for abnormal pattern recognition of control charts. *Computers & Industrial Engineering, 36*(1), 97–108.

Gupta, M., Gao, J., Aggarwal, C., & Han, J. (2014). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering, 26*(9), 2250–2267.

Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. *Computers & Industrial Engineering, 63*(1), 204–222.

Hansen, P., Ruiz, M., & Aloise, D. (2012). A vns heuristic for escaping local extrema entrapment in normalized cut clustering. *Pattern Recognition, 45*(12), 4337–4345.

Hwarng, H. (1995). Multilayer perceptions for detecting cyclic data on control charts. *International Journal of Production Research, 33*(11), 3101–3117.

Hwarng, H. B., & Hubele, N. F. (1992). Boltzmann machines that learn to recognize patterns on control charts. *Statistics and Computing, 2*(4), 191–202.

Hwarng, H. B., & Hubele, N. F. (1993a). Back-propagation pattern recognizers for x control charts: Methodology and performance. *Computers & Industrial Engineering, 24*(2), 219–235.

Hwarng, H. B., & Hubele, N. F. (1993b). X control chart pattern identification through efficient off-line neural network training. *IIE Transactions, 25*(3), 27–40.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR), 31*(3), 264–323.

Jang, K.-Y., Yang, K., & Kang, C. (2003). Application of artificial neural network to identify non-random variation patterns on the run chart in automotive assembly process. *International Journal of Production Research, 41*(6), 1239–1254.

Jin, J., & Shi, J. (2001). Automatic feature extraction of waveform signals for in-process diagnostic performance improvement. *Journal of Intelligent Manufacturing, 12*(3), 257–268.

Kao, L.-J., Lee, T.-S., & Lu, C.-J. (2014). A multi-stage control chart pattern recognition scheme based on independent component analysis and support vector machine. *Journal of Intelligent Manufacturing*, 1–12.

Kawamura, A., Chuarayapratip, R., & Haneyoshi, T. (1988). Deadbeat control of pwm inverter with modified pulse patterns for uninterruptible power supply. *IEEE Transactions Industrial Electronics, 35*(2), 295–300.

[1] http://www.mathworks.com/matlabcentral/fileexchange/45924-weighted-support-vector-machines.

Kim, S. B., Jitpitaklert, W., Park, S.-K., & Hwang, S.-J. (2012). Data mining model-based control charts for multivariate and autocorrelated processes. *Expert Systems With Applications, 39*(2), 2073–2081.
Kind, A., Stoecklin, M. P., & Dimitropoulos, X. (2009). Histogram-based traffic anomaly detection. *IEEE Transactions on Network and Service Management, 6*(2), 110–121.
Kubat, M., Holte, R., & Matwin, S. (1997). Learning when negative examples abound. *Machine Learning: ECML-97*, 146–153.
Kubat, M., Holte, R., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning, 30*(2), 195–215.
Lancichinetti, A., & Fortunato, S. (2012). Consensus clustering in complex networks. *Scientific Reports, 2*.
MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, California, USA* (Vol. 1, p. 14).
Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal, 26*(4), 354–359.
Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems, 2*, 849–856.
Perry, M. B., Spoerre, J. K., & Velasco, T. (2001). Control chart pattern recognition using back propagation artificial neural networks. *International Journal of Production Research, 39*(15), 3399–3418.
Saeed, F., Salim, N., & Abdo, A. (2012). Voting-based consensus clustering for combining multiple clusterings of chemical structures. *J. Cheminformatics, 4*(1), 1–8.
Selim, S. Z., & Ismail, M. A. (1984). *K*-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1), 81–87.
Shewhart, W. A. (1931). Economic control of quality of manufactured product, New York 501.
Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 888–905.
Sukegawa, N., Yamamoto, Y., & Zhang, L. (2012). Lagrangian relaxation and pegging test for the clique partitioning problem. *Advances in Data Analysis and Classification*, 1–29.
Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence, 25*(03), 337–372.
Veiga, P., Mendes, L., & Lourenço, L. (2015). A retrospective view of statistical quality control research and identification of emerging trends: A bibliometric analysis. *Quality & Quantity*, 1–20.
Wang, C.-H., & Kuo, W. (2007). Identification of control chart patterns using wavelet filtering and robust fuzzy clustering. *Journal of Intelligent Manufacturing, 18*(3), 343–350.
Warren Liao, T. (2005). Clustering of time series data survey. *Pattern Recognition, 38*(11), 1857–1874.
Western Electric Company (1958). *Statistical quality control handbook*. Indianapolis, Indiana: Western Electric Co. Inc.
Wu, C., Liu, F., & Zhu, B. (2014). Control chart pattern recognition using an integrated model based on binary-tree support vector machine. *International Journal of Production Research, 53*(7), 2026–2040.
Wu, B., & Yu, J.-b. (2010). A neural network ensemble model for on-line monitoring of process mean and variance shifts in correlated processes. *Expert Systems with Applications, 37*(6), 4058–4065.
Xanthopoulos, P. (2014). A review on consensus clustering methods. In *Optimization in science and engineering* (pp. 553–566). Springer.
Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. *Computers & Industrial Engineering, 70*, 134–149.
Yang, J.-H., & Yang, M.-S. (2005). A control chart pattern recognition system using a statistical correlation coefficient method. *Computers & Industrial Engineering, 48*(2), 205–221.
Yang, W.-A., Zhou, W., Liao, W., & Guo, Y. (2015). Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines. *Neurocomputing, 147*, 260–270.
Yu, J.-b., & Xi, L.-f. (2009). A neural network ensemble-based model for on-line monitoring and diagnosis of out-of-control signals in multivariate manufacturing processes. *Expert Systems with Applications, 36*(1), 909–921.
Zorriassatine, F., Al-Habaibeh, A., Parkin, R., Jackson, M., & Coy, J. (2005). Novelty detection for practical pattern recognition in condition monitoring of multivariate processes: A case study. *The International Journal of Advanced Manufacturing Technology, 25*(9), 954–963.