Combined Generative Topographic Mapping and Graph Theory Unsupervised Approach for Nonlinear Fault Identification

Matheus S. Escobar, Hiromasa Kaneko, and Kimito Funatsu
Chemical System Engineering Dept., The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan

DOI 10.1002/aic.14748
Published online February 12, 2015 in Wiley Online Library (wileyonlinelibrary.com)

Identifying anomalies in chemical processes is highly desirable. Usually, one relies on previous knowledge of normal and faulty samples, excluding anomalies from model training and associating deviations to faults. How reliable is such knowledge, however, is questionable, especially during atypical scenarios. Unsupervised approaches, using no labels, provide an unbiased analysis. A generative topographic mapping (GTM) and graph theory combined approach, then, is proposed for unsupervised fault identification. GTM, given its probabilistic nature, highlights system features, reducing variable dimensionality. With this information, correlation between samples is calculated. Graph theory, then, generates a network, clustering similar samples. Two anomaly cases are analyzed: an artificial dataset and Tennessee Eastman Process. Principal component analysis (PCA) and Dynamic PCA indexes Q and T^2 along GTM and graph theory-independent monitoring methodologies are used for comparison, considering supervised and unsupervised approaches. The proposed method performed similarly to all supervised methodologies, motivating its application and developments.
 AIChE J, 61: 1559–1571, 2015
Keywords: generative topographic mapping, graph theory, fault identification, process monitoring, nonlinear systems

Introduction

In the realm of chemical processes, machine learning techniques are widely used with, generally, one idea in mind: to highlight the most important features, characteristics, and relationships between variables in a given system. This general notion can be applied for different goals, such as fault identification, which deals with the recognition of anomalies from a so-called normal process state. These anomalies can represent hidden plant states, disturbances, controller malfunction, among other things, leading to different applications. Soft sensors,[1] for instance, are directly affected by it, since model database can be monitored to exclude anomalous samples. Process control applications are also interesting, where identifying faulty data is useful in alarm technologies and hierarchical control systems, ensuring a faster response to anomalous scenarios.

In this work, we focus on Multivariate Statistical Process Control and Monitoring[2,3] aspects. By evaluating how variables interact with each other and how this interaction influences the system, a complete understanding of one's process is possible. The most common methodology considers a previously known stable state, where any deviations from it are flagged as anomalous.[4,5] Such supervised approach associates to each sample a label: "normal" or "anomaly." In many cases, nonetheless, such labels might be inexistent, unreliable, or just not openly available. From this premise, unsupervised approaches for fault identification are interesting, where no information on normal or anomalous samples is given and only the relationship between variables and their evolution over time is relevant for data discrimination.

Not relying on data labels also provides analyses free of bias, free of vices, which can be helpful given the proper scenario. It might be odd to imagine chemical plants dealing with completely unsupervised approaches, considering that operators, for example, possess intricate knowledge about the process, acting as indicators of normal or faulty behavior. To what extent, however, is such knowledge reliable? Operators still suffer from human error, despite all their efforts. Furthermore, how subtle can different anomalies be identified? Can the human eye capture small, but anomalous, changes in the process before the system is far from its normal operational region? Considering slow data drift, imagine unexpected pipes clogging as time goes by. Operators may lack the fine-tuning required for detecting such gradual transition. While we acknowledge that such knowledge is relevant to a certain extent, a fully unsupervised approach is unbiased, relying only on how variables relate with each other and how patterns arise from those relationships. Emergency scenarios pose an interesting challenge to operation as well, since few to no labeled data are available. Consider an equipment shutdown, for example, which leads to several alarms being triggered simultaneously not only in the particular equipment, but also throughout the plant. The presence of so many alarms is very difficult to manage and almost impossible to track safely. Detecting fundamental anomalies that might threat the safety of the plant based on changes from previous stable states is highly desirable.

Such unsupervised monitoring relies on several factors to be successful. Initially, for any unsupervised approach, the

Corresponding concerning this article should be addressed to K. Funatsu at funatsu@chemsys.t.u-tokyo.ac.jp

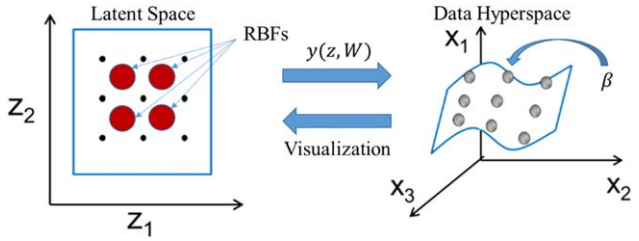

Figure 1. GTM overall concept representation.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

quality of the information available is fundamental for the development of trustworthy models. Any real dataset suffers from noise and redundant information, which if taken into account for modeling can mask the true relation between different features and, therefore, different samples. Dimensionality reduction plays an important role in this aspect, identifying regions with similar characteristics and filtering irrelevant information from data. Principal component analysis (PCA) is one of the most widespread methods used for process monitoring,[6] which relies on assessing linear correlation between different process variables, so to achieve dimensionality reduction of highly correlated variables. Its use was so successful that several PCA-based MSPMs were developed, such as, for instance, dynamic PCA (DPCA),[5] recursive PCA,[7] distributed PCA,[8] and maximum-likelihood PCA.[9] Despite its linear nature, extensions were developed to overcome this issue and deal with non-linear systems, such as kernel PCA.[10] Other methods also tackle nonlinearity from scratch, such as support vector machines,[11] Gaussian Mixture Models,[12] generative topographic mapping (GTM),[13] and even the use of inferential models.[14]

Unsupervised fault identification relies on data discrimination, which depends on how similar samples are to each other. From a practical point of view, process monitoring methodologies should be applied to more complex, real industrial scenarios. Given that PCA in inherently linear, GTM, as a non-linear probabilistic technique, is more suited to handle such complexity well. GTM has a probabilistic nature, where each sample plotted in the latent space has a unique probability distribution (PD), a fingerprint, associated to each point pre-established in the latent grid. Assuming that samples with correlated PD profiles represent data with similar characteristics, GTM can be used for fault identification and dimensionality reduction simultaneously, including discrimination of normal and anomalous data.

The main issue with this approach, however, is how to express this similarity in a way that the inner variations of samples can be overlooked, but still keeping the core relationship between data belonging to similar clusters. Samples belonging to the same cluster might not be highly correlated to every single other sample in it. Graph theory deals with networks that model pairwise relations between objects,[15] where two basic elements are always present: nodes (samples) and edges (connections). For this work, each sample is connected to those whose GTM probability profile correlation is higher than a given threshold. In the end, a web of connections establishes a network, where the density and number of connections unravel clusters with different characteristics. This combination of GTM and Graph theory, therefore, is proposed for unsupervised fault detection. GTM highlights important data information and calculates similarity between samples. Graph theory creates a network and clusters it in normal and anomalous groups.

Two case studies were defined for performance comparison. Initially, an artificial dataset with different types of anomalies was created. Second, the Tennessee Eastman Process (TEP)[16] was considered for validation of the methodology. The proposed method was compared against unsupervised PCA, DPCA, GTM, and graph theory-independent approaches and supervised PCA, DPCA and GTM. Dimensionality Reduction and Graph Theory section presents a review on dimensionality reduction and graph theory. Basic Structure section presents all fault identification methods considered for comparison in this work. Process Monitoring section describes in detail the proposed method. Results and Discussion section presents several results, discussing the impact of different methodologies on anomaly detection. Conclusion and Final Remarks section presents final remarks and future work.

Dimensionality Reduction and Graph Theory

Principal component analysis

For systems with too many variables or with nonlinear features, visualizing the relationship between inputs and outputs (states) can be rather complicated. PCA is the most straightforward linear approach known, where process variables are converted into linearly uncorrelated variables called principal components (PC), via an orthogonal transformation.[6] Equation 1 shows the basic concept behind PCA

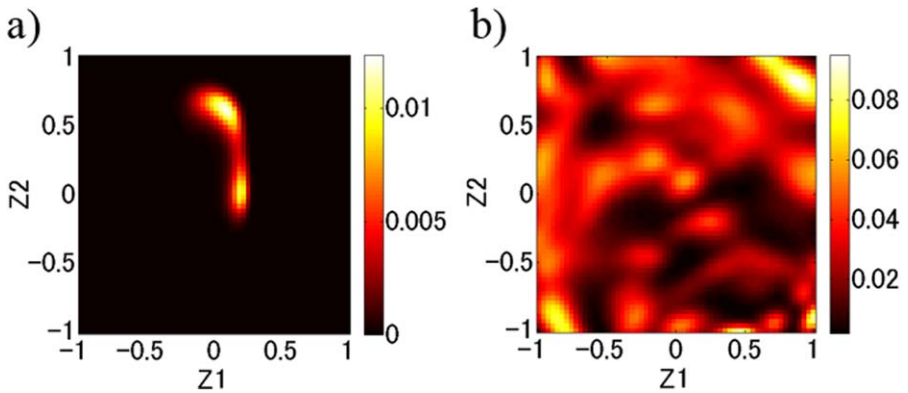

Figure 2. GTM PD heat map for (a) one sample and (b) a dataset.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.2 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.3 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0.2 & 0 \\ 0.2 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0.4 \\ 0 & 0.3 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0.3 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0.2 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 \end{bmatrix}$$

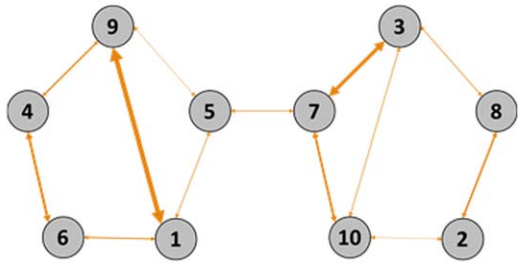

Figure 3. Schematic representation of a weighted AM and its respective graph.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$X = TP^{\mathrm{T}} + E \tag{1}$$

Where X is the original dataset matrix, T is the score matrix, P is the loading matrix, and E is the residual matrix. P relates X and T, allowing the projection of X values onto the transformed space T, whose column vectors are the PCs. Each PC has a contribution to the original information contained in X proportional to their eigenvalue. This can be translated as the equation described in Eq. 2

$$\mathrm{C}_{t_i} = \frac{\sigma^2(t_i)}{M} \tag{2}$$

where C_{t_i} is the component contribution for PC t_i and M is the number of input variables. To select only relevant information, PC with reduced relevance are excluded, keeping only the ones whose accumulate component contribution is below a given threshold. Heuristics mention 99% of component contribution as an acceptable value for selecting relevant PC.

Dynamic PCA

DPCA is an extension to regular PCA where the introduction of dynamic features aim to represent better nonlinear time series systems. The approach is rather simple, where time shifted variable information is added as extra variable, establishing a relation between current and past samples.[17] Equation 3 shows how the new dataset is represented

$$X_{\mathrm{Dyn}} = [X_1 \quad X_2 \quad \ldots \quad X_d] = \begin{bmatrix} x_{d+1} & x_d & \ldots & x_1 \\ x_{d+2} & x_{d+1} & \ldots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_N & x_{N-1} & \ldots & x_{N-d} \end{bmatrix} \tag{3}$$

where X is the original dataset, N is the total number of samples, and d is the sample delay. x_n is a row vector with all variables for the nth sample. Essentially, DPCA has the same approach as PCA, only with time shifted duplicate vectors. From this premise, all analysis related to PCA, such as determining the optimal number of PC, for example, apply to DPCA as well.

Generative topographic mapping

GTM is a widely used technique applied for visualization of data with several variables. It consists of a probabilistic nonlinear approach, where a low-dimensional latent space grid, usually 2-D, is represented as a high-dimensional manifold on the original data space. This manifold approximates the original data, modeled by a Gaussian function. If the transition between both spaces were to be regulated between the grid and the original space, the computational load would be too high. To cope with that, an intermediary layer of radial basis functions (RBFs), also Gaussian, is created and its parameters are determined via the Expectation-Maximization (EM) Algorithm.[18] Figure 1 shows the schematics behind GTM.

RBFs are embedded in $y(z, W)$, which defines the non-Euclidean manifold and connects both spaces. Two parameters are optimized via EM during training: W and β. The former is a parameter matrix that regulates RBF weights and the latter is the inverse variance scalar for all Gaussians defined in the hyperspace. Once the map is trained, it is possible to determine for each sample the likelihood of it belonging to each latent point in the grid, establishing a PD profile. Such profiles can be represented as individual heat maps or as one plot for all data, as Figure 2 suggests.

PD profiles are unique considering that the variables in hyperspace have different values for each sample, which, for this work, allows the similarity assessment of all samples on the same basis.

GTM relies on a set of hyperparameters for its utilization: latent grid size, number of RBFs, width of RBFs, and regularization parameter λ. The optimal value for each parameter

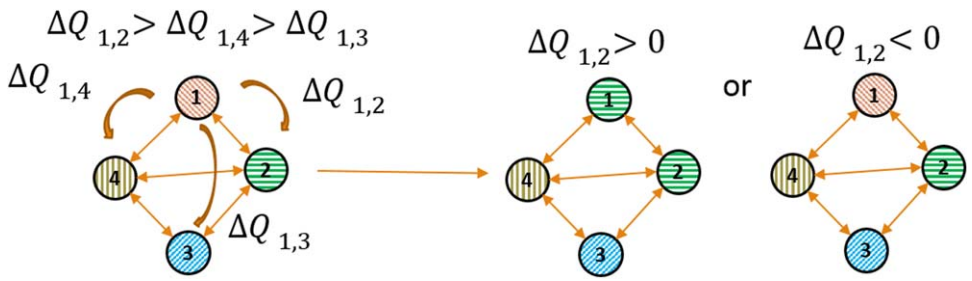

Figure 4. Modularity gain test, where different background patterns indicate different communities.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

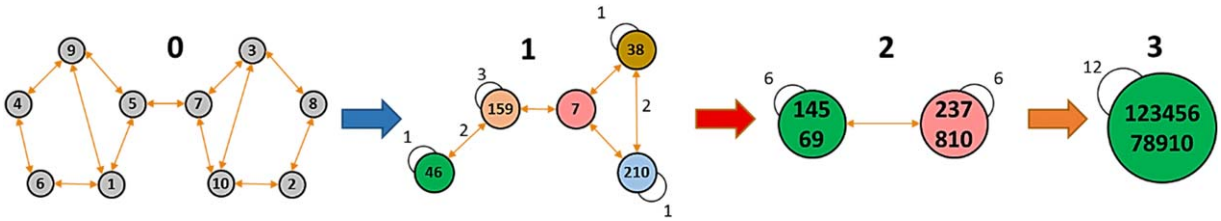

Figure 5. Graph evolution according to LCF algorithm.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

is usually determined via exhaustive search, using cross-validation to look for the minimization of reconstruction error (RE), that is, distance from the manifold, once data is recreated into the original hyperdimensional space. Root mean squared error is usually used as an index for such assessment, which does not take into account the smoothness of the map and the need for significant sample PD profiles. From this premise, Root mean squared error of midpoint (RMSEM) is used,[19] where midpoints to those existent in training data are used for accuracy assessment. If those samples can be predicted accurately, then not only training data has high prediction accuracy, but also the regions in between, preventing overfitting and ensuring map smoothness. RMSEM is calculated according to Eq. 4

$$\mathrm{RMSEM}=\sqrt{\frac{\sum_i^l \|x_i^{\mathrm{mid}}-x_{\mathrm{rec},\ i}^{\mathrm{mid}}\|^2}{l(l-1)/2}} \tag{4}$$

where l is the number of midpoints selected, x_i^{mid} is the ith midpoint and $x_{\mathrm{rec},i}^{\mathrm{mid}}$ is the respective reconstructed sample. Midpoints are sampled randomly from all possible combinations of training data, usually in a greater number than the original dataset.

Graph theory

Basic Structure

Graphs are symbolic representations of networks that model pairwise relations between objects.[20] For practical purposes, all graph-related structures presented in this work will be referred as graphs. There are two basic elements for every graph: nodes and edges. The former represents observations (samples) and the latter indicates connections between those observations. For a given dataset, adjacency matrix (AM) formalizes this web of connections, by representing all connections via a square matrix whose size is directly related to the number of observations available. Figure 3 shows an example of such representation.

All null values show that there is no connection between respective pair nodes. Values different from zero, conversely, reveal links between nodes, where the strength of the connection is correlated to the respective adjacency value. AM is the core element of any graph, from where graph analysis, visualization, and clustering is possible. For this work, discrimination between different data states is essential; therefore, focus on graph clustering (GC) is required.

Graph clustering

Albeit all approaches take into account graph features for discrimination, GC can have very distinct algorithms, such as spectral partitioning (SP)[21] and Newman–Girvan algorithm (NGA),[22] for example. Some drawbacks, however, are common and make discrimination difficult. SP lacks a termination criterion for optimal clustering and NGA relies on betweenness,[15] a graph centrality measure, for finding important hubs in the graph for clustering assessment, which may not be available for a given graph.

To cope with some of these limitations, Louvain community finding (LCF)[23] is an algorithm with interesting characteristics, based on a metric that, generally speaking, evaluates the density of edges within a group, called modularity.[24] LCF algorithm can be split in two steps: local modularity optimization and graph update. The initial assumption considers a weighted graph of N nodes, where different clusters are assigned to each node, that is, there are as many clusters as nodes. From this framework, a maximization of modularity is pursued, according to the pseudoalgorithm below:

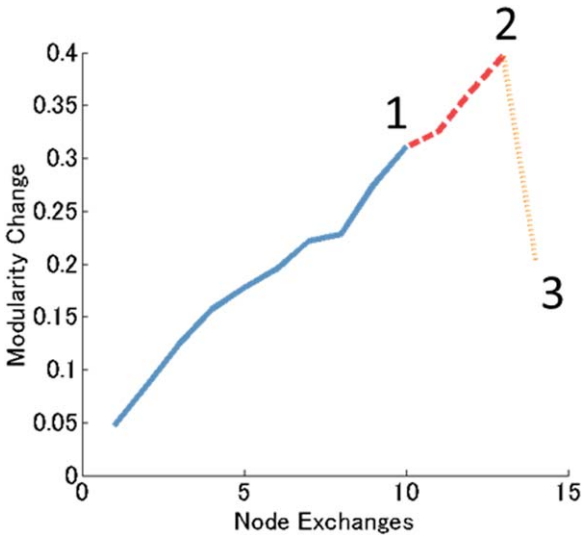

Figure 6. Modularity evolution during LCF cycles.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

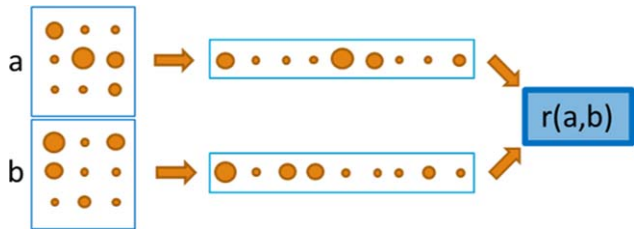

Figure 7. Correlation assessment between two samples using the same GTM grid.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

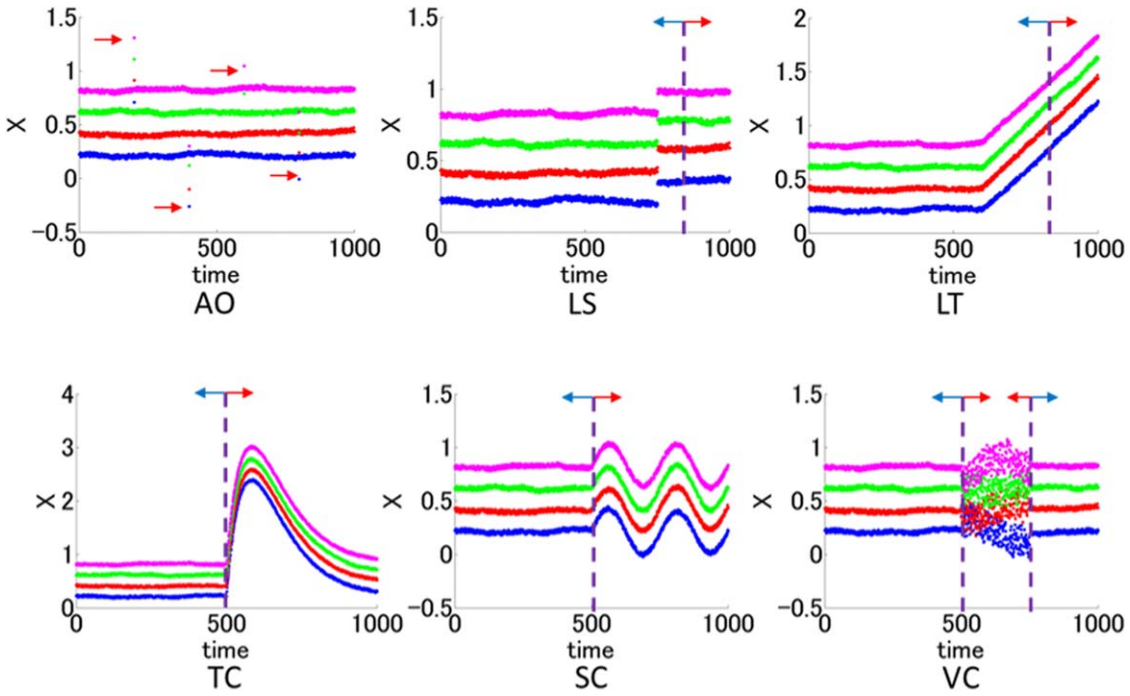

Figure 8. Artificial dataset.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

1. For each node i, consider all neighboring communities j of i.

2. Compute modularity gain ($\Delta Q_{i,j}$) when i moves to each community j. i moves to the cluster with maximum gain, if the gain is positive. Otherwise, i stays in its original community. Figure 4 shows the schematic representation of step 2 for one node, when tested against three other communities.

3. Test modularity gain for all nodes in sequence, till no further improvement is achieved.

Modularity gain is calculated as described in Eq. 5

$$\Delta Q_{i,j} = \frac{k_{i,j}}{2m} - \frac{k_i \Sigma_{\text{tot}}}{2m^2} \tag{5}$$

where m is the total sum of edge weights in the graph, $k_{i,j}$ is the sum of edge weights from i to j, k_i is the sum of edge weights incident to i and Σ_{tot} is the sum of edge weights incident to nodes in j. Second phase is the graph update, where all nodes of each community are condensed into a single node, keeping in mind that edges between nodes of the same community lead to self-loops. After the update, both steps are repeated until no more modularity gain is achieved. Figures 5 and 6 show graph and modularity evolution during LCF cycles for a trivial example.

After the second cycle, the graph reaches a modularity peak, indicating that this is the optimal configuration. By observing the original graph on cycle 0, one can easily see that it corresponds indeed to the best clustering scenario. Any further clustering beyond that results in a modularity drop. Appendix shows LCF pseudocode.

Process Monitoring

Principal component analysis and DPCA

Both PCA and DPCA can be used for process monitoring using two indexes, T^2 and Q, which evaluate data variation and prediction residuals, respectively,[25] as shown in Eqs. 6 and 7. Those values are calculated for each sample, where a $Q \times T^2$ plot is used to indicate discrimination between groups

$$T_n^2 = \sum_{i=1}^{k} \left(\frac{t_{ni}}{\lambda_i}\right)^2 \tag{6}$$

$$Q_n = \sum_{i=1}^{M} (x_{ni} - \hat{x}_{ni})^2 \tag{7}$$

where M is the number of input variables and k is the number of PC selected. t_{ni} is the score component for the nth

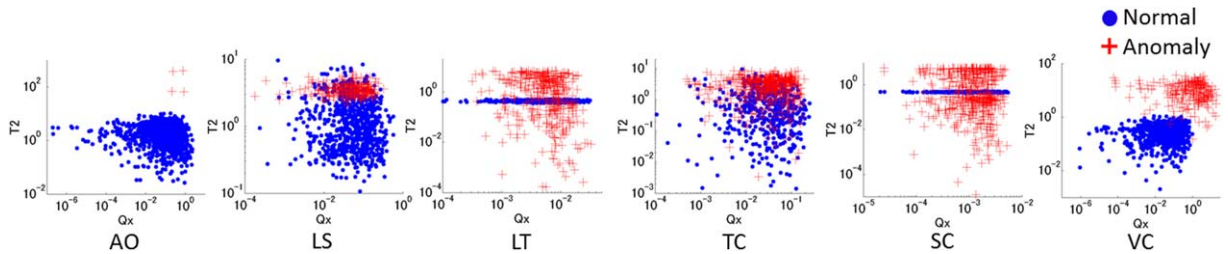

Figure 9. Unsupervised PCA $Q \times T^2$ plots for ADS.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

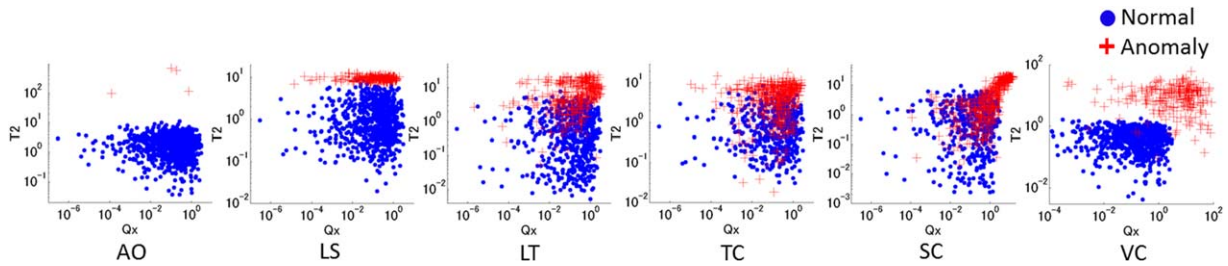

Figure 10. Supervised PCA $Q \times T^2$ plots for ADS.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

sample and ith t-score variable and λ_i is the estimated standard deviation of this t-score. $\hat{x}$ is the estimated input given k PC for data reconstruction.

Data belonging to different states would have different ranges of T^2 and Q, allowing for data discrimination and, therefore, process monitoring. The usual approach to this technique is supervised, where the reduced model is trained only with data from a particular state, looking for all data that deviates from it. Unsupervised training is also possible, where the model uses all data available and tries to establish two or more groups with clearly distinct T^2 and/or Q ranges.

Generative topographic mapping

Knowing that latent space approximates the hyperdimensional space, trained GTM maps can be used for process monitoring, once data reduced to a latent framework is replotted into the original space, giving a RE that can be calculated,[26] according to Eq. 8

$$\mathrm{RE}_n = \sqrt{\sum_{i=1}^{M} (X_n - X_{\mathrm{GTM}n})^2} \tag{8}$$

where X_n and $X_{\mathrm{GTM}n}$ are the original and reconstructed nth sample, respectively. Similar to PCA, both supervised and unsupervised approaches are possible, where the former only uses one previously defined "normal" group for map training and the latter uses all data for training. The purely unsupervised approach performs poorly, however, since the map fits both normal and outlier samples into the map, where there is no sensible difference between calculated REs.

Graph theory

Graph theory is being incorporated in this work since its visualization and clustering capacities match the data similarity concept embraced by GTM. For similarity assessment, however, one does not need latent variables and probabilistic distributions necessarily. If enough variables are available, data comparison can be calculated from them for each nth sample. By skipping GTM, a process monitoring method based solely on graph theory can be achieved, where LCF algorithm is used for clustering.

Proposed method—GTM and graph theory combined approach

The proposed approach is based on two key elements: extraction of essential information and effective data clustering. GTM reduces data to a 2-D latent plot, removing redundant and irrelevant information from the original dataset. Every sample in the latent space has a unique PD profile, which is used for similarity assessment, as represented schematically in Figure 7.

Each PD can be expanded in a vector, which then is used for squared Pearson product–moment correlation coefficient (r^2) calculation. One aspect that needs to be taken into account, however, is that the size of the grid has a direct impact in the sparseness of the vectors and the skewness of the correlation obtained. The bigger the grid, the smaller probability values on each point tend to be. Ideally, then, a small grid would be desirable, to minimize the sparseness effect. This is not realistic, though, given the fact that GTM grids have to be somewhat refined to be able to discriminate data in the first place. To cope with this limitation, a sliding window approach for r^2 calculation is proposed. While keeping grid size big, r^2 is calculated locally via a squared window that slides through the entire grid. The average r^2, then, is calculated and used as the similarity index.

Once similarity assessment is finished, AM can be constructed. For any given AM, however, only values equal to zero establish no connection between nodes. Regardless of how poorly correlated two samples might be their correlation will not be exactly equal to zero. It is necessary, thus, a similarity threshold where all values below it are considered null. The threshold should not be too high as to ignore important connections between samples and not too low as to consider correlation by chance. While it is true that LCF takes into account the strength of connections for clustering and it can handle low edge weights, a fully connected graph

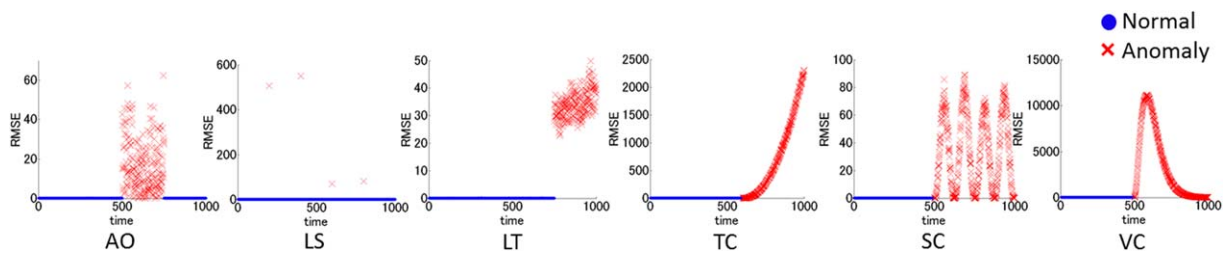

Figure 11. Supervised GTM RE plots for ADS.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

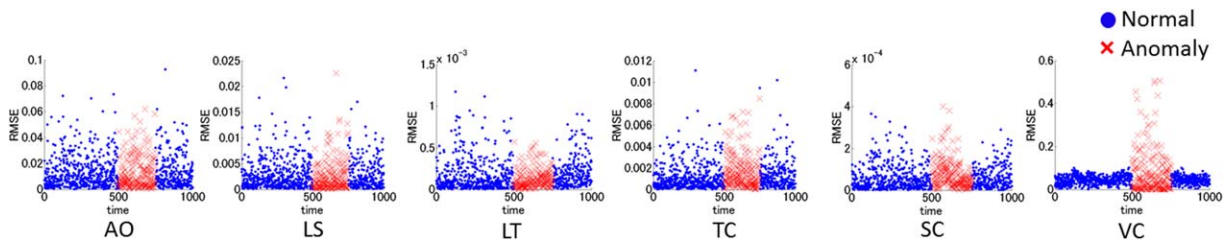

Figure 12. Unsupervised GTM RE plots for ADS.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

results in very poor clustering, even if most of the connections have negligible weights. Determining, then, a low threshold rather than a high threshold is more important, excluding correlation by chance and reducing considerably the number of unnecessary connections. For this work, the threshold was defined heuristically as 0.1, guaranteeing at least that the structure of the network will be preserved, leading to effective clustering. In addition, most correlation values are rather small ($<10^{-5}$). Considering the threshold equal to 0.1, therefore, guarantees that such negligible values will not be part of the network with a good margin. From this premise, varying the threshold to slightly higher and lower values has little effect on the overall structure of the network. The total number of connections is affected, yes, but with little impact in the communities formed.

With the AM built, LCF can cluster data into groups with similar characteristics. As an unsupervised method, the main point is to be able to identify among all clusters found, which one represents the state of interest. For fault identification, for example, it is assumed that faults are a minority of the system and due to their faulty nature, their behavior is usually more erratic, that is, less stable. Normal operational data, conversely, represents generally a majority of the samples available, where data itself is stable. From a graph theory perspective, this means that normal data has a far higher number of connected nodes combined with higher connection density, which is used as reference for identifying the optimal normal cluster. It is also important to notice that anomalous data might be detected as not one cluster, but several representing different fault characteristics.

The network can also deal with new data, since it can be integrated to the network without any concerns. Once correlation is calculated against all samples, connections can be established and the new observation is now part of the optimal, reduced network. Then, through LCF, it can be diagnosed whether that sample belongs to normal or anomalous clusters. For the work presented here, new data is integrated to the network for diagnosis, but soon discarded, that is, the original network remains intact. The main aspect discussed in this work is whether data can be discriminated in normal and anomalous data. If data is incorporated to the network as other samples are also incorporated, other issues like applicability domain[27,28] and data maintenance[29] within the graph itself arise, which is subject for another article.

Results and Discussion

To evaluate the potential of the proposed methodology, two case studies are presented. A simple artificial dataset is introduced initially to check how different types of anomalies can be detected. Besides that, TEP,[16] a simulated complex nonlinear system is used for corroboration of the methodology.

Fault identification performance is compared against PCA, DPCA, GTM, and graph theory monitoring-independent approaches, where both supervised and unsupervised PCA, DPCA, and GTM are tested. All networks presented in this work were created using Gephi,[30] a free graph visualization tool.

Artificial data set

An artificial dataset with four pseudorandom variables was created, with 1000 equally spaced samples. Five distinct anomaly scenarios were proposed, considering the most common deviations encountered in chemical plants. Additive Outliers (AO) represents spikes in data. Level Shift (LS)

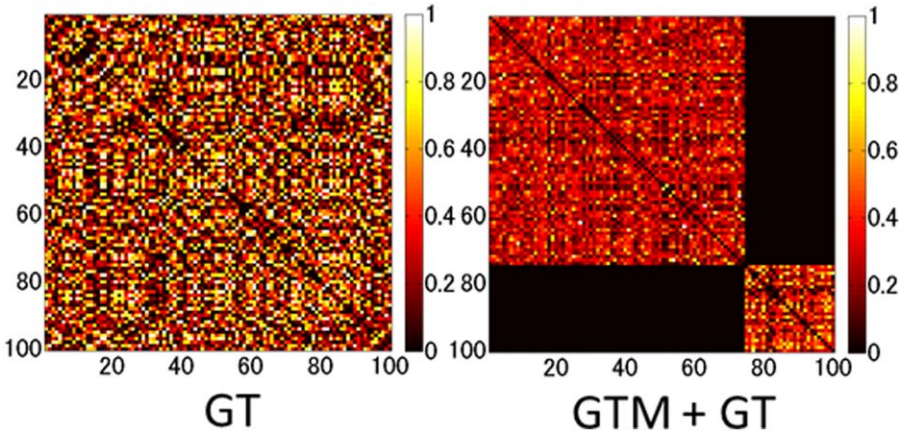

Figure 13. LS Similarity matrixes for GT and GTM + GT methods.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

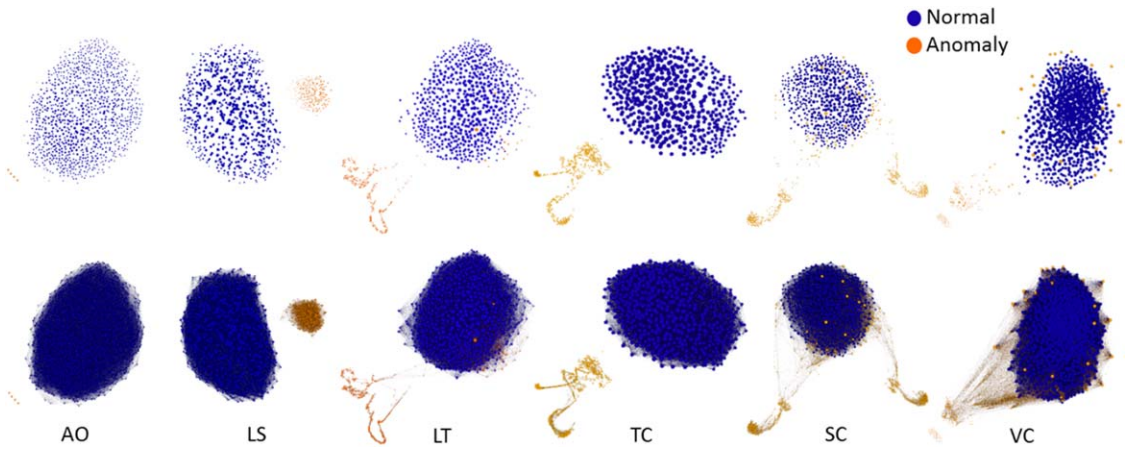

Figure 14. GTM+GT ADS networks where upper networks are represented without any edges and lower networks are represented with all available connections.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

indicate sudden and sustained change in state. Local Trend (LT) shows continuous drift over time. Transient Change is related to sudden changes damped over time. Seasonal Change (SC) presents oscillatory faulty behavior. Finally, Variance Change (VC) shows changes in data variation. The disturbances occur for all variables simultaneously. Figure 8 shows data plots over time for each anomaly scenario.

For unsupervised PCA, $Q \times T^2$ plots shown in Figure 9 indicate poor monitoring performance for PCA, since only AO and LS scenarios could be split in two groups. The hardship on unsupervised monitoring lies on how to discriminate groups with distinct characteristics effectively. PCA's simplistic approach and inherent linear capabilities show how poorly PCA distinguishes time varying anomalies.

Supervised PCA can be seen in Figure 10, where a mild improvement is obtained, even though still some scenarios cannot be discriminated efficiently. Despite the supervised knowledge available, PCA's linear nature still takes a toll on the overall fault identification performance.

As for the dynamic aspect of it, there are positive and negative aspects of such insertion that should be mentioned. While it is true that DPCA gives a better insight on the relationship between current and past samples, harm can also come from it. In cases where faults are instantaneous, like AO, not only outliers are detected as anomalies, but also nearby normal data. The use of DPCA should be saved for scenarios which clearly benefit from it, such as TEP. In addition, for this simple case study, the main goal is to motivate

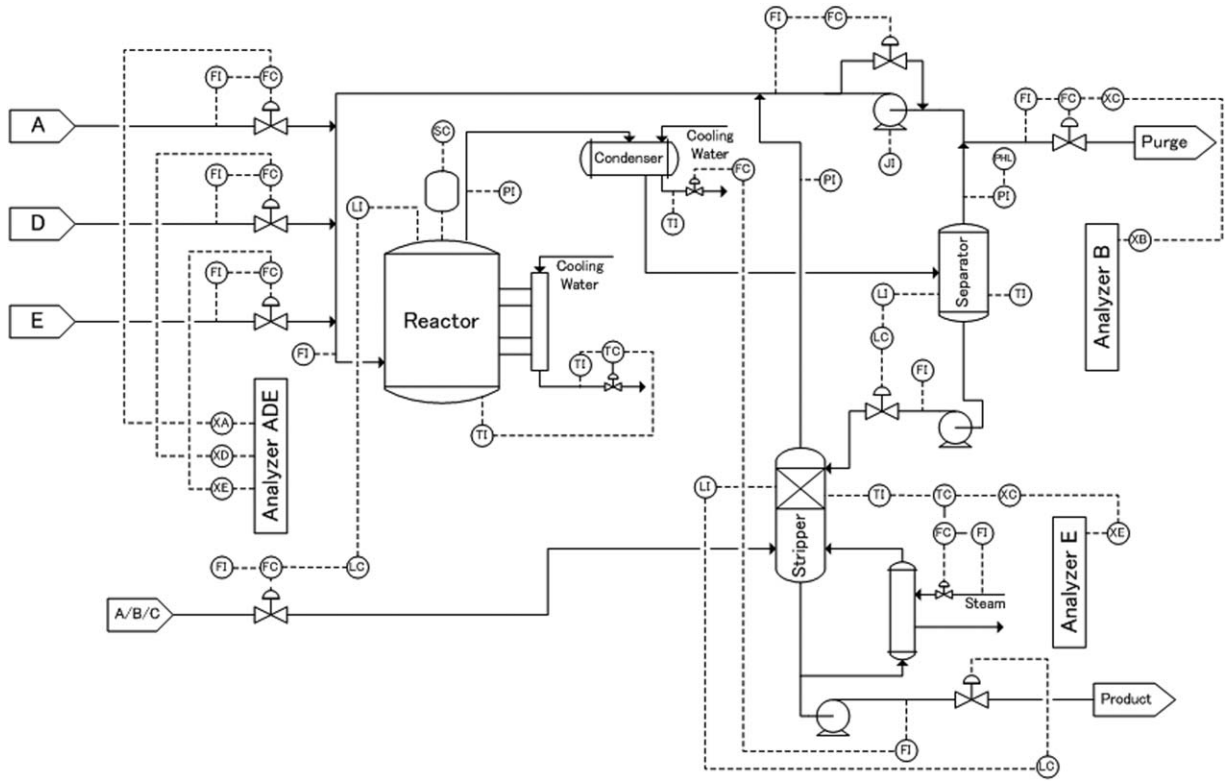

Figure 15. Tennessee Eastman Process flow sheet.

Table 1. Faults Defined in the TEP Process.

Fault ID	Description	Type
F1	A/C feed ratio, B composition constant	Step
F2	B composition, A/C ratio constant	Step
F6	A feed loss	Step
F8	A, B, C feed composition	Variation
F13	Reaction kinetics	Slow Drift

the use of the proposed method, relying on the nature of dimensionality reduction and network clustering. The use of DPCA is, therefore, not as interesting. DPCA analysis is, thus, not shown here, so to be explored in its fullest in the TEP section.

As for GTM, there is a clear difference between unsupervised and supervised approaches. The latent map structure is highly dependent on the data used for training. When only normal data is used for map training, anomalous data clearly has a greater RE, as seen in Figure 11 for all cases. Unsupervised results, conversely, show a poor, yet predictable scenario. The concept of RE assumes that samples not belonging to the map domain will have higher RE than the ones used for the delimitation of this domain. Since all data, both normal and anomalous, are being used for training, there is no sensible difference between faulty and normal samples, as Figure 12 demonstrates. Unsupervised GTM, therefore, cannot be used for fault identification when RE is used as an index.

Graph theory also had poor performance, since its similarity matrixes could not detect any difference between normal and faulty data. Figure 13 compares, for example, LS scenario for Graph Theory and GTM/Graph Theory. If the similarity matrix cannot recognize any patterns, the network has similar connection density everywhere, resulting in one single cluster or various small ones completely disconnected. This is the result of all data complexity being reduced to four features, which are not enough for discrimination of different states. Furthermore, all data information, both relevant and irrelevant are being taken into account for similarity assessment. The results are similar for the other cases as well. The proposed method, conversely, deals with filtered information, which clearly improves the resolution of the similarity matrixes obtained.

Once GTM can select the most important data features, discrimination becomes much more evident. This case study motivates the use for the proposed method as a tool for fault identification, where the networks shown in Figure 14 can show such discrimination. Upper networks present all samples without connections, so to ease the visualization of different clusters. Lower networks show the relationship between samples in their entirety, presenting all nodes and edges.

For all cases, it is clear that one densely connected cluster is formed, where anomalous clusters are just scattered. Particularly for time varying faults, it can be seen that discrimination is not perfect, but close to it. In LT, the variation of the two initial anomalous samples was not big enough to compensate the inner variation of normal data. It was, therefore, considered as normal. For SC, within the oscillatory behavior, each sample that crossed the normal state was said to be normal. As for VC, the samples whose change in variation was not enough to compensate the noise of the normal state were still considered as normal. GTM treats every sample independently of time, which means the dynamics are not being taken into account. Nonetheless, these results motivate the use of the proposed approach for process monitoring.

This simplified case study also gives some insight on the nature of certain anomalies and their plants. In the case of planned transitions, for example, the system could behave similarly to the fault presented in LT. Assuming that normal data is only the steady normal operation data, such transition would be detected as a fault. Once the characteristics of the transitioned cluster are known, however, one could take this

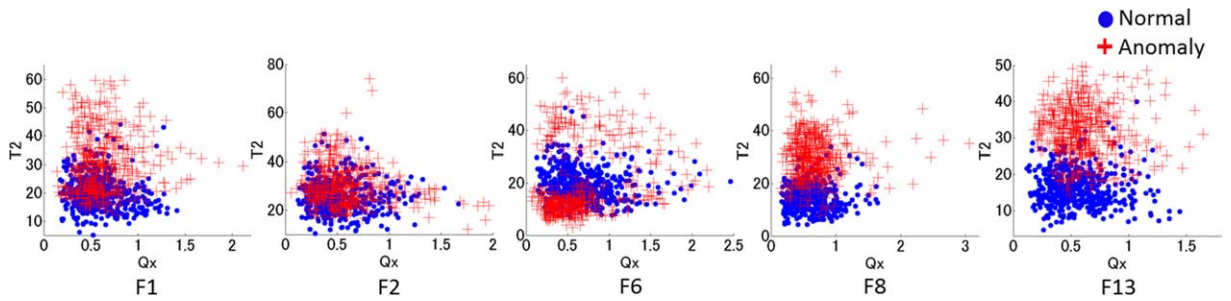

Figure 16. Unsupervised PCA $Q \times T^2$ plots for TEP.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

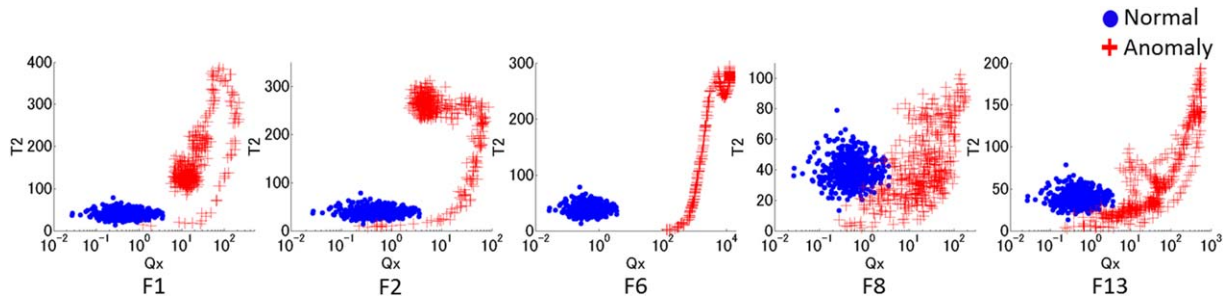

Figure 17. Supervised PCA $Q \times T^2$ plots for TEP.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

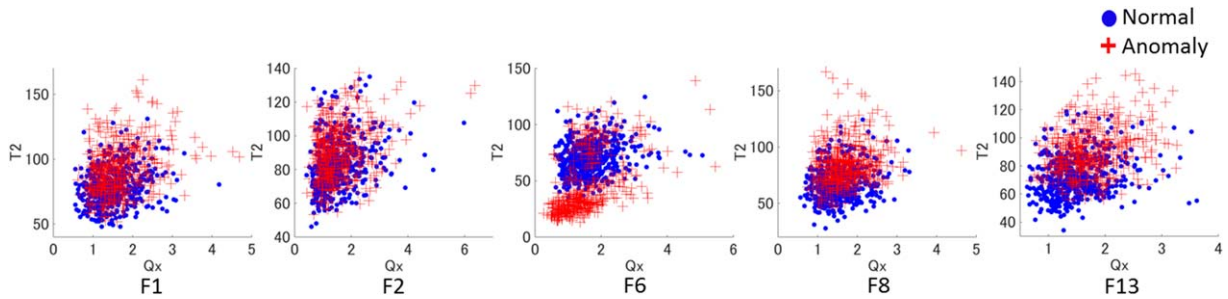

Figure 18. Unsupervised DPCA $Q \times T^2$ plots for TEP.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

information and incorporate it to the normal dataset, or hide it from the network, or even ignore alerts coming from that particular cluster. A more comprehensive framework is required, however, the potential for such analysis is present.

Tennessee eastman process

TEP is a realistic industrial process for evaluating process control and monitoring methods,[16] as shown in Figure 15. The system has eight components (A–H), 12 manipulated variables and 41 process (measured) variables. In this process, 21 preprogrammed faults (anomalies) are included, with both training and test data available. Five of those faults are explored in this work, according to Table 1. The dataset was obtained from the literature.[5]

As shown in Table 1, three distinct types of faults were considered: step change, random data variation, and slow drift over time. In addition, differently from ADS, faults now are local, happening in one variable instead of all simultaneously. This adds an extra level of complexity for identification.

PCA had different results for this process, compared with ADS. Unsupervised PCA kept performing poorly, as it can be seen in Figure 16. Supervised PCA, conversely, could handle discrimination rather well, according to Figure 17. Labeled data can indeed help a great deal for data discrimination and clustering, even if the technique is as simplistic as PCA. It is important to keep in mind, however, that such positive result only motivates the comparison with our proposed method, which is fully unsupervised.

The use of DPCA relies on knowing which delay is appropriately representing the existing dynamic information in the system. By referring to the literature,[17] the time shift d was considered to be equal to 2, implying that each sample is related to two immediate past samples. This approach shows interesting results, which depict how the system behaves once dynamical information is added, as it can be seen in Figures 18 and 19. In its unsupervised form, clear discrimination is rather troublesome. Despite knowing that labels are undisclosed, PCA presents a better potential for discrimination than DPCA. For the latter, most normal and anomalous data occupy similar regions, resulting in nearly impossible effective clustering. Once the supervised approach is considered, however, there is almost clear discrimination for all cases. Improvement is achieved given that labels are known. This gives an indication of how insertion of dynamics might even lead to poorer discrimination in unsupervised approaches, supporting the use of our proposed approach instead.

As for GTM, the notion of a fully unsupervised GTM approach for process monitoring fails similarly to the ADS scenario, where no discernible difference between normal and faulty REs can be detected. Supervised GTM, conversely, shows the benefit of labeled data again, where there is a clear discrimination between samples, as shown in Figure 20.

The proposed method, despite its unsupervised nature, presents good discrimination results, where again one big, densely connected cluster is formed, followed by other scattered clusters with no particular pattern. Figure 21 shows the networks generated for each scenario.

Finally, the graph theory approach had interesting results. The presence of more variables allowed a better assessment of correlation between samples, which allowed discrimination to a certain extent. For all cases but one faulty data identification performance was poor, even though for the single case where it worked the performance was on almost the same level as the proposed method. This shows the contribution of not only GTM, but also graph theory to the identification of data with different patterns.

Table 2 shows a summary, with the discrimination accuracy for all approaches. The scenarios marked with a hyphen indicate where discrimination was not remotely possible, which includes unsupervised PCA, DPCA, and GTM and

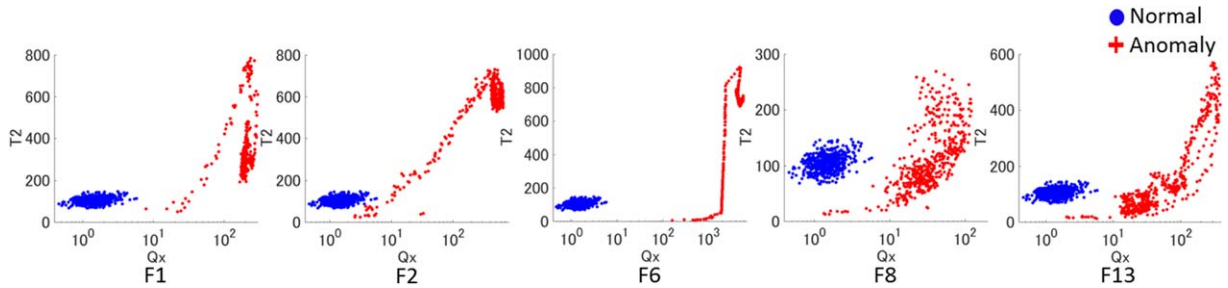

Figure 19. Supervised DPCA $Q \times T^2$ plots for TEP.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

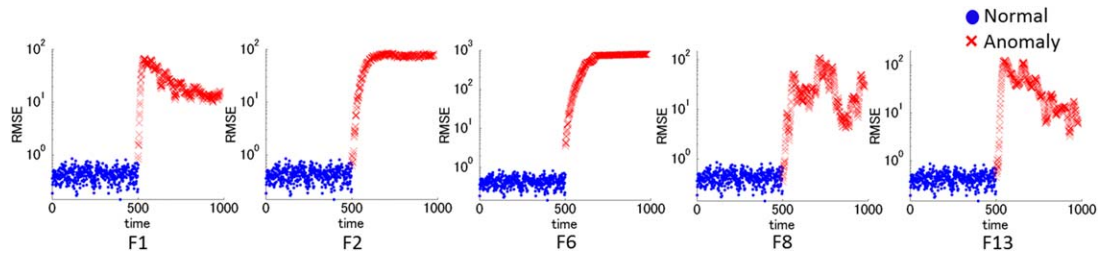

Figure 20. Supervised GTM RE plots for TEP.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

some graph theory cases. The most important aspect of this analysis is that the proposed method, as an unsupervised method, could discriminate normal and faulty data almost as well as fully supervised PCA, DPCA, and GTM approaches. This shows the potential that the proposed method has for fault identification.

GTM can extract all essential information required for data discrimination, while graph theory can take this information and structure it in a way that samples with different characteristics can be effectively clustered. One interesting aspect of graph theory that can be misleading is what criterion is being used for clustering. As Figure 21 shows, even clusters with completely different characteristics have some interaction, expressed by the connections between normal and anomalous data. Connections alone, however, are not important for clustering. What really matters is the density of connections within each cluster and the discrepancy between densities of different clusters. Once highly connected clusters are available, it is far more likely that this cluster will isolate itself from other minor, unimportant clusters in the network.

As a final analysis, it is of general knowledge that chemical plants do not express one single type of anomaly throughout its operation. The detection of multiple outliers, therefore, or at least the discrimination between multiple faults and normal scenarios, is highly desirable. With that in mind, two new TEP datasets were created by simply merging different anomaly datasets. The first scenario involves Faults 1 and 8, where anomalous data from Fault 8 was simply added to the end of the Fault 1 sequence, giving two distinct periods with different outliers. The second scenario, more extreme, also added anomalous samples to the end of Fault 1's data sequence, but this time from all remaining faults discussed in this work (Faults 2, 6, 8, and 13). Knowing that the proposed method had an overall good performance compared with the other methods presented, only its results are shown. The final networks can be seen in Figure 22 for both cases.

When both step change (F1) and random variation (F8) are assessed together, the proposed methodology can discriminate both anomalies from the normal data well, where the anomalies assessed as outliers are the ones undetected from F8 and normal samples are still highly connected to each other. When all faults are considered, the connectivity of normal samples is still present, however, there is much more interference of outliers in the normal cluster. Furthermore, there is a greater fuzziness present in the connections between faulty and normal clusters, indicating a greater interaction between samples in the GTM map.

Conclusion and Final Remarks

The use of unsupervised data rise some caution about the nature of the analysis and about the real availability of supervised data in practical applications. This work aimed to show how it is possible to identify normal and faulty data in a process, even when the notion of what is normal and what is anomaly is inexistent. This can be triggered from a simple lack of knowledge, given special scenarios such as emergencies, but it is mainly about a change of perspective based on unbiased analysis. More than the availability of knowledge, the reliability of such should be questioned. If not always, at least from time to time.

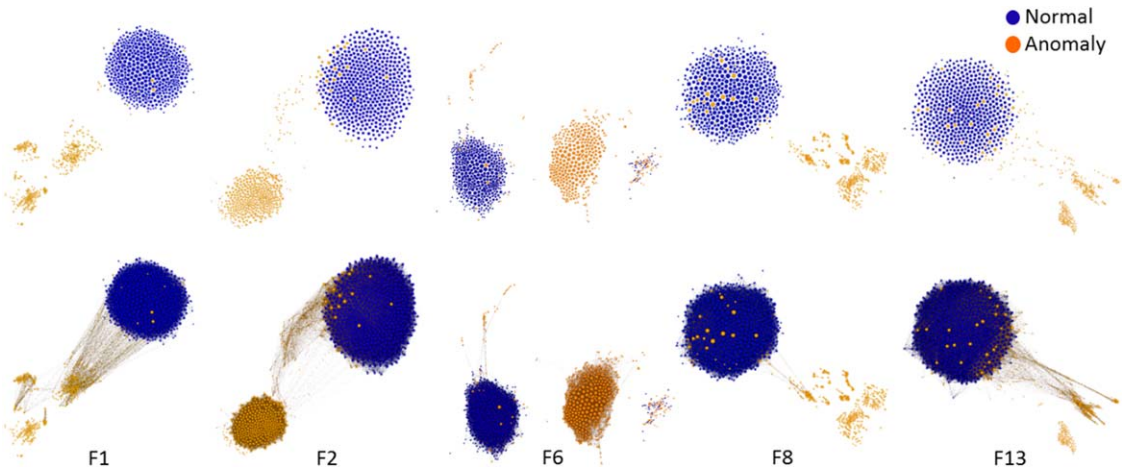

Figure 21. GTM+GT TEP networks where upper networks are represented without any edges and lower networks are represented with all available connections.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 2. Discrimination Accuracy for TEP faults.

	F1	F2	F6	F8	F13
GTM+GT	0.986	0.976	0.914	0.939	0.941
Sup. PCA	0.996	0.994	1.000	0.982	0.984
Uns. PCA	–	–	–	–	–
Sup. GTM	1.000	0.996	1.000	0.989	0.990
Uns. GTM	–	–	–	–	–
GT	0.710	–	–	0.961	0.698

Complete understanding of a chemical process subjected to several uncertainties is an illusion and, at most, a naive perspective. A false sense of security can be easily manufactured by routine itself or by epistemic overconfidence. Being able to perform an analysis free of biases and vices might prevent false information to be carried out, which consequently could avert accidents and/or economical mishaps.

By comparing two distinct scenarios, where labels are assumed to be completely reliable (supervised approach) or inexistent (unsupervised approach), this work showed that the proposed scenario involving the combination of GTM and graph theory performed closely to the supervised methods presented (PCA, DPCA, and GTM) for both case studies. Unsupervised process monitoring brings some challenges, specially related to extracting the true relationship between variables in a system. GTM highlights what is important from the dataset, and graph theory finds a way to bring this together into a concise, visual, and clear representation of different clusters, validating its use for fault identification of nonlinear processes in scenarios where labeled data might not be available.

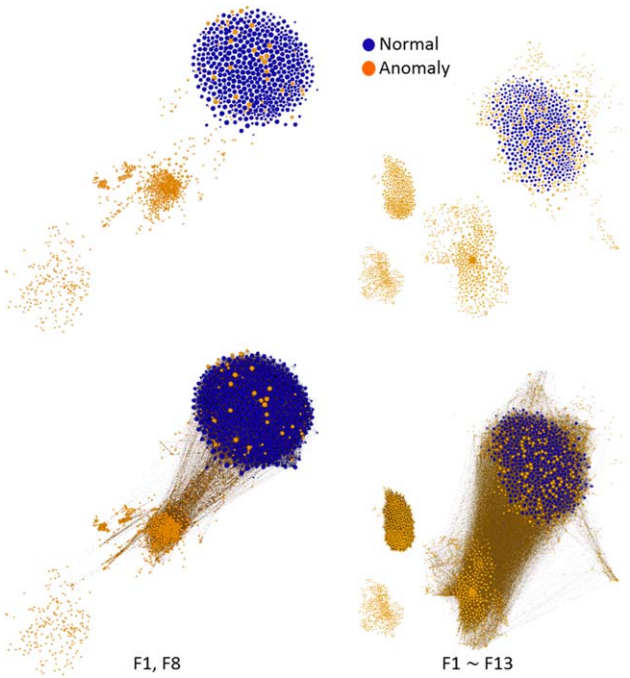

Figure 22. GTM+GT TEP networks for multiple anomalous scenarios where upper networks are represented without any edges and lower networks are represented with all available connections.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Finally, the use of graph theory for chemical engineering applications is rather scarce, if not inexistent, where most applications focus on representing the process itself as a graph and not the observations from this process. The results presented here reveal an unexplored potential arisen from the exploration of connections between data, which has space for further developments. Determining the optimal similarity threshold for each network, for instance, can improve the methodology even further, as also as adding a comprehensive framework for considering transition clusters within the network, leading to more achievements in the future.

Acknowledgments

The author acknowledges the support of the Core Research for Evolutionary Science and Technology (CREST) project 'Development of a knowledge-generating platform driven by big data in drug discovery through production processes' of the Japan Science and Technology Agency (JST).

Literature Cited

1. Kaneko H, Arakawa M, Funatsu K. Applicability domains and accuracy of prediction of soft sensor models. *AIChE J*. 2011;57(6):1506–1513.
2. Kourti T. Application of latent variable methods to process control and multivariate statistical process control in industry. *Int J Adapt Control Signal Process*. 2005;19(4):213–246.
3. Bersimis S, Psarakis S, Panaretos J. Multivariate statistical process control charts: an overview. *Qual Reliab Eng Int*. 2007;23(5):517–543.
4. Chiang LH, Russell EL, Braatz RD. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemom Intell Lab Syst*. 2000;50(2):243–252.
5. Russell EL, Chiang LH, Braatz RD. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemom Intell Lab Syst*. 2000;51(1):81–93.
6. Jolliffe IT. Principal Component Analysis. Tokyo: Springer, 2002.
7. Li W, Yue HH, Valle-Cervantes S, Qin SJ. Recursive PCA for adaptive process monitoring. *J Process Control*. 2000;10(5):471–486.
8. Ge Z, Song Z. Distributed PCA model for plant-wide process monitoring. *Ind Eng Chem Res*. 2013;52(5):1947–1957.
9. Choi SW, Martin EB, Morris AJ, Lee I-B. Fault detection based on a maximum-likelihood principal component analysis (PCA) mixture. *Ind Eng Chem Res*. 2005;44(7):2316–2327.
10. Lee J-M, Yoo C, Choi SW, Vanrolleghem PA, Lee I-B. Nonlinear process monitoring using kernel principal component analysis. *Chem Eng Sci*. 2004;59(1):223–234.
11. Kittiwachana S, Ferreira DLS, Lloyd GR, Fido LA, Thompson DR, Escott REA, Brereton RGl. One class classifiers for process monitoring illustrated by the application to online HPLC of a continuous process. *J Chemom*. 2010;24(3–4):96–110.
12. Yu J, Qin SJ. Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. *AIChE J*. 2008;54(7):1811–1829.
13. Yu J. A nonlinear probabilistic method and contribution analysis for machine condition monitoring. *Mech Syst Signal Process*. 2013;37(1–2):293–314.
14. Masuda Y, Kaneko H, Funatsu K. Multivariate statistical process control method including soft sensors for both early and accurate fault detection. *Ind Eng Chem Res*. 2014;53(20):8553–8564.
15. Harary F. *Graph Theory*. Colorado, USA: Perseus Books, 1994.
16. Downs JJ, Vogel EF. A plant-wide industrial process control problem. *Comput Chem Eng*. 1993;17(3):245–255.
17. Ku W, Storer RH, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis. *Chemom Intell Lab Syst*. 1995;30(1):179–196.
18. Bishop CM, Svensén M, Williams CKI. GTM: the generative topographic mapping. *Microsoft Research*. 1998;10(1):215–234.
19. Arakawa M, Miyao T, Funatsu K. Development of drug-likeness model and its visualization. *J Comput-Aided Chem*. 2008;9:70–80.
20. Wasserman S, Faust K. *Social Network Analysis: Methods and Applications*. Cambridge, United Kingdom: Cambridge University Press, 1994.

21. Wieling M, Nerbonne J. Bipartite spectral graph partitioning for clustering dialect varieties and detecting their linguistic features. *Comput Speech Lang*. 2011;25(3):700–715.
22. Newman MEJ, Girvan M. Finding and evaluating community structure in networks. *Phys Rev E*. 2003;69(2):026113.
23. Blondel VD, Guillaume J, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech: Theory Exp*. 2008; 2008(10):P10008.
24. Shen HW. *Community Structure of Complex Networks*. Berlin: Springer, 2013.
25. Chen Q, Kruger U, Meronk M, Leung AYT. Synthesis of T^2 and Q statistics for process monitoring. *Control Eng Prac*. 2004;12(6):745–755.
26. Svensen JFM. *GTM: The Generative Topographic Mapping*. University of Aston in Birmingham, 1998.
27. Escobar MS, Kaneko H, Funatsu K. Flour concentration prediction using GAPLS and GAWLS focused on data sampling issues and applicability domain. *Chemom Intell Lab Syst*. 2014;137(0):33–46.
28. Kaneko H, Funatsu K. Applicability domain of soft sensor models based on one-class support vector machine. *AIChE J*. 2013;59(6):2046–2050.
29. Kaneko H, Funatsu K. Database monitoring index for adaptive soft sensors and the application to industrial process. *AIChE J*. 2014; 60(1):160–169.
30. Bastian M, Heymann S, Jacomy M. Gephi: An Open Source Software for Exploring and Manipulating Networks. *International AAAI Conference on Weblogs and Social Media*. 2009

Appendix: LCF Pseudo-code

```
READ adjacencyMatrix
GET edgeList for each node from adjacencyMatrix
ASSIGN N clusters to N nodes
C_OLD = N
WHILE termination = false
  termination = true
  WHILE converged = false
    converged = true
    FOR i = 1 to N
      DECLARE ΔQ_MAT
      FOR j = edgeList(i)
        ΔQ_ij = modularityDelta(i,j) #Modularity calculation according to Eq. 1
        ΔQ_MAT(j) = ΔQ_ij
      END
      IF max(ΔQ_MAT) > 0
        ASSIGN i to respective max (ΔQ_MAT) cluster j
        converged = false
      END
    END
    ERASE K empty clusters
    C = C_OLD - K
  END
  IF C_OLD ≠ C
    GET selfLoops for each cluster #Edges between observations inside cluster C
    GET interConnections for each cluster #Edges connecting different clusters
    adjacencyMatrix = newadjacencymatrix(selfLoops, interConnections)
    GET edgeList for each cluster from adjacencyMatrix
    termination = false
  END
END
```

Manuscript received Oct. 14, 2014, and revision received Jan. 7, 2015.