博士論文

A STUDY ON DATA VISUALIZATION AND STATISTICAL PROCESS MONITORING FOR NONLINEAR SYSTEMS
(非線形システムを対象にしたデータの可視化および統計的プロセス監視に関する研究)

デ・ソウザ・エスコバール・マテウス

Matheus de Souza Escobar

To my parents, whose dedication and support led me to pursue my dreams on the other side of the Earth, and to all those whose path crossed mine during this amazing, insightful and revealing journey.

The first principle is that you must not fool yourself - and you are the easiest person to fool.

Richard Feynman

Acknowledgements

Immeasurable appreciation and deepest gratitude for the help and support are extended to the following persons and institutions, which in one way or another have contributed in making this study possible.

Prof. Dr. Kimito Funatsu, research adviser and supervisor, for his support, guidance, valuable contribution and provisions that benefited much in the completion and success of this study.

Dr. Hiromasa Kaneko, assistant professor, for his diligence, patience, and utmost support, whose valuable insights on a weekly basis were essential for the completion of this study.

MSc. Koji Kanma, friend and former tutor, for his friendship, support and help during my first years in Japan.

To all members of Funatsu Laboratory, for all cooperation and heated discussions, which only improved the quality of my work.

Ministry of Education, Culture, Sports, Science and Technology in Japan, for providing the funding necessary for my studies and stay in Japan.

Japan Science and Technology Agency (JST), for the support through 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'.

Abstract

In the realm of chemical processes, machine learning techniques are used to highlight important characteristics of a system by analyzing the relationship of variables and samples. Despite the myriad of applications available, one particular field is explored in this thesis: process monitoring. In order to do so, a combination of human knowledge, experience and statistical analysis is used, leading to information that can be used for future assessment, called supervised knowledge.

One might argue, however, about the reliability of the information available, since different aspects, such as human biases, atypical scenarios, poor data assignment or just the wrong mindset can lead to misinformed decisions and false conclusions regarding the nature of the dataset and the quality of obtained results. The main goal, thus, is to evaluate different aspects of data reliability, trying to challenge different notions often overlooked.

Initially, the distinction between supervised and unsupervised approaches is explored, where the former uses pre-existent labels for analysis and the latter uses only the relationship between variables and samples for assessment. Supervised approaches are said to be inherently better, since they provide more information about the process. As mentioned previously, however, always relying on this information might lead to misleading results. To challenge this notion, an unsupervised monitoring methodology is proposed, using a combination of Generative Topographic Mapping (GTM) and Graph Theory. GTM is a probabilistic technique, which highlights system features, reducing variable dimensionality. Graph Theory can then visualize this information, through a network, clustering samples with similar characteristics. Simulation data sets, Tennessee Eastman Process and a real industrial scenario are used as case studies for validation of the conjoined approach. The proposed unsupervised methodology performed as well as the supervised approaches presented, which motivates their use as a support tool for unbiased analysis of the system.

Complementary to this analysis, the study also focuses on applicability domain (AD) and fault detection, by analyzing how the definition of proper training and test data sets affect modeling. A flour data set is used as a case study, where distinct splitting scenarios are created to explore this concept, evaluating predictive modeling and anomaly detection capabilities. A flour dataset is used for this assessment, allied to generative algorithm tools. Generative Algorithm Partial Least Squares (GAPLS) and Genetic Algorithm-based Wavelength Selection (GAWLS) are used for modeling, where their non-deterministic nature is essential for evaluation of AD. Different models are created, resulting in several predictions for each sample available. By analyzing the relationship between precision, variation of prediction error, accuracy and the average of those predictions, AD can be assessed and anomalies on Y-values can be detected.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AD	Applicability Domain
AM	Adjacency Matrix
AO	Additive Outliers
DPCA	Dynamic Principal Component Analysis
EM	Expectation-Maximization Algorithm
ES	Experiment Splitting
FA	Force Atlas
FDG	Force-Directed Graphs
FN	False Negative
FP	False Positive
FR	Fruchterman-Reingold
GA	Genetic Algorithm
GAPLS	Genetic Algorithm Partial Least Squares
GAWLS	Genetic Algorithm-based Wavelength Selection
GC	Graph Clustering
GNA	Girvan-Newman Algorithm
GTM	Generative Topographic Mapping

LCF	Louvain Community Finding
LS	Level Shift
LT	Local Trend
MLP	Multi-Level Programming
MSPC	Multivariate Statistical Process Control
MSPM	Multivariate Statistical Process Monitoring
NC	Normal Cluster
NIR	Near-Infrared
OE	One Experiment
PCA	Principal Component Analysis
PC	Principal Component
PD	Probability Distribution
PLS	Partial Least Squares
QSAR	Quantitative Structure Activity Relationship
QSPR	Quantitative Structure Property Relationship
RBF	Radial Basis Function
RE	Remapping Error
RMSEM	Root Mean Square Error of Midpoint
RMSE	Root Mean Square Error
RS	Random Selection
SC	Seasonal Change
SNA	Social Network Analysis
SP	Spectral Partitioning
SR	Sample Random

SS	Sample Splitting
SYR	Sample Y-Ranking
TC	Transient Change
TEP	Tennessee Eastman Process
TN	True Negative
TP	True Positive
VC	Variance Change
YR	Y-Ranking
YS	Y-Splitting

Chapter 1

Introduction

It is fair to say that, at least in the realm of chemical processes, machine learning techniques are used with the following key element in mind: highlighting important features and characteristics of a process by analyzing the relationship of variables and samples belonging to this particular system. This notion, as general as it may be, can be applied for all sorts of different goals, such as pattern recognition, predictive modeling, classification, etc. Among all these applications, one stood out as particularly interesting, being the focus of this entire work: process monitoring.

The operation of any chemical plant requires attentive eyes and ears of a big task force, so to assure its smooth functioning. It is only natural that, over time, automated systems were developed, trying to rely less on human capabilities and more on data coming from the process itself. This is not to say that human experience and knowledge is unimportant, but rather that such burden should not rely entirely on their shoulders.

More concretely, process monitoring judges whether the process is currently in its normal operational state or it is suffering from an anomaly. This can be divided in two main tasks to be accomplished: fault identification and diagnosis. The former is concerned about finding anomalies, discriminating them from the normal states in the plant[1, 2]. The latter tries to evaluate which part of the system is responsible for the fault[1, 2]. Faults in the process can arise from hidden plant states, disturbances, controller malfunctions, and so on. Soft sensors[3], for instance, are affected tremendously by it, since faulty samples in the model database lead to poor model prediction. Process control also benefits from process monitoring greatly, where identifying faulty data is useful in alarm technologies[4] and hierarchical control systems[5], guaranteeing fast response to anomalous scenarios. By integrating the aforementioned elements, therefore, one main goal of this work is to explore fault identification and diagnosis capabilities for nonlinear systems, through useful applications.

1.1 Fault Identification

The notion of fault identification, or detection, in any process is of fundamental importance. It is directly related to a deep understanding of the process. Such knowledge can be brought via a phenomenological background, a heuristic background or a statistical background, which all have their merits and drawbacks.

1.1.1 Heuristics Fault Detection

Heuristics have little relevance as a sole concept for fault detection. Despite being a rather broad terminology, when it comes to applications in chemical engineering, it can be translated to approaches where experience and trial and error alone are taken into account when developing an anomaly detection system. This approach is limited to companies who are not technologically advanced or whose process are not complex enough to justify any changes. It may involve, for example, arbitrary definition of maximum and minimum ranges for key variables. While its sole use is not recommended, arbitrariness may be involved, as long as its use is not the key element behind one's methodology.

1.1.2 Phenomenological Fault Detection

Phenomenological models try to describe a body of knowledge, in a way consistent with a previously established theory, always trying to relate them with empirical observations. The models created are based on fundamental equations, which try to better represent the current scenario being analyzed. For chemical engineers, several equations can be incorporated to a model, coming from all sorts of topics: thermodynamics[6], transport phenomena[7], reaction engineering[8], unit operations[9], etc.

Once models are defined, parameters must be tuned in order to better represent the system being analyzed. Parameter sensitivity, optimization approach and AD are one of the few aspects to be considered carefully for proper modeling. One should, for example, be insightful enough to include only equations and inputs that are relevant and meaningful to the output variables presented, otherwise achieving proper modeling can be very complex and troublesome. From a practical standpoint, phenomenological models need to be very specific and finely tuned for each process, which can be rather time consuming. Furthermore, only systems that can be expressed by phenomenological models can be used for analysis, depending, thus, on the nature of equipment present.

Several works rely on phenomenological model-based fault detection, allying a complex, yet finely tuned, representation of the process with fault detection methodologies. Whether fault detection in embedded in a dynamic hybrid simulator[10] or associated to a control-system for fast response[11], the integration of phenomenological models is possible. It is important to keep in mind, however, that there is a great deal of work and expert knowledge that must be inputted in this analysis for proper modeling.

1.1.3 Statistical Fault Detection

Despite all the merits and drawbacks aforementioned, the statistical approach for modeling is presented in this work as particularly interesting, since it relies entirely on the feedback coming from the process itself. It could be argued that such approach is phenomenological as well, since it is reconciling empirical observations with models, but it differs from what is mentioned in the previous section due to its "black box" nature. There is no theoretical background that indicates the relationship between variables, but the analysis of data via statistical approaches.

One of the main benefits of a statistical approach is how broad, and to a certain extent simple, its strategy is. Once data is the most important element, and not the process, methodologies can be easily extended to other scenarios, without worrying about the intricate peculiarities of each system. This is not to say, evidently, that knowledge about the process should be disregarded, but rather that it loses its utmost significance in determining the relationship between variables. Insights regarding the process can still be useful for variable selecting, labeling of reliable information and so on.

The core idea is to extract essential information that allows one to represent the effect that input variables have on desired output variables[12], without really analyzing what is physically between those variables. Since fault detection goes beyond just representing the model, the methodology related to identifying anomalies can also rely on a statistical approach to do so. In this work, the proposed method that will be presented aims to take a fully statistical approach, benefiting from what was described.

Within data driven methodologies, this work focuses particularly on Multivariate Statistical Process Control (MSPC) and Monitoring (MSPM)[13, 14] approaches. The main concern here is to evaluate and visualize how variables and samples interact with the process and with each other. Thereby, a more complete understanding of the process can be achieved, where data discrimination is more objective and meaningful.

1.2 Fault Diagnosis

Being able to not only understand that something is wrong with the system, but also pointing out the source of a fault is the complementary step right after fault detection[15]. Fault diagnosis is complex and it relies greatly on experience. Closed loop systems are rather challenging, where despite faults attacking locally the process, the whole system acts suppressing this variable, making a clear diagnosis more troublesome[16].

1.3 On Supervised and Unsupervised Strategies

The most straightforward approach to process monitoring count on a supervised approach, where previously known "normal" and "anomalous" labels are given to each sample and a classification model, for example, can then be developed for future fault detection. One might argue, however, about the reliability of such labeling, given the right circumstances. The notion of supervision is deeply connected to certainty, where labels are usually accepted without further inquiries. This is not to say that everyone is forgetful about the transient nature of a process, but rather that labels inherently bring a sense of ease and trust to those looking at a process. Such mindset can be harmful due to several different reasons. As mentioned before, it is fair to say that processes are transient by nature, where the notion of stability is, at most, temporary. This is not to say that finding a steady data set is irrelevant, but rather that one should be careful with the data being chosen over an extended period. Some companies, evidently, are aware of this issue, leading to actions towards database maintenance.

Being aware of a process transient nature, however, hides in plain sight another issue, which, yet similar, brings a far more profound feature of human nature towards faults. It is of human nature to accept and internalize psychological biases. Biases are small practical rules, a natural part of human condition, which makes our lives easier and more predictable. They are directly related to all shortcuts that we establish throughout our lives, so to quickly respond to most situations we face on a daily basis[17]. Biases can be harmless or harmful, depending on so many different factors. The main point, though, is that biases deprive us from a fair judgment, by instead encouraging its propagation.

When we translate biases to our engineering reality, there is a common pitfall related to experience and knowledge. It is recurrent to rely on operator's knowledge for assessment of bottlenecks, normal operation, etc. This experience can be seen as truth, in a dogmatic way, or it can be trusted if the meets certain criteria. Regardless, analyzing the system from an unbiased perspective might lead to interesting insights. One of the key aspects of the work, therefore, lies on analyzing data purely based on the relationship between variables, an unsupervised approach, not relying on labels given for each sample. After that, the system might even be expanded later to one where experience is taken into account, by adding small, yet relevant pieces of information to the model created.

Still, it might be difficult for the reader to conceive how chemical plants would deal with completely unsupervised approaches, considering among other factors, the intricate knowledge engineer and operators might have about the process. Firstly, there is reliability. To what extent is such knowledge reliable? People, as aforementioned, are prone to human error, despite their best efforts. Secondly, how subtle can these anomalies be? What if an anomaly occurs due to a change of correlation between variables? Considering slowly drifting variables, for example, if a pipe is clogging, how soon can faults be detected based on human perception? Operators might lack the fine-tuning required for such task. The same reasons can be extended to current applications (software, sensor, etc.) installed in the plant. Within their reference data, which samples are normal and which are faulty might be questionable. It is true that questioning the use of supervised data is challenging, but a paradigm shift towards valuing unsupervised approaches might be beneficial, especially if one imagines using both approaches for a better understanding of the process. Relying only on process data can reveal optimal data sets to be incorporated in future evaluations, where then supervised approaches can be reintegrated to the process using those newly updated sets.

It is also important to highlight that for some scenarios, supervision is just inexistent. Emergency scenarios pose an interesting challenge to operation, since few to no labeled data is available. Once an equipment shuts down, for example, several alarms are instantly triggered. How to discern between alarms coming from the shutdown and the real cause behind the fault itself? Detecting fundamental anomalies that might threat the safety of the plant based on changes from previous stable states is highly desirable.

In order to develop unsupervised methodologies, though, various factors have to be considered to achieve successful monitoring. Firstly, the quality of the information available is fundamental for the development of trustworthy models. Real data sets have to deal with noise and redundant information, elements that might mask the true relation between distinct features and samples in the process. Dimensionality reduction, thus, plays an important role, isolating irrelevant information from the data set. One of the most widespread methods for process monitoring is Principal Component Analysis (PCA)[18], which assesses linear correlation between different process variables, so to reduce the dimensionality of highly correlated variables. Its use is so widespread that several PCA-based MSPMs were created, such as dynamic PCA (DPCA)[19], recursive PCA[20], distributed PCA[21] and maximum-likelihood PCA[22]. Likewise, some extensions were developed to overcome PCA's linear nature, such as kernel PCA[23] for nonlinear systems. Other methods not related to PCA also tackle non-linearity from scratch, such as Support Vector Machines (SVM)[24], Gaussian Mixture Models (GMM)[25], Generative Topographic Mapping (GTM)[26] and even the use of inferential models[27].

1.4 Applicability Domain

All principles presented here rely, evidently, on proper modeling, which, despite being an expected premise, can be highly overlooked by many works in the literature[28, 29]. This goes beyond the usual big historical data sets found in chemical plants, since, for any soft sensor assembled, faulty samples can have a negative impact on the performance. Smaller data sets, actually, might be more sensitive to anomalies, since the ratio between normal and anomalous samples is small. The key element disregarded is Applicability Domain (AD), *i.e.*, to what extent is a model applicable to predict future data. Along that aspect, issues related to how to define adequate training and test data sets for modeling also arise, leading to discussions that can be incorporated to the scope of this work, associating applicability domain to fault detection capabilities[30].

The main concern is how to define training and test data that lead to coherent results. There are some issues associated to data similarity that should be considered. If, for example, both training and test data are too similar, even with high prediction accuracy, the overall predictability of the model might be low. On the other hand, if both data sets are fundamentally different, modeling is meaningless and frustrating, since accuracy will always be low. The important element is Applicability Domain (AD), *i.e.*, to what extent is a model applicable to predict future data. A poor representation of training and test data sets can lead to misguided conclusions regarding the predictability and outlier detection potential of models created[31, 32]. It is important to notice that this is usually more troublesome when data is independent, *i.e.*, not time-series, since there is greater freedom to define which samples are used for model training and which samples are used for test of the methodology utilized.

One of the first approaches explored in this thesis towards process monitoring tried to associate AD to fault detection capabilities via the evaluation of a flour and protein content data set[30], evaluating the predictability and reliability of different sub-data sets. Once adequate modeling was discussed, fault detection capabilities were proposed, by using soft sensor monitoring indexes[33] and genetic algorithm modeling features[34, 35]. Genetic algorithms are non-deterministic and, therefore, may result in different regression models for each run. This information can be used for ensemble prediction[36], where several models are conjointly used for anomaly assessment by verifying the relation between distinct predictions.

1.5 Applications

Despite how developed the methodology, to what extent or how it can be applied to practical applications is essential. From this premise, the following explore two of the main feasible applications: soft sensors and process control.

1.5.1 Soft Sensors

Soft sensor, also called inferential models, are statistical models that aim to estimate process variables whose measurement is difficult[37]. There are several reasons why certain variables cannot be measured easily. Technical difficulties, offline measurement delays and high investment costs are a few of those causes. Soft sensors models are created, therefore, between variables that are easy to measure and those who are not.

One of the most widespread methodologies for modeling is Partial Least Squares (PLS)[38, 39]. Since its development, several techniques were derived from it, such as nonlinear PLS extensions[40] and variable selection variations[30, 34, 35]. PLS is but one methodology, though, where several other ones were developed, such as support vector regression[41] and artificial neural networks-based methods[42].

From a fault detection perspective, regardless of which methodology is involved, it is fundamentally important to know which data is relevant for the construction of a soft sensor model. If abnormal data is associated to the current database, the accuracy of the soft sensor created might decrease, affecting the prediction of regular samples yet to come. The aim of fault detection in this scenario is, thus, to detect preemptively whether these samples are normal or faulty. By doing so, one can prevent substantial changes on soft sensors' accuracy. This is intricately related to database maintenance, where the goal is essentially the same, to avoid the presence of outliers in a reliable data set.

1.5.2 Process Control

Process control applications for fault detection and diagnosis are related closely to hierarchical control and alarm management. For this work, process control applications will not be dealt with directly. Instead, its potential will be presented here and there to motivate future scenarios.

Hierarchical control assumes that the control structure of a system relies on different control layers, where higher layers can execute different functions. Monitoring is one of the most common tasks, where higher layers can evaluate whether controllers are operating optimally, or even, for our purposes, to see if the plant is going through an anomaly, triggering alarms that can alert operators of the nature of the fault and how to react quickly to it. Some low level controllers also rely on models that are updated based on data being fed to them constantly. Identifying anomalies is rather relevant in those cases as well, allowing the controller to maintain its performance, regardless of the anomalous scenario presented.

Alarm management[43, 44] is another key aspect, since triggering of alarms is a complicated issue in the chemical plant. The overwhelming presence of different alarms being triggered at the same time during and emergency scenario, for example, makes it difficult to identify the source of the problem, since alarms from all different features would be ringing simultaneously.

Process control applications on fault detection and diagnosis would hardly be used for anything else then higher levels of control, monitoring the chemical plant.

1.6 Proposed Strategies

1.6.1 GTM & Graph Theory Combined Approach

From a practical point of view, real industrial processes are intrinsically complex and nonlinear. Given that PCA itself is linear, little can be attested regarding its performance. GTM, conversely, is a nonlinear methodology relying on a probabilistic framework, being more suited to handle such complexity. Each sample plotted in the latent space has a unique probability distribution (PD), a so-called "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 faulty data.

After this matrix of similarities is calculated, however, one must express, or visualize, this similarity in a tangible way. Clusters must encapsulate similar data through the core relationship between samples, while understanding that samples belonging to the same cluster might not be highly correlated necessarily with all those samples. Graph Theory models pairwise relations between objects[45], by expressing them as a network. 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, this web of connections creates a weighted graph, where the number of connections and their density around the graph unravel data clusters with distinct features. This combined approach using GTM and Graph Theory is proposed, therefore, acting as a fault identification tool, keeping in mind, evidently, its unsupervised nature. GTM highlights important data information and calculates similarity between samples. Graph Theory not only clusters data in normal and anomalous groups, but also allows an objective and clear representation of all data in the process.

Once faults are properly detected, however, finding their source is also an important aspect of process monitoring. Two distinct approaches are taken into account, all derived or based from the proposed methodology: one from a GTM perspective and another from a Graph Theory point of view. The former uses GTM remapping error (RE) and dissimilarity index as criteria for diagnosis, by training a map only with the normal data set found. Then, one can find which variables exceed the error threshold defined by the normal data itself, so to point different candidates responsible for the fault. The latter uses the network structure itself to see whether new data belongs to a normal or anomalous cluster.

1.6.2 On Applicability Domain & Fault Detection

The analysis on AD portrayed in this thesis in intimately connected with soft sensor performance and adequate training and test data splitting, as described in the literature. Different soft sensor methodologies, such as Genetic Algorithm Partial Least Squares[34] (GAPLS) and Genetic Algorithm based Wavelength Selection[46] (GAWLS) are used along different data splits to assess the AD of different models for a flour and protein content data set.

Knowing that genetic algorithms are non-deterministic, each run may result in a different regression model. This information, if used correctly, can lead to ensemble prediction, which can be used for fault detection. Assessing AD acts as a bridge connecting data splitting and soft sensor performance with anomaly detection, by interpreting similar results from two distinct perspectives.

1.6.3 Strategy Summary

Four case studies were defined for performance comparison. To discuss AD features associated to fault detection, a flour and protein content data was considered. The remaining case studies are related directly to the GTM and Graph Theory combined methodology. A simulation data set with different types of anomalies was created, analyzing single and multiple anomaly scenarios, so to explore the potential of the proposed methodology aforementioned. Secondly, the Tennessee Eastman Process (TEP)[47], a virtual realistic data set, was considered for validation of the methodology, enhancing discussion and understanding. The proposed method was compared against unsupervised PCA, DPCA, GTM and Graph Theory independent approaches and supervised PCA, DPCA and GTM.

Chapter 2 explores all aspects related to the proposed GTM and Graph Theory strategy. Initially, a review on dimensionality reduction, graph theory and process monitoring is presented, describing all fault identification and diagnosis methods considered for comparison in this work, followed by the proposed approach described in detail. Finally, several results discussing the impact of different methodologies on anomaly detection and diagnosis are presented.

Chapter 3 discusses applicability domain and fault detection features. Similarly to Chapter 2, review, proposed strategy and results are presented. Chapter 4 presents final remarks and future work.

Chapter 2

GTM & Graph Theory Combined Approach

Throughout this work, rather different topics will be considered for discussion. Even though data visualization is the key element involving the proposed GTM and Graph Theory strategy, many other basic, but nonetheless important, topics should be explored. This section aims to present, thoroughly, relevant literature on all different features somewhat explored.

Initially, one has to consider how to handle data properly. This starts essentially on how to be able to extract only relevant information from a given data set. In order to do so, bypassing redundant and unnecessary features is an important step. This leads to a review on dimensionality reduction, aiming to clarify some of the main criteria taken into account when developing the methodology proposed in this work.

Handling, however, also involves knowing how to present and use this information in a meaningful way, according to the application. Graph theory is the great responsible for it, where not only data can be brought together in distinct clusters, but characteristics of each cluster gives further insight on the nature of different anomalies by evaluating what set each cluster apart and how they behave independently. Sections 2.1 to 2.3 describe thoroughly a review on all topics aforementioned. Section 2.4 presents the details regarding the combined approach proposed. Sections 2.5 and 2.6 present the main results of this chapter, exploring simulation and TEP data sets.

2.1 Dimensionality Reduction

2.1.1 Principal Component Analysis

Visualization of the relationship between distinct variables can be rather complex, especially if the system possesses many variables or is nonlinear. PCA is the most straightforward linear approach known, relying on variables being converted into linearly uncorrelated variables called principal components (PC), through an orthogonal transformation[18]. The basic logic behind PCA can be seen in Equation 2.1.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \tag{2.1}$$

where **X** is the original data set matrix, **T** is the score matrix, **P** is the loading matrix and **E** is the residual matrix. **P** establishes the relation between **X** and **T**, resulting in the projection of **X** values onto the transformed space **T**, where the PCs are its column vectors.

By finding the directions where data has maximum variance, it is possible to extract relevant information about the data set. Each PC contributes to the original data contained in **X** proportionally to the eigenvalues of column vectors in the covariance matrix. Such effect can also be expressed by the data variation of each principal component as described in Equation 2.2, assuming that centering and scaling are performed in advance.

$$C_{t_i} = \frac{\sigma^2(\mathbf{t_i})}{M} \tag{2.2}$$

where C_{t_i} is the component contribution for PC $\mathbf{t_i}$ and M is the number of input variables. The main goal is to select only those PCs that contain relevant information, excluding the rest. The heuristics considered in this work keeps only those components whose accumulated component contribution is just below 99%.

PCA is recognized as one of the main techniques for dimensionality reduction, with all sorts of applications in different fields, such as forensics[48], metabolic engineering[49] and cardiology[50], for instance. As for chemical engineering applications, process control[51] and process monitoring[52, 53] are a constant source of developments. This tool has been explored for so long that all sorts of derivations from the original PCA were developed. In order to better deal with time-series, for example, Recursive PCA[54] was created. For monitoring applications, once enough features are available, PCA sub-blocks could be obtained to adapt to different changes in process, called distributed PCA[21].

Since one of its main limitations is its inherent linear nature, which limits its application for more complex, nonlinear systems, other techniques were developed to cope with that, such as Kernel PCA[55], one of the most popular PCA extensions. In this work, we are focused on the original PCA, DPCA[56] and Kernel PCA, to be explained in details in the next subsections. This extension elegantly inserts artificially delayed variables to the pool of existing features, so to represent in a simple way the time-series aspect of a system.

2.1.2 Dynamic Principal Component Analysis

DPCA extends the regular PCA concept by introducing dynamics to better understand and represent nonlinear time series processes. The methodology itself is remarkably simple. Time shifted variables are added as extra variables, establishing a relation between current and past samples[57]. Equation 2.3 shows how to represent this new data set.

$$\mathbf{X_{Dyn}} = [\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_d}] = \begin{bmatrix} \mathbf{x_{d+1}} & \mathbf{x_d} & \cdots & \mathbf{x_1} \\ \mathbf{x_{d+2}} & \mathbf{x_{d+1}} & \cdots & \mathbf{x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_N} & \mathbf{x_{N-1}} & \cdots & \mathbf{x_{N-d}} \end{bmatrix} \tag{2.3}$$

where $\mathbf{X_i}$ is the original data set being delayed, N is the total number of samples and d is the sample delay. $\mathbf{x_n}$ is a row vector with all variables for the nth sample. DPCA has fundamentally the same approach as PCA, but with extra time shifted vectors. Thus, all analysis related to PCA, such as determining the optimal number of principal components, apply to DPCA as well.

By adding dynamics to the analysis, it is possible to achieve better discrimination within samples that, despite having similar variable values, belong to distinct stable and transient states. As for how far should variables should be delayed, most delays involves one or two time steps at most, related to the complexity of the system[57].

2.1.3 Kernel Principal Component Analysis

Kernel Principal Component Analysis (KPCA) can efficiently compute principal components in high-dimensional feature spaces, by relying on integral operators and nonlinear kernel functions[23]. The concept behind KPCA is quite intuitive, where linearly inseparable data is projected onto a new feature space, resulting in better discrimination. The mapping of a sample $\mathbf{x_i}$ can be written as $\mathbf{x_i} \rightarrow \phi(\mathbf{x_i})$, where ϕ is called kernel function.

Instead on applying PCA on the original data, a kernel matrix $\mathbf{K}$ is used analogously, where each element $k(\mathbf{x_i}, \mathbf{x_j})$ of this matrix is defined by the dot products shown in Equation 2.4

$$k(\mathbf{x_i}, \mathbf{x_j}) = \langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle \tag{2.4}$$

Kernels can map nonlinear data in different ways. Sigmoidal, Polynomial and Gaussian kernels[58] are the most common ones. In this work however, Gaussian kernel is used for evaluation, according to Equation 2.5

$$k(\mathbf{x_i}, \mathbf{x_j}) = e^{-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{c}} \tag{2.5}$$

where c is a width parameter regulating the Gaussian coverage. This parameter is defined based on the average minimum distance between samples in the original space. **K** is a square matrix, meaning that the number of features (and PCs) in the kernel space matches the number of samples available. Principal components are extracted based on their eigenvalue contribution. Analogously to PCA, relevant PCs are selected by keeping those components whose accumulated component contribution is just below 99%, as shown in Equation 2.6, where λ_i^{eig} is the eigenvalue for PC $\mathbf{t_i^{eig}}$.

$$C_{t_i} = \frac{\lambda_i^{eig}}{\sum_{i=1}^{N} \lambda_i^{eig}} \tag{2.6}$$

2.1.4 Generative Topographic Mapping

GTM is a widely used technique applied for visualization of data with several variables. It consists of a probabilistic non-linear approach, where a low-dimensional latent variable **z** is represented in a 2D space, so to approximate original data **x** as a high-dimensional manifold on the original data space. This manifold is modeled by a Gaussian function. Acting as a bridge between spaces, an intermediary layer of radial basis functions (RBFs), also Gaussian, is created[59]. RBFs are embedded in a mapping function $y(\mathbf{z};\mathbf{W})$, which defines the non-Euclidean manifold and connects both spaces. Figure 2.1 shows the schematic representation behind GTM.

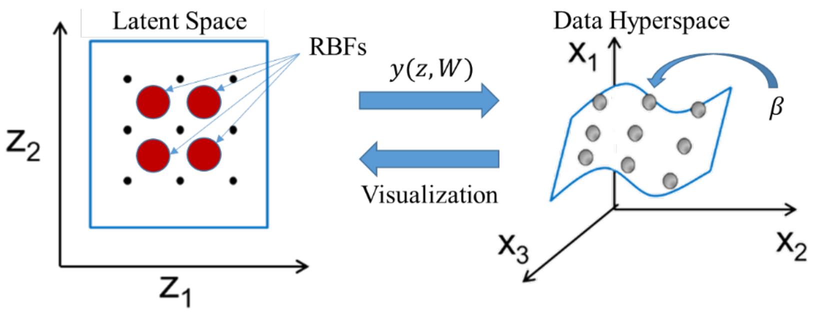

Fig. 2.1 GTM overall concept representation.

GTM structure

The main goal of GTM is to find a representation for the distribution $p(\mathbf{x})$ of data in a D-dimensional space $\mathbf{x} = (x_1, \ldots, x_D)$ associated to a number L of latent variables $\mathbf{z} = (z_1, \ldots, z_L)$. In order to achieve that, a function $y(\mathbf{z};\mathbf{W})$ is devised, mapping points $\mathbf{z}$ in the latent space into the equivalent $y(\mathbf{z};\mathbf{W})$. The transformation $y(\mathbf{z};\mathbf{W})$ maps the latent variable space into and L non-Euclidean manifold S embedded within the data space. $\mathbf{W}$ is a parameter matrix that governs the mapping from $\mathbf{z}$ to $\mathbf{x}$.

The distribution of $\mathbf{x}$ is chosen, for a given $\mathbf{z}$ and $\mathbf{W}$, to be a radially symmetric Gaussian centered on $y(\mathbf{z};\mathbf{W})$ having variance β^{-1}, as shown in Equation 2.7.

$$p(\mathbf{x}|\mathbf{z},\mathbf{W},\beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} e^{-\frac{\beta}{2}\|y(\mathbf{z};\mathbf{W})-\mathbf{x}\|^2} \tag{2.7}$$

The distribution in $\mathbf{x}$-space is then obtained by integration over the $\mathbf{z}$-distribution, assuming a known value for $\mathbf{W}$, according to Equation 2.8.

$$p(\mathbf{x},\mathbf{W},\beta) = \int p(\mathbf{x}|\mathbf{z},\mathbf{W},\beta)p(\mathbf{z}) \tag{2.8}$$

where $p(\mathbf{z})$ is the prior distribution of $\mathbf{z}$. Once a data set of N data points $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_N})$ is given, the unknown parameters $\mathbf{W}$ and β can be optimized, using maximum likelihood. It is more convenient, though, to maximize log likelihood, as presented in Equation 2.9.

$$\mathscr{L}(\mathbf{W},\beta) = ln\prod_{i=1}^{N} p(\mathbf{x_n},\mathbf{W},\beta) \tag{2.9}$$

One problem with this representation, however, is that despite specifying $p(\mathbf{z})$ and the functional form of $y(\mathbf{z};\mathbf{W})$, the integral specified in Equation 5 is usually analytically intractable. To circumvent this issue, $y(\mathbf{z};\mathbf{W})$ is chosen to be a linear function of $\mathbf{W}$ and $p(\mathbf{z})$ has to be defined accordingly. One option is to define $p(\mathbf{z})$ as Gaussian, then the integral becomes a convolution of two Gaussians. In this case, however, the model is closely related to PCA, where the maximum likelihood solution for $\mathbf{W}$ columns leads to scaled principal eigenvectors. In order to expand this formalism to nonlinear $y(\mathbf{z};\mathbf{W})$ functions, $p(\mathbf{z})$ has to be defined in a specific form, as shown is Equation 2.10.

$$p(\mathbf{z}) = \frac{1}{G}\sum_{i=1}^{G} \delta(\mathbf{z}-\mathbf{z_i}) \tag{2.10}$$

where G is the number of nodes in latent space assuming a regular grid. $p(\mathbf{z})$ is given by a sum of delta functions centered on nodes in a latent space grid. This implies that probability distribution is local in each point of the lattice and not continuously distributed along the latent space. The **x**-distribution function now takes a different form from Equation 2.8, as presented in Equation 2.11.

$$p(\mathbf{x}, \mathbf{W}, \beta) = \frac{1}{G} \sum_{i=1}^{G} p(\mathbf{x}|\mathbf{z_i}, \mathbf{W}, \beta) \tag{2.11}$$

and the log likelihood function is now given by Equation 2.12

$$\mathscr{L}(\mathbf{W}, \beta) = ln \left\{ \frac{1}{G} \sum_{i=1}^{G} p(\mathbf{x}|\mathbf{z_i}, \mathbf{W}, \beta) \right\} \tag{2.12}$$

Expectation-Maximization Algorithm

This structure can now be optimized for $\mathbf{W}$ and β, once $y(\mathbf{z};\mathbf{W})$ is defined. Knowing that the model developed consists of a mixture distribution, the Expectation-Maximization (EM) Algorithm might be the most suited for optimization[60]. This algorithm relies on a suitable choice of $y(\mathbf{z};\mathbf{W})$, such as a generalized linear regression model as described in Equation 2.13.

$$y(\mathbf{z};\mathbf{W}) = \mathbf{W}\phi(\mathbf{z}) \tag{2.13}$$

where $\phi(\mathbf{z})$ consists of B fixed basis functions $\phi_i(\mathbf{z})$, and $\mathbf{W}$ is a parameter matrix D x B relating these functions with the non-Euclidean manifold S. For a large class of basis functions, Radial Basis Functions (RBF) are universal approximators[61]. These structures, particularly the Gaussian RBFs, are interesting, due to their fast training. GTM training using multi-level programming (MLP), for example, is prohibitive[62].

Once the basis function structure is defined, the optimization can be executed. In the expectation step, current parameters $\mathbf{W_{old}}$ and β_{old} are used to evaluate the posterior probabilities, also called responsibilities, of each Gaussian component i for every data point $\mathbf{x_i}$ using Bayes' theorem, as shown in Equation 2.14.

$$r_{in}(\mathbf{W_{old}}, \beta_{old}) = p(\mathbf{z_i}|\mathbf{x_n}, \mathbf{W_{old}}, \beta_{old}) = \frac{p(\mathbf{x_n}|\mathbf{z_i}, \mathbf{W_{old}}, \beta_{old})}{\sum_{i=1}^{G} p(\mathbf{x_n}|\mathbf{z_i}, \mathbf{W_{old}}, \beta_{old})} \tag{2.14}$$

This leads to the expectation of the log likelihood data presented in Equation 2.15

$$\langle \mathscr{L}(\mathbf{W_{old}}, \beta_{old}) \rangle = \sum_{n=1}^{N} \sum_{i=1}^{G} r_{in}(\mathbf{W_{old}}, \beta_{old}) ln\{p(\mathbf{x_n}|\mathbf{z_i}, \mathbf{W}, \beta)\} \tag{2.15}$$

$\mathbf{W_{new}}$ and β_{new} can then be obtained on the maximization step, by maximizing Equation 2.15 with respect to both parameters independently, as shown in Equations 2.16 and 2.17.

$$\sum_{n=1}^{N} \sum_{i=1}^{G} r_{in}(\mathbf{W_{old}}, \beta_{old}) \{\mathbf{W_{new}} \phi(\mathbf{z_i}) - \mathbf{x_n}\} \phi^T(\mathbf{z}) = 0 \tag{2.16}$$

$$\frac{1}{\beta_{new}} = \sum_{n=1}^{N} \sum_{i=1}^{G} r_{in}(\mathbf{W_{old}}, \beta_{old}) ||\mathbf{W_{new}} \phi(\mathbf{z_i}) - \mathbf{x_n}||^2 \tag{2.17}$$

This cycle of expectation and maximization is repeated until the objective function reaches a maximum, according to a satisfactory convergence.

Data Visualization and Latent Probability Distribution

Once the map is trained, it is possible to determine for each sample the likelihood of it belonging to each node in the latent grid, establishing a PD profile. The profile comes from the responsibility matrix obtained from the optimization procedure aforementioned, as suggested by equation 2.18.

$$\mathbf{R} = [\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_N}] = \begin{bmatrix} \mathbf{r_{11}} & \mathbf{r_{12}} & \cdots & \mathbf{r_{1N}} \\ \mathbf{r_{21}} & \mathbf{r_{22}} & \cdots & \mathbf{r_{2N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r_{G1}} & \mathbf{r_{G2}} & \cdots & \mathbf{r_{GN}} \end{bmatrix} \tag{2.18}$$

Such profiles can be represented as individual heat maps or as one plot for all data, as Figure 2.2 suggests. PD profiles are unique considering that the variables in hyperspace have a different combination of values for each sample, which, for this work, allows the similarity assessment of all samples on the same basis. The overall structure of one of the proposed methodologies presented in this work relies on calculating similarity and GTM sophisticatedly fulfills this requirement as well.

GTM data visualization can also be performed by collapsing the PD profiles into mean and mode plot for all samples. Those plots give extra information on how samples are distributed along the map, since each datum can be represented as a dot, where visualization of distinct clusters in the data set might be more apparent. Figure 2.3 shows a comparison between PD heat map and mean/mode GTM plots.

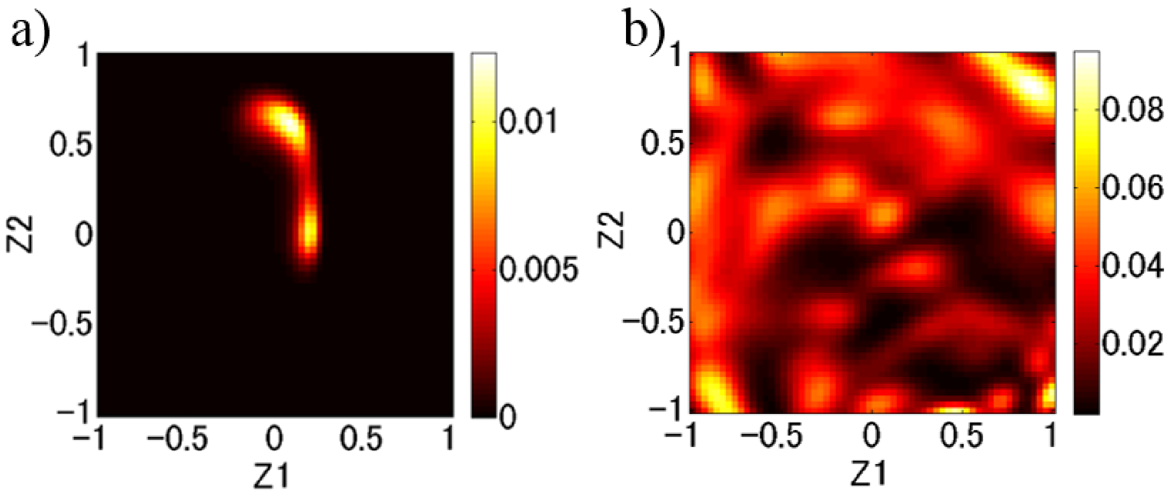

Fig. 2.2 GTM PD heat map for a) one sample and b) a data set.

In addition to the representation of data in the GTM map itself, it is important to notice how data is dealt within the optimization algorithm, since this impacts greatly how maps are trained. GTM considers all samples to be independent, identically distributed vectors (i.i.d.), which implies that dynamic information is not being considered. For monitoring applications, not using dynamic information is, at most, a waste of valuable information in one's data set. From this premise, some techniques were developed over the years trying to establish the connection between samples over time. GTM Through Time (GTMTT)[63], for instance, was developed using the original GTM algorithm and incorporating it as the emission density in a hidden Markov model. This leads to GTM training at every instant, combining all trained maps at the end of analyzing all samples. Despite the interesting approach, though, training GTM maps at each instant is computationally prohibitive, which constrains its application severely.

A faster and far simpler approach is to consider time-delayed variables, following the same procedure described in section 2.1.2. Analogous to DPCA, therefore, extra variables are created, allowing dynamic information to be incorporated to the system to some degree.

On GTM Hyperparameters

One aspect of GTM that deserves special attention is how to set the hyperparameters, which are structural parameters defined previously to optimizing the parameters $\mathbf{W}$ and β mentioned earlier. GTM relies on the following 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 is usually determined via exhaustive search, using cross-validation to look for the minimization of reconstruction error, i.e. distance from the manifold, once data is

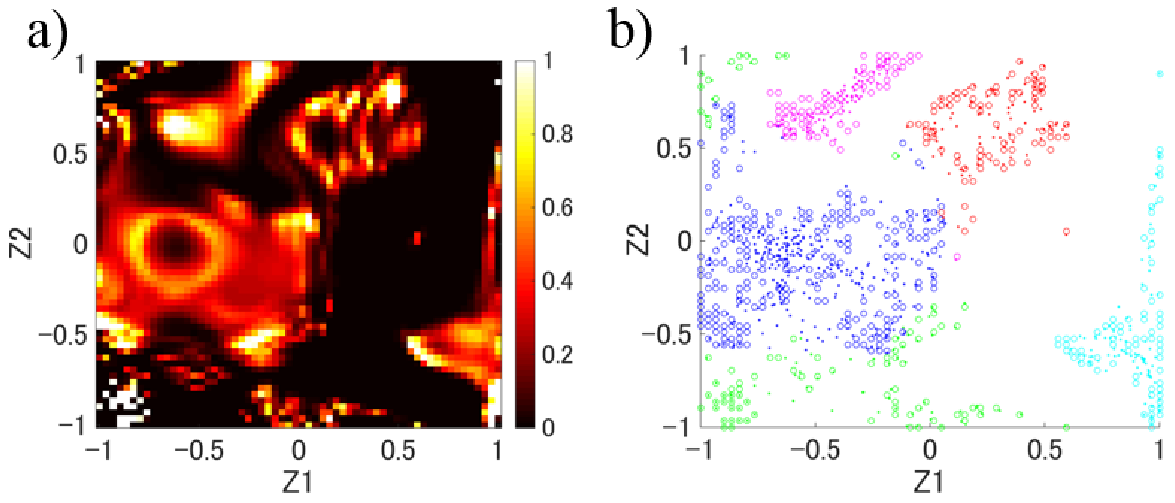

Fig. 2.3 a) GTM PD heat map and b) mean/mode plot. Different colors in the mean/mode plot represent different process states, where dots are collapsed PD means and circles are collapsed PD modes.

recreated into the original hyperdimensional space. Root Mean Squared Error (RMSE) is usually used as an index for such assessment, as described in Equation 2.19.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\|x_{i,j} - x_{i,j}^{rmp}\|^2}{NM}} \tag{2.19}$$

where N is the number of samples and M is the number of variables. $x_{i,j}$ is the original i^{th} sample value for the j^{th} variable and $x_{i,j}^{rmp}$ is the respective remapped value. Regular RMSE, however, does not take into account certain factors, such as the smoothness of the map, which allied to poor choosing of hyperparameters might lead to overfitting, a serious problem in GTM.

As pointed out by several works in the literature[64–66], GTM overfitting is often overlooked since most applications do not consider new data being incorporated to the map. For monitoring applications, however, the idea of new online data coming to a map is particularly interesting, so to assess data in real time. Overfitting, thus, results in great representation of training data, but poor evaluation of new samples coming to the map.

In order to cope with this methodology, one option is to use GTM extensions, which, allegedly, try to deal with this limitation. Variational Bayesian GTM[67] is the most prominent methodology, which replaces the regularization hyperparameter present for a full Bayesian treatment to a Gaussian process based variation of the optimization model. Such extension, however, is far too time consuming and computationally heavy to be used indiscriminately.

The key element is the regularization mechanism, since it is directly related to the smoothness of the map, where higher maps lead to smoother maps, but with generally greater reconstruction error. There are good practices embraced by the community using GTM, such as keeping this hyperparameter always above a threshold, which evidently is to be tuned based on the experience of the user with different data sets. Other developments can be allied to it, especially when it comes to the criterion used for evaluation. Root Mean Squared Error of Midpoint (RMSEM) tackles this issue[68], 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 concentrated sample's PD. RMSEM is calculated according to Equation 2.20.

$$RMSEM = \sqrt{\frac{\sum_{i=1}^{L}\sum_{j=1}^{M}\|x_{i,j}^{mid} - x_{i,j}^{mid,rmp}\|^2}{LM}} \tag{2.20}$$

where L is the number of midpoints selected, $x_{i,j}^{mid}$ is the i^{th} midpoint value for the j^{th} variable and $x_{i,j}^{mid,rmp}$ is the respective remapped point. Midpoints are sampled randomly from all possible combinations of training data, usually in a greater number (five-fold) than the original data set.

2.2 Graph Theory

Graphs are symbolic representations of networks that model pairwise relations between objects[45]. In chemical engineering, their application is mostly close to chemistry applications, such as drug design[69, 70], thermodynamics[71] and reaction pathways[72].

The main focus here, nonetheless, is process monitoring. There is little to no application of graph theory in chemical engineering for this purpose. The current research usually involves transforming the process itself in nodes[73] or using process variables as nodes for the development of a network[74]. Exploring the potential of samples as the main elements of a network, as proposed in this work, is new[75], leading to future insights on both fault detection and data visualization.

2.2.1 Basic Structure

There are two basic elements for every graph: nodes and edges. The former represents observations (samples) and the latter indicates connections between those observations. Graphs can be undirected or directed, according to the nature of connection between nodes. For undirected graphs, edges do not establish any direction for the connection between two samples, *i.e.*, there is no starting node and ending node for any edge. Directed graphs, on the other hand, establish that edges have direction, indicating flow of information. Electrical grid networks, for example, are directed graphs showing the flow of electrical energy. Social media networks are also usually directed, showing who start the interaction and towards whom. Graphs are, by definition, undirected unless said otherwise. For chemical engineering applications, both types are possible. Associating graphs with process design can lead to directed graphs, where the connection between equipment shows the flow of energy, mass, etc. Process monitoring where samples are associated to networks, on the other hand, can be undirected.

For a given data set, 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 2.4 shows an example of such representation.

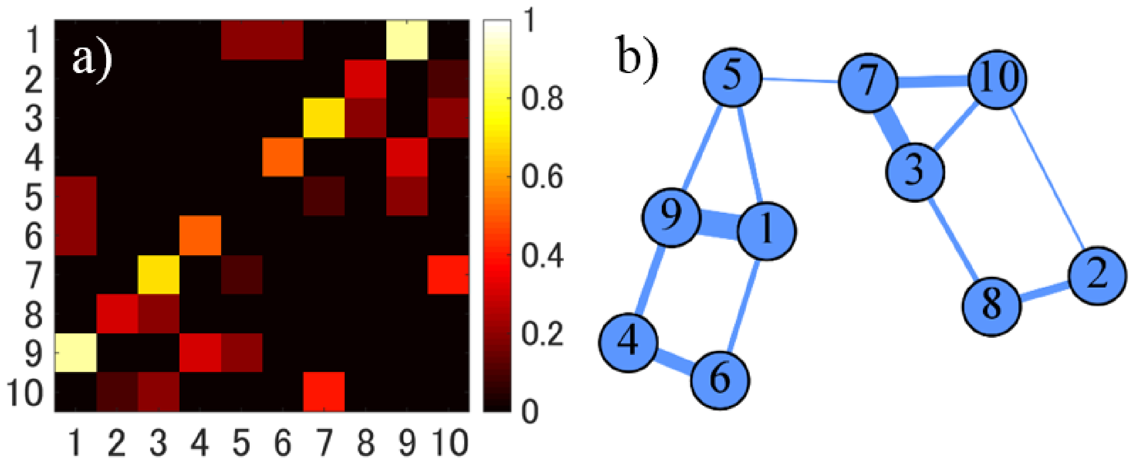

Fig. 2.4 Schematic representation of a a) weighted AM and b) its respective undirected graph.

All null values show that there is no connection between respective pair nodes. Values different from zero, on the other hand, 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.

2.2.2 Graph Analysis and Visualization

Visualization of graphs is associated to indexes (measures) that reveal different characteristics of the network. In addition, networks can be seen under different layouts, which also changes drastically the structure of the graph and how one would interpret the data. In the end, graphs are malleable, meaning that the same system can be seen through different filters, giving different insights about the data and the relationship between all samples.

If one considers each node as a sample in a chemical process belonging to a time-series data set, the behavior and characteristics of the network worth noting tend to follow those of social network analysis (SNA)[76]. SNA works with big data sets and try to investigate the relationship between individuals in social structures such as social media networks[77], disease networks[78], friendship networks[79], among so many others. SNA was the inspiration for using graphs relating chemical process measurements over time, to be depicted along this work.

Graph Layout

Adequately representing graphs in space is fundamental to visualize data in its entirety. There are several layout approaches available, each one with their particular characteristics. Force-Directed Graphs (FDG)[80] are more recurrent, since they try to draw graphs focused on aesthetic aspects. The idea is simple, using repulsion and attraction to the point where the system reaches a mechanical equilibrium, *i.e.*, all forces are balanced in the network. One of the most known FDG is the Fruchterman-Reingold (FR) layout[81], whose elegant representation is used to this day. Methodologies that are more efficient were developed over time, like Hu's layout[82], culminating in a robust, elegant and efficient approach, Force Atlas (FA)[83]. Figure 2.5 shows a comparison between three distinct layouts for the same data set.

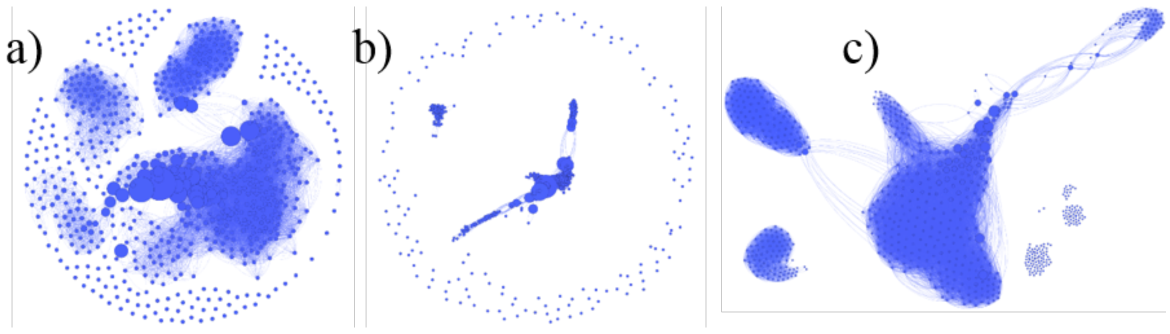

Fig. 2.5 a) FR, b) Hu and c) FA layouts for a simulation data set.

Both FR and Hu keep a spherical layout, which constrains the representation under one fixed area. FR tends, however, to fill all the area when compared to Hu, which can be positive or negative depending on the application. FA, on the other hand, is not restricted to that space and it shows more clearly the depiction of different clusters within the data set. Discussing whether one layout is better than the other, however, depends greatly on the data set and application of the analysis. For process monitoring purposes, where discriminating distinct clusters is important, however, FA style layout is used, since it gives more insight on this particular data aspect.

Centrality Measures

Once the layout is defined, there are still several analysis that can be performed considering the relationship between nodes and edges in the graph, through indexes called centrality measures[84]. All measures shown here can be applied for directed graphs, however considering the nature of the application proposed in this work, all measures are being applied for undirected graphs.

The most straightforward measure is the degree centrality, which reveals the total weight of edges towards one node[84]. It reveals in the network the most connected nodes overall. This measure has some limitations, however, since it does not specify whether this value come from several small edges or one big edge. In addition, it does not give any information related to the flow of information in the network.

Other centrality measures are complementary to degree analysis, revealing the most important nodes on the graph. Closeness[85], eigenvector[85] and betweenness[85] centrality are just a few to cite. For process monitoring applications, considering that discrimination among clusters is a determining factor, betweenness centrality might be the most interesting. It quantifies how many times a node acts as bridge along the shortest path between other nodes. It is interesting for determining important links between clusters where few nodes are associated between them. Figure 2.6 shows the distinction between degree and betweenness centrality for the same data set presented in Figure 2.6. Betweenness helps identifying the nodes that communicate with different clusters the most, indicating their transitional nature in the system. This analysis, however, is a minor aspect of the analysis for process monitoring. Being able to extract clusters in the system is far more interesting. The next sub-section will explore that aspect of graphs in more details.

2.2.3 Graph Clustering

Albeit all approaches consider graph features for discrimination, Graph Clustering (GC) can rely on rather different techniques, such as Spectral Partitioning (SP)[86], Girvan-Newman Algorithm (GNA)[87] and Louvain Community Finding (LCF)[88]. Each methodology has its merits and drawbacks, revealing the plurality of techniques available for adequate clustering in a network.

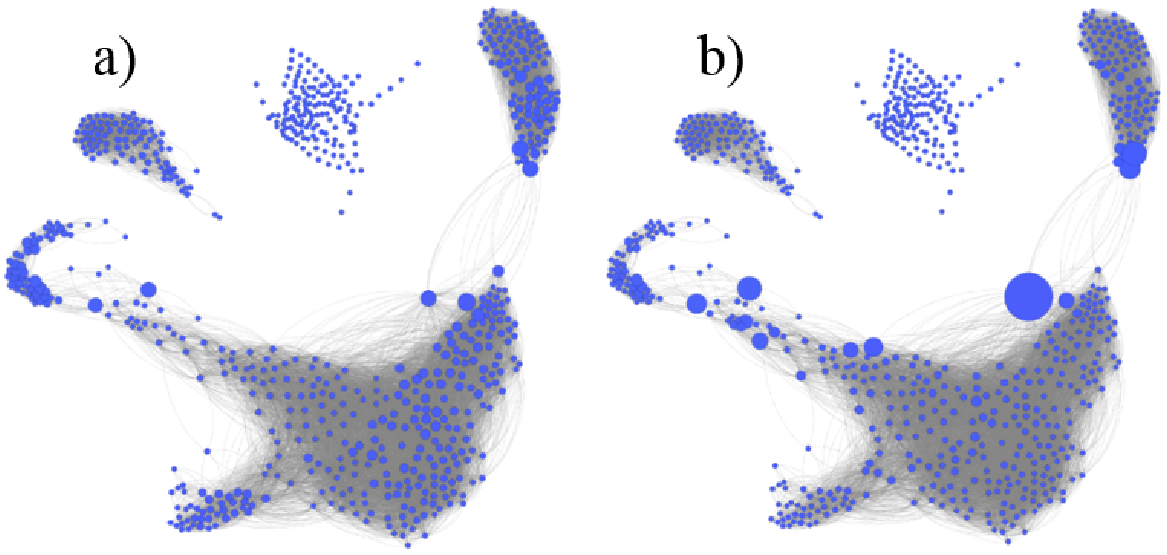

Fig. 2.6 a) Degree and b) betweenness graphs for a simulation data set.

Spectral Partitioning

SP is one of the simplest methods for GC[89]. It relies on the Laplacian matrix for connected graphs, as expressed by Equation 2.21.

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{2.21}$$

where **D** is the degree matrix and **A** is the AM. The degree matrix can easily be obtained by computing all node degrees from **A**. Once the Laplacian is obtained, the eigenvector and eigenvalues of **L** are determined. The eigenvector associated with the lowest second vector, called Fiedler Vector, is then used for bisection of the graph. All values above zero belong to one community, all values below zero to another.

The limitations are apparent. SP only bisects the graph, even though further splitting is possible assuming that partitioning is performed on the remaining sub-graphs available. If the user has no information on the optimal number of clusters, however, it is difficult to know when to stop bisecting. This limitations motivates the use of other criteria for discrimination, as described in the next GC methodology.

Girvan-Newman Algorithm

GNA finds clusters by removing edges from the original network progressively[90]. In order to do so, the algorithm focuses on edges the are in between communities by evaluating edge betweenness. Edges are gradually removed to the point where the initially connected graph is divided in two connected graphs. The algorithm can keep running indefinitely until all edges are removed. Betweenness has to be updated every time an edge is removed from the network, resulting in a computationally expensive algorithm.

This methodology characterizes hierarchical clustering, where a dendrogram is created keeping track of communities being split. Despite the different criterion for community detection, however, this approach still suffers from lack of termination criterion. The splitting of the community is more gradual and conceptually more interesting, but it gives little solution to a clear termination criterion for community detection.

Louvain Community Finding Algorithm

As discussed in the previous sub-section, SP and GNA has no termination criterion for optimal clustering and GNA relies on betweenness[45], a graph centrality measure for finding important hubs in the graph, which may not be available for a given graph or it is computationally too expensive to calculate.

In order to cope with these limitations, Louvain Community Finding (LCF)[88] presents itself as an algorithm with intriguing features, based on, generally speaking, evaluating the density of edges within a group via an index called modularity[91, 92]. LCF algorithm has two steps: local modularity optimization and graph update. Initially, a weighted graph of N nodes is created, where different clusters are assigned to each node, i.e., there are as many clusters as nodes. From this framework, a maximization of modularity is trailed, following the pseudo-algorithm below. A more detailed version of the algorithm can be found in the Appendix I.

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, only if the gain is positive. Otherwise, i stays in its original community. Figure 2.7 shows the schematic representation of step 2 for one node, when tested against three other communities.

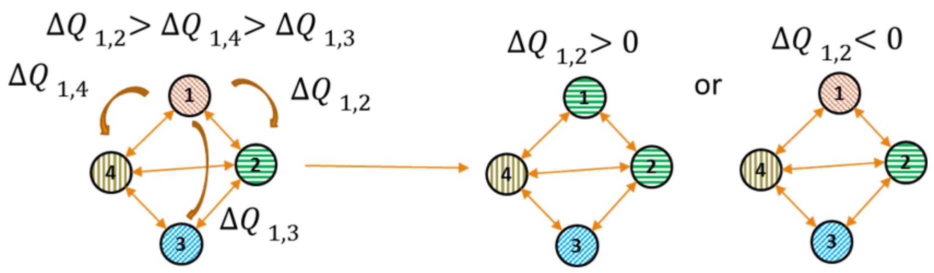

Fig. 2.7 Modularity gain test, where different background patterns indicate different communities.

3. Test modularity gain for all nodes in sequence, till no further improvement is encountered. Modularity gain is described in Equation 2.22.

$$\Delta Q_{i,j} = \frac{k_{i,j}}{2m} - \lambda \frac{k_i \Sigma_{tot}}{2m^2} \tag{2.22}$$

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, Σ_{tot} is the sum of edge weights incident to nodes in j and λ is called the resolution limit, regulating both terms of Equation 6. Lower λ results in fewer clusters, where higher λ results in more clusters. Once the algorithm stabilizes, the graph is updated by condensing all nodes belonging to a single community into a single node, keeping in mind that edges between nodes of the same community lead to self-loops. After the update, all steps above are repeated until no more modularity gain is achieved. Figure 2.8 and Figure 2.9 show graph and modularity evolution during LCF cycles for a trivial example.

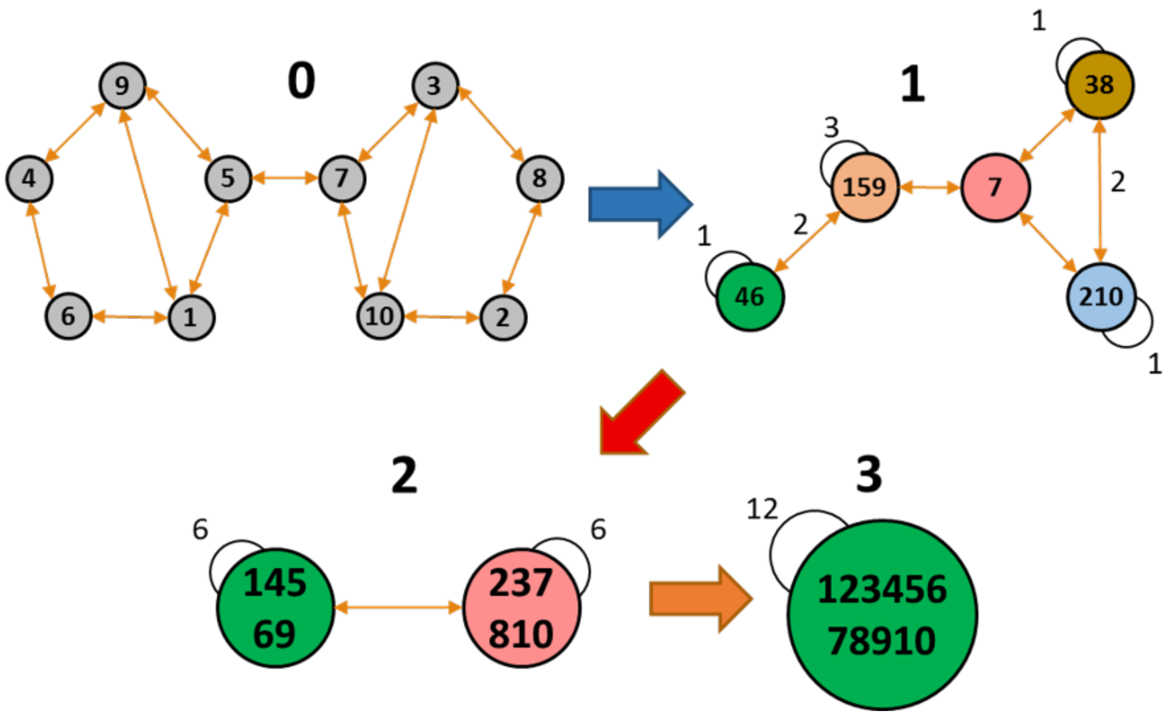

Fig. 2.8 Graph evolution according to LCF algorithm.

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

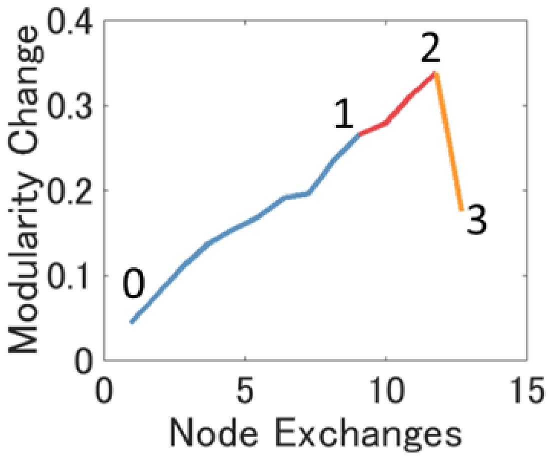

Fig. 2.9 Modularity evolution during LCF cycles.

in a modularity drop. LCF uses modularity in a similar way that Newman-Girvan algorithm, but it provides an intuitive algorithm with a clear termination criterion.

2.3 Process Monitoring

2.3.1 Principal Component Analysis, Dynamic PCA & Kernel PCA

PCA, DPCA and KPCA rely on two distinct indexes for process monitoring: T^2 and Q[33]. The former assesses input data variation and the latter, prediction residuals. Several articles in the literature have dealt with those indexes[93, 94], reassuring their application for the scenario previously described. Equations 2.23 and 2.24 present both indexes for PCA and DPCA. Those values are calculated for each sample, leading to a Q x T^2 plot used to indicate discrimination between data groups.

$$T_n^2 = \sum_{i=1}^{k} (\frac{t_{ni}}{\lambda_i})^2 \tag{2.23}$$

$$Q_n = \sum_{i=1}^{M} (x_{ni} - \hat{x}_{ni})^2 \tag{2.24}$$

where M is the number of input variables and k is the number of principal components selected. t_{ni} is the score component for the n^{th} sample and i^{th} t-score variable and std_i is the estimated standard deviation for the respective t-score. $\hat{x}$ is the estimated input given k principal components for data reconstruction.

KPCA, on the other hand, while maintaining the same conceptual background, relies on a different set of equations, since its assessment is in the nonlinear kernel space. Prediction residuals and input data variation can be all associated to the score **t**, as presented in Equations 2.25 and 2.26

$$T_n^2 = [t_{n1}^{eig}, t_{n2}^{eig}, \ldots, t_{nk}^{eig}]\Lambda^{-1}[t_{n1}^{eig}, t_{n2}^{eig}, \ldots, t_{nk}^{eig}]^T \tag{2.25}$$

$$Q_n = \sum_{i=1}^{N} t_{ni}^{eig^2} - \sum_{i=1}^{k} t_{ni}^{eig^2} \tag{2.26}$$

For unsupervised approaches, once data belongs to different states, one could assume that T^2 and Q would also be on different ranges, or that at least one of the indexes would behave in such way. If samples are dissimilar enough, groups of data can be isolated.

Such dissimilarity would lead to data discrimination and, thus, process monitoring. While it is true that the usual approach is supervised, where the reduced model is trained only with data from a particular state, one can apply it to an unsupervised scenario, if there is a clear distinction between two or more data groups. For supervised approaches, a threshold for both indexes is defined as 99% of the maximum value in the normal data. Values exceeding at least one of the thresholds are detected as outliers.

In the case where a clear discrimination is not possible, another option is to cluster data based on the reduced PCA subspace, by using, for example, k-means clustering on the main PC subspace[95], where here k indicates the predetermined number of clusters to obtained. It is true that one need to know beforehand the number of clusters required for discrimination, however depending on the characteristics of the system, this parameter could be inferred with reasonable certainty. Furthermore, not only the PCA subspace, but also the T^2 and Q plot itself can be used for discrimination, using their values as position references.

2.3.2 Generative Topographic Mapping

The main aspect behind GTM is that its latent space approximates the original, hyperdimensional variable space. Once all issues related to possible overfitting are considered, these trained maps can be used for process monitoring, once RE, remapping error is calculated. Once data reduced to a latent framework is re-plotted into the original space, RE can be calculated[96], according to Equation 2.27.

$$RE_n = \sqrt{\sum_{i=1}^{M} (X_{ni} - X_{ni}^{GTM})^2} \tag{2.27}$$

where x_{ni} and $x_{GTM_{ni}}$ are the original and reconstructed, respectively, i^{th} input for the n^{th} sample. Analogous to PCA, both supervised and unsupervised approaches are possible, where the former only uses the previously defined normal group for map training and the latter uses all data for training. For the unsupervised approach, finding at least two groups with clearly distinct RE ranges is desirable, yet unlikely. For supervised approach, a maximum threshold considering 99% of the maximum RE in normal data is used, where RE exceeding this threshold are said to be anomalous.

Considering the GTM structure, however, one can argue on the effectiveness of an unsupervised approach for this assessment. Knowing that GTM fits data on the latent space aiming for a non-Euclidean manifold in the original hyperdimensional space, there is no discrimination between data in different states. All samples are being fitted to the latent map. From this premise, the remapping error should be equivalent for all samples, anomalous or not in the system.

While it is true that only a supervised approach would be adequate for this evaluation, once normal samples can be extracted from the original data set, the map could be trained with this extracted sub set, leading to discrimination between anomalous and normal samples.

Besides Remapping Error, similarity between probability distributions can be used as an index. The same matrix used for the network generation can be used for assessment. From an unsupervised perspective, little knowledge can be obtained about the process, since there is no reference. If normal data is known, however, it is possible to see whether external samples are at least 99% similar to at least one of the samples in the normal data pool. Assuming that all normal samples are part of the same state, if any external sample is similar to any sample in the pool, it can be considered normal as well. This idea leads to an unique threshold for each sample, since maximum similarity for each query sample against the normal data pool is different.

2.3.3 Graph Theory

Once data is represented as a network and GC is performed, different clusters can be isolated in the system's graph. From the LCF procedure, a graph structure is defined with clusters representing different states in the chemical plant. By defining a criterion for isolating the normal group, biggest cluster for example, a clear distinction between outliers and normal samples can be established.

Given a similarity criteria that creates the network in the first place, new data can be added to it, where LCF can constantly re-evaluate the network and assess whether data is classified as normal or anomaly. From that point on, monitoring can be performed online, while, if necessary, one can visualize the system through the network being currently updated.

2.4 Proposed Methodology (GTM+GT)

The main methodology explored in this work involves two key elements: extraction of essential information and effective data clustering. This is achieved by combining GTM and Graph Theory[75]. Primarily, GTM reduces data to a 2D latent plot, removing redundant and irrelevant information from the original data set. Every sample in the latent space has a unique PD profile, which is used for similarity assessment, as represented schematically in Figure 2.10 for two responsibility vectors $\mathbf{r_1}$ and $\mathbf{r_2}$.

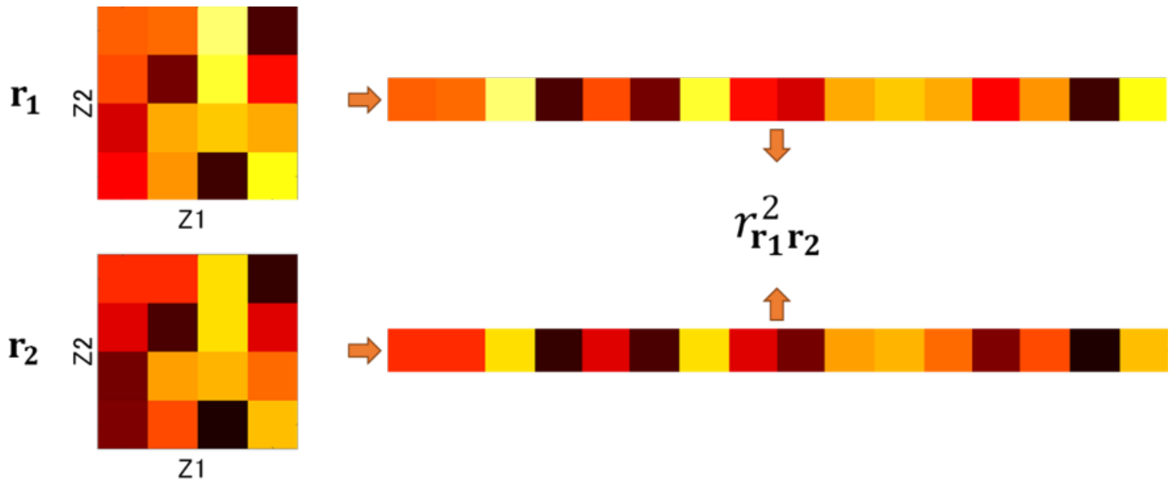

Fig. 2.10 Correlation assessment between two samples using the same GTM grid.

Each sample's PD can be expanded in a vector, which then is used for squared Pearson product-moment correlation coefficient (r^2) calculation. Each assessment between samples fills one element of the AM. Once all samples are cross-evaluated, AM construction is finished.

With the AM built, LCF can cluster data into groups with similar characteristics. For unsupervised fault identification, it is assumed that faults are a minority of the system and due to their faulty nature, their behavior is usually more erratic, *i.e.*, less stable. Normal operational data, on the other hand, 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 (NC). It is also important to notice that anomalous data might be detected as not one cluster, but several, representing different fault characteristics or different states within one fault development over time.

2.4.1 AM Construction

The benefit of GTM is to extract only relevant information for assessing similarity between data. How to calculate it properly, however, becomes the challenge. It was presented earlier that r^2 was used for calculation. The reason behind this assumption, however, was not clarified. PD profiles are analogous to images, where comparing two images relies on evaluating the discrepancies in pixels. From image processing, similarity can be defined according to structural discrepancies, also called Structural Similarity (SSIM)[97]. It relies on three distinct features: luminance, contrast and structure, described by the Equations 2.28 to 2.30[98], respectively, for two PD (responsibility) vectors $\mathbf{r_1}$ and $\mathbf{r_2}$.

$$l_{\mathbf{r_1},\mathbf{r_2}} = \frac{2\overline{\mathbf{r_1}}\,\overline{\mathbf{r_2}} + C_1}{\overline{\mathbf{r_1}}^2 + \overline{\mathbf{r_2}}^2 + C_1} \tag{2.28}$$

$$c_{\mathbf{r_1},\mathbf{r_2}} = \frac{2\sigma_{\mathbf{r_1}}\sigma_{\mathbf{r_2}} + C_2}{\sigma_{\mathbf{r_1}}^2 + \sigma_{\mathbf{r_2}}^2 + C_2} \tag{2.29}$$

$$s_{\mathbf{r_1},\mathbf{r_2}} = r_{\mathbf{r_1}\mathbf{r_2}}^2 = \frac{\sigma_{\mathbf{r_1}\mathbf{r_2}} + C_3}{\sigma_{\mathbf{r_1}}\sigma_{\mathbf{r_2}} + C_3} \tag{2.30}$$

where σ is standard deviation, $\overline{\mathbf{r_1}}$ and $\overline{\mathbf{r_2}}$ are the average values of $\mathbf{r_1}$ and $\mathbf{r_2}$ and C_i are arbitrary constants to avoid instability when the denominator is very close to zero. $\sigma_{r_1r_2}$ is shown in Equation 2.31.

$$\sigma_{\mathbf{r_1}\mathbf{r_2}} = \frac{1}{G-1}\sum_{i=1}^{G}(r_{1i} - \overline{\mathbf{r_1}})(r_{2i} - \overline{\mathbf{r_2}}) \tag{2.31}$$

remembering that G is the number of nodes in latent space assuming a regular grid. Luminance considers differences in the average PD value, contrast compares variance changes in PDs and structure calculates the correlation between PDs. When two images are compared to assess degradation, for example, all those three elements are important and equally relevant. For PD evaluation, however, luminance and contrast are far less important than structural comparison. Figure 2.11 shows the usual PD for two samples in a GTM map.

When images are compared, all pixels have important information and therefore all three indexes are relevant, trying to extract all tiny nuances in both images. GTM PDs are much more crude and limited in their representation, occupying specific regions of the map. Dissimilar samples will occupy different regions in the map, resulting in low similarity. Similar samples will overlap and be somewhat similar. This structural assessment depicted in Equation 2.28 is enough for our applications. It is important to notice that given some mathematical manipulation, Equation 2.28 corresponds directly to r^2.

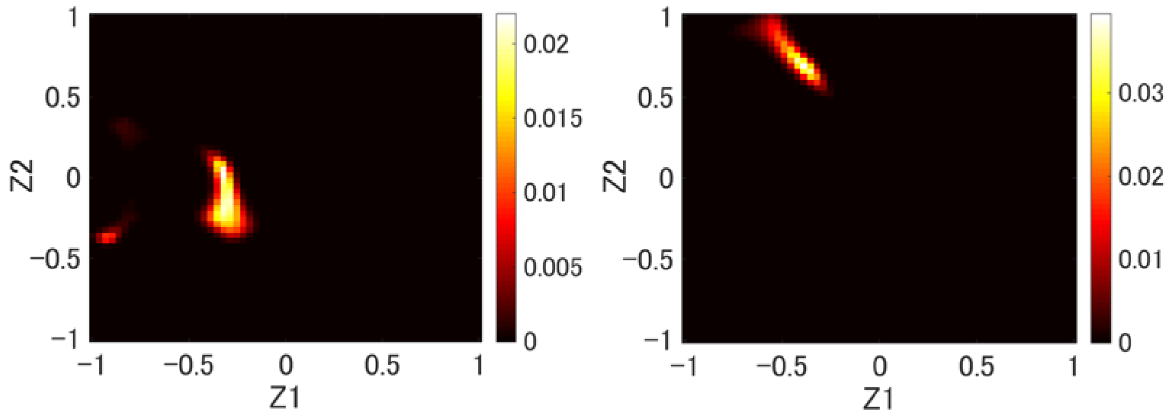

Fig. 2.11 GTM PD for two samples in a trained map.

Assessing similarity between all samples in the system, then, is being portrayed as simple r^2 calculation. Depending on the map size chosen for GTM, however, each PD vector can easily go to thousands of variables. Even for small map sizes such as 10 x 10, there are 100 features being compared between samples for similarity assessment. Knowing that samples have local PD, calculating solely correlation between samples is not enough, since it would result in very low similarity for any samples marginally different. In order to cope with that, good practices recommend local assessment, by creating a moving window, which slides point by point throughout both latent grids. Once all local values are calculated, an average similarity is calculated and integrated to **A**, the AM, as shown in Equation 2.32 for $\mathbf{r_1}$ and $\mathbf{r_2}$ presented previously.

$$a_{12} = \frac{1}{W}\sum_{j=1}^{W} r^2_{r_{1j}r_{2j}} \tag{2.32}$$

where $\mathbf{r_{1j}}$ and $\mathbf{r_{2j}}$ are local vectors and W is the number of local windows. Once similarity assessment is finished for all responsibility combinations, AM can be constructed. Finally, for any AM very low correlation values are recurrent ($< 10^{-5}$). In order to keep the structure of the network intact while discarding irrelevant correlation values, a low threshold (10^{-5}) is set to cut all similarity values below it. Determining a low threshold rather than a high threshold is more important, excluding correlation by chance and reducing considerably the number of unnecessary connections.

2.4.2 Normal Cluster Refinement

Once GTM map is trained, AM can be generated, resulting in a network representing the system. GC is then performed using LCF, isolating clusters with different characteristics.

The biggest stable one is defined as the normal cluster (NC), where the remaining ones are defined as anomalous clusters. According to this description, the NC is susceptible to anomaly contamination. Assuming a big gradient in similarity between different groups of samples, it is possible for faults that are somewhat similar with the normal data to be misjudged as belonging to this cluster.

Very dissimilar anomalies will be far from normality in the GTM map created, but slightly dissimilar anomalies might be close enough to normal data. From this premise, an iterative methodology using the GTM and Graph Theory cycle is concurrently proposed, where only the sub-data set belonging to the NC is used for GTM retraining, leading to sub-graphs and new clustering results. Once data with a smaller similarity range is chosen for retraining, it is more likely for marginally dissimilar samples to be excluded from the NC, increasing the likelihood of clearer final clusters. Figure 2.12 shows the entire flow of this procedure.

Once NC stops changing its size, *i.e.*, when no more samples are excluded from the cluster, the final NC is obtained. One important aspect to notice is that LCF's resolution parameter λ described in Equation 19 is updated every iteration, which regulates the importance of connections inside and outside the cluster. Higher λ means more clusters and lower λ results in fewer clusters. If the data set does not change in size, an initial adjustment is enough. If the system remains the same, but the data change size, the clustering results can be affected, pointing one downside of the modularity approach, called resolution limit[92, 99]. Its effect is usually felt on big data networks, where small communities are hard to detect unless subnetworks are extracted and λ is adjusted.

For the iterative methodology described, knowing that the amount of data decreases over iterations, the likelihood of one cluster splitting into smaller ones is big if the parameter is not adjusted. From this premise, λ is adjusted according to Equation 2.33.

$$\lambda_{i+1} = \frac{\lambda_i l_{i+1}}{l_i} \tag{2.33}$$

where l_i is the number of samples under GTM training at iteration i and λ_i is the respective resolution parameter. The initial value for λ depends on the data set involved, by evaluating how many clusters are being generated. There is a connection between the density of connections in the network, and the likelihood of fewer clusters being the optimal solution for LCF. When λ is too big, big clusters can be split in two or more smaller clusters, even though their connection is evident. When λ is too small, clusters with different characteristics connected by a few links can be misguidedly brought together. An intermediary value is recommend, so to avoid those issues.

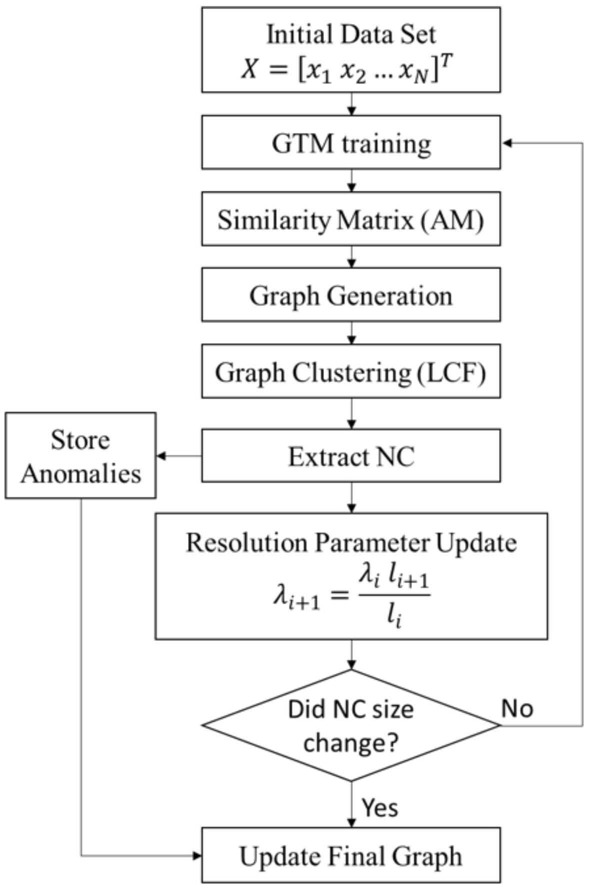

Fig. 2.12 Schematic representation of the proposed fault detection procedure, considering NC refinement.

Along with the update of the resolution parameter, there is another aspect that needs to be taken into account related to how to determine the hyperparameters. As mentioned previously, assessing similarity is related to the PDs profiles of each sample in the map. If samples are similar, there must be some overlap between them, so that a higher similarity value is obtained. Regardless of the criterion, cross-validation is used for determining the best combination of parameters. A grid search like structure is created, were all possible combinations of hyperparameters are tested, according to pre-determined values assigned to each hyperparameter. By using good practices and the right criterion, overfit can be avoided. Some issues, however, might still arise from a similarity assessment perspective. Depending on the range of those values, data's PD profiles may be too concentrated, to the point where even really similar data has low similarity. As the clusters get small, and fewer data are used for training, the likelihood of low similarity values is high, which compromises clustering, even with λ update. The number of RBFs and their width affect negatively the map. If the number of RBFs is far too great and their width is too small, even without overfitting obtaining meaningful AM can be difficult. Determining a reasonable number of RBFs, thus, is desirable.

In order to cope with that, Bayesian Information Criterion (BIC)[100] is used for determining the optimal number of RBFs. BIC is formally defined according to Equation 2.34

$$B = -2ln\hat{\mathscr{L}} + klnN \tag{2.34}$$

where $\hat{\mathscr{L}}$ is the maximized value of the likelihood function and k is the number of parameters needed for model generation. Each term of this equation takes into account different aspects of model selection. The likelihood term considers how good the model fit is. The second term penalizes models based on how complex the model is (number of parameters) and how many samples are involved. By minimizing B, the optimal model can be selected.

BIC is mainly used when the modeling application is descriptive, *i.e.*, when the goal is to create a model featuring the most meaningful factors influencing the outcome, based on an assessment of relative importance. For the work explored in this thesis, what is proposed is a fit based on Gaussian Mixture Models (GMM)[101], where the optimal model reflects the optimal k number of Gaussians required for model representation. Since GTM works with RBFs as Gaussian functions, this value is used as a guide for their determination. Knowing that the displacement of RBFs in the latent space only accepts values leading to integer square roots $(4, 16, 25, \ldots)$, the optimal value found by BIC is adjusted accordingly, always rounded up. The remaining GTM hyperparameters, RBF width and regularization, are determined by cross-validation.

2.5 Case Study - Simulation Data Set

Trying to include several topics related to anomaly detection through a data visualization centered methodology, the combined GTM and Graph Theory strategy was presented. Different aspects of fault detection such as multiplicity of anomalies and unsupervised approach philosophy are being explored using two case studies[75]. This section shows a simulation data set for single and multiple anomaly scenarios, exploring in a controlled environment the capabilities of the proposed methodology. Section 2.6 shows results associated to Tennessee Eastman Process, a virtual chemical plant whose structure mimics real plant behavior, in order to validate the proposed methodology and further enhance the discussion. Finally, section 2.7 depicts a real industrial case of an exhaust gas denitration process.

2.5.1 Single Anomaly Scenarios

In order to evaluate the potential of the combined GTM and Graph Theory approach for process monitoring, artificial data sets were created, with different goals in mind. Initially, single anomaly scenarios were generated, so to evaluate the basic premise of the proposed methodology, *i.e.*, using GTM for similarity assessment, generating the AM and creating the system graph. Aiming only to evaluate the potential of this procedure, the analysis is limited to the first discrimination cycle, meaning that normal data refinement is not explored at this step.

An artificial data consisting of four pseudo-random variables was created. Aiming to explore the most common types of anomalies encountered in chemical plants, six scenarios were proposed, according to what can be seen in Figure 2.13. Additive Outliers (AO) are spikes in the system, usually present when there are sensor malfunctions. Level Shift (LS) indicate sudden and sustained change in state, sign of sensor non-gauging or unexpected changes in set-point. Local Trend (LT) represents continuous data drift over time, which usually appears as disturbances in the system, such as steam leaking or raw material shortage. Transient Change (TC) occurs when a sudden change is dampened over time. Process control malfunctions or sudden bursts of pressure might trigger such behavior. Seasonal Change (SC) is oscillatory behavior, where it can appear as a disturbance, like temperature changes during day-night cycles, or as pure anomalies when controllers become poorly tuned. Finally, Variance Change (VC), which as the name suggests, is a result of change in data variation. Sticking valves are a classic example of such anomaly. For simplicity, all disturbances are assumed to be happening simultaneously on all variables. Even though it might not represent chemical plant faults in their entirety, faults can affect several variables in a short period.

For generation of the data sets, four different baselines $\mathbf{x_b} = (0.2, 0.4, 0.6, 0.8)$ were created, where both random Gaussian noise and Brownian motion were used for defining the overall structure of each variable $\mathbf{x_i}$, as presented in Equation 2.35.

$$\mathbf{x_i} = x_{b_i}\mathbf{j} + 0.005\mathbf{g} + 0.02\mathbf{b} \tag{2.35}$$

where x_{b_i} is the i^{th} baseline value, $\mathbf{j}$ is a vector of ones, $\mathbf{g}$ is a random Gaussian noise vector from 0 to 1 and $\mathbf{b}$ is a Brownian motion vector with variance of norm distribution equal to 1.

Using Equation 33, four pseudo-random variables of 1000 samples were created, where part of those samples were altered for each scenario to be the anomalies. Equations 2.36 shows the general structure of each fault described for a variable $\mathbf{x_i}$, except for VC described in Equation 2.37, keeping in mind that all faults happen simultaneously in all variables.

$$\mathbf{x_{out_j}} = \mathbf{x_i}(\mathbf{t_j}) + \mathbf{o_j} \tag{2.36}$$

$$\mathbf{x_{out_{VC}}} = \overline{\mathbf{x_i}(\mathbf{t_{VC}})} + \mathbf{o_{VC}} \tag{2.37}$$

where $\mathbf{x_{out_j}}$ is the j^{th} anomalous input vector, $\mathbf{t_j}$ is the j^{th} anomaly time vector, $\mathbf{o_j}$ is the defined anomaly itself for the j^{th} anomaly and $\overline{x_i(t_{VC})}$ is the faulty input average for VC. Table 2.1 shows all $\mathbf{t_j}$ and $\mathbf{o_j}$ parameters used for the six scenarios.

Table 2.1 Parameters for single anomaly scenarios generation.

	$\mathbf{t_j}$	$\mathbf{o_j}$
AO	$\mathbf{t_{AO}} = [200\ 400\ 600\ 800]$	$\mathbf{o_{AO}} = [0.5\ -0.5\ 0.2\ -0.2]$
LS	$\mathbf{t_{LS}} = [751\ 752\ \cdots\ 1000]$	$\mathbf{o_{LS}} = 0.15\mathbf{j}$
LT	$\mathbf{t_{LT}} = [601\ 602\ \cdots\ 1000]$	$\mathbf{o_{LT}} = 0.0025(\mathbf{t_{LT}} - 600)$
TC	$\mathbf{t_{TC}} = [501\ 502\ \cdots\ 1000]$	$\mathbf{o_{TC}} = 0.07e^{-\frac{6(\mathbf{t_{TC}}-500)}{500}}(\mathbf{t_{TC}} - 500)$
SC	$\mathbf{t_{SC}} = [501\ 502\ \cdots\ 1000]$	$\mathbf{o_{SC}} = 0.2sin\left[\frac{4\pi(\mathbf{t_{SC}}-\mathbf{500})}{500}\right]$
VC	$\mathbf{t_{VC}} = [501\ 502\ \cdots\ 750]$	$\mathbf{o_{VC}} = 7[\mathbf{x_i}\mathbf{t_{VC}} - \overline{\mathbf{x_i}\mathbf{t_{VC}}}]$

The dynamic information was added using time delayed variables. For these scenarios, delay d was defined as 1. Initial λ was defined as 1.

For unsupervised PCA, Q and T^2 plots are presented in Figures 2.14 and 2.15, respectively. The lack of discrimination between both normal and anomalous is clear, indicating poor monitoring performance. Except for AO and VC, data cannot easily be discriminated in distinct clusters. PCA's simplistic approach and inherent linear capabilities led to such results.

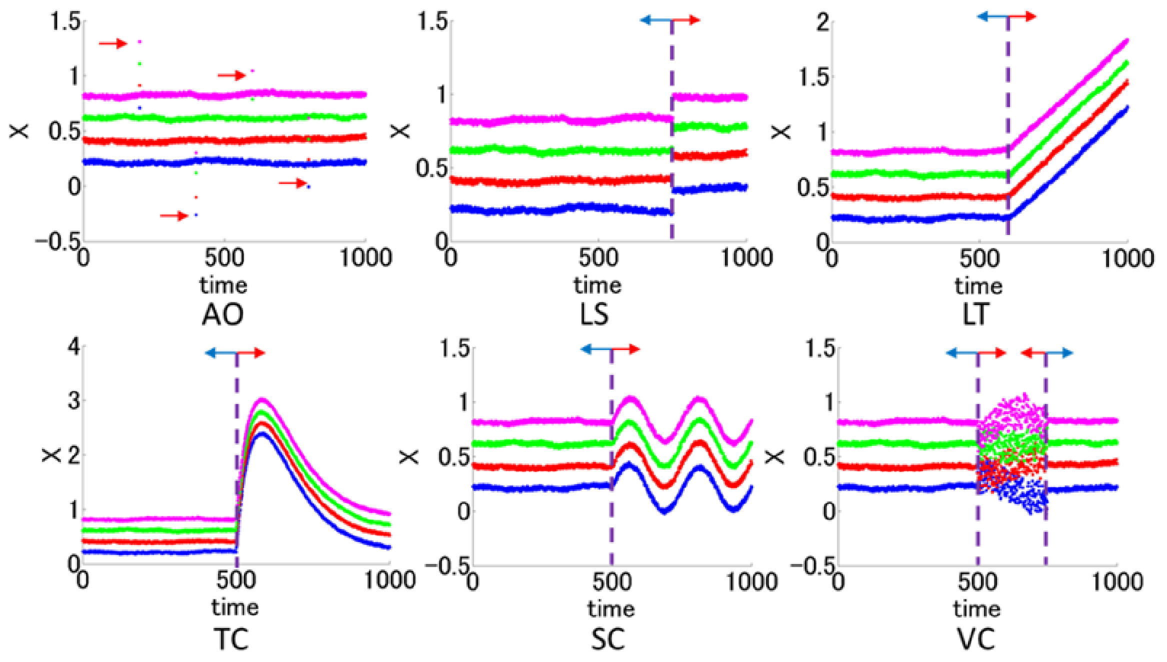

Fig. 2.13 Single anomaly simulation data set.

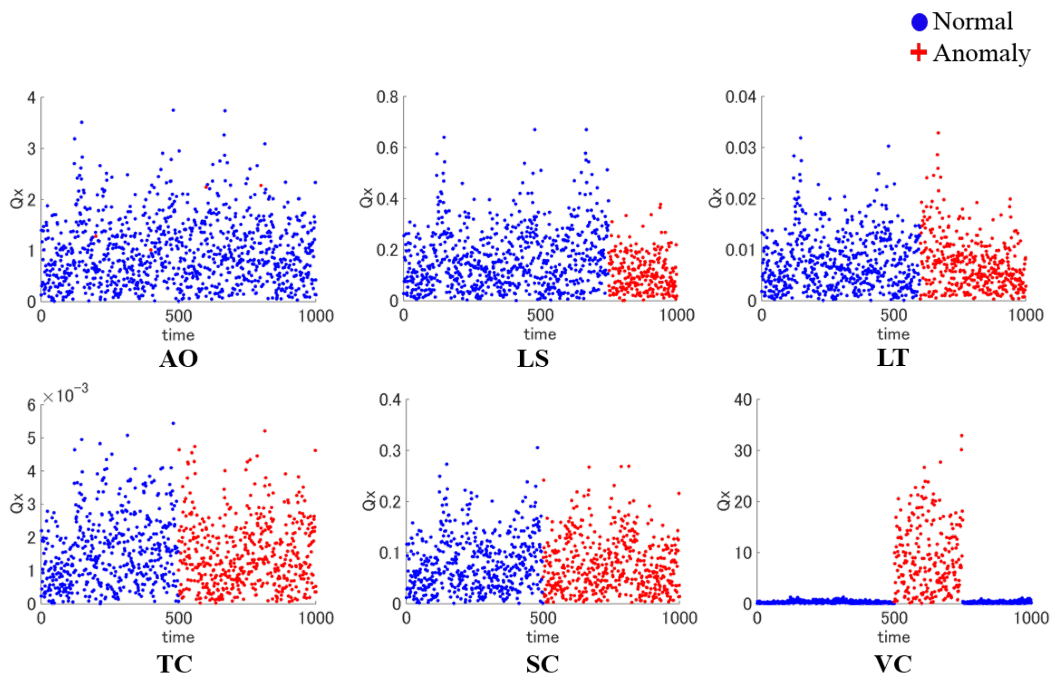

Fig. 2.14 Unsupervised PCA Q plot for single anomaly scenarios.

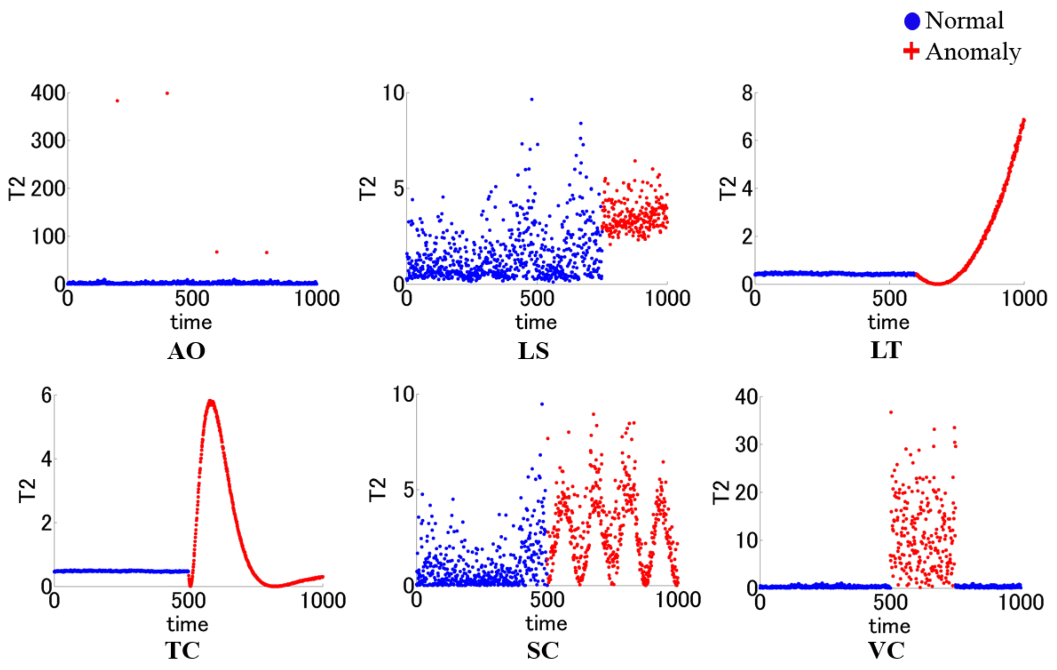

Fig. 2.15 Unsupervised PCA T^2 plots for single anomaly scenarios.

Supervised PCA results can be seen on Figures 2.16 and 2.17, where only normal data is used for PCA model construction and, then, anomalies use this model for Q and T^2 calculation. Compared to the previous unsupervised approach, a slight improvement is noticeable. For some scenarios, however, such as LT, SC and TC, discrimination is rather poor. Despite knowing which samples are which, PCA's linear nature still takes a toll on the overall fault identification performance.

PCA does not include any information regarding the dynamic of the system, which is unfair once the data used is time dependent. The insertion of dynamics here is presented initially using DPCA. While it is true that adding dynamic information provides better insight on the relationship between current and past samples, positive and negative aspects are worth mentioning. In cases where faults are sudden and not sustained, such as AO, not only outliers are detected, but nearby normal data, due to the delay. On the other hand, for faults where anomalies are set only due to their dynamic behavior, some advantages might arise from it. In SC, for example, anomalous samples cross the normal data range, even though their dynamic characteristics are completely different. Figures 2.18 and 2.19 shows unsupervised DPCA Q and T^2 plots.

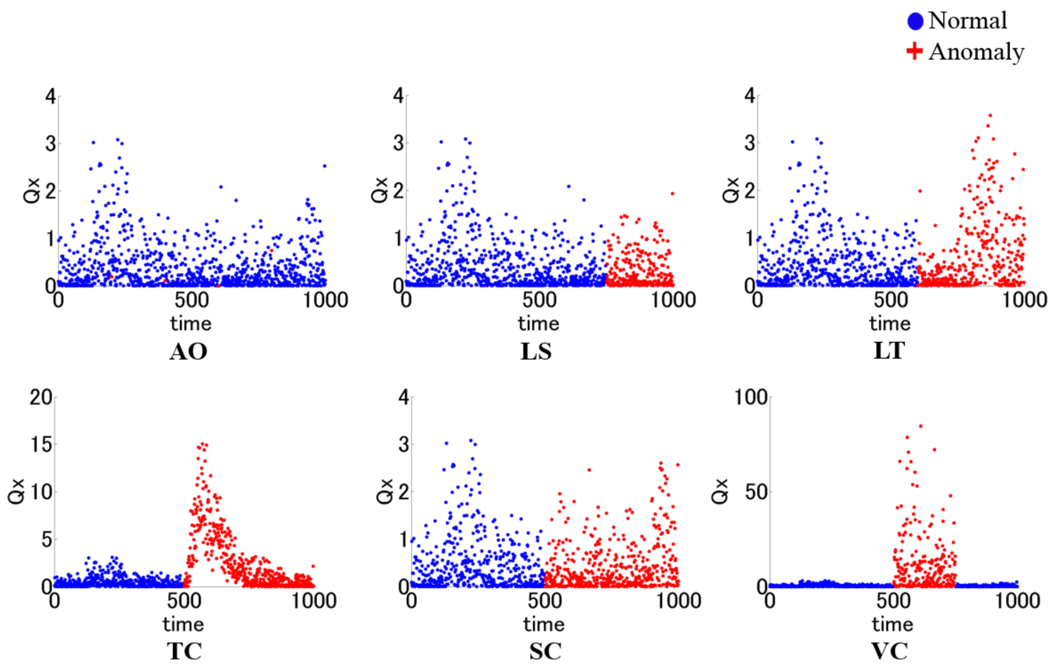

Fig. 2.16 Supervised PCA Q plots for single anomaly scenarios.

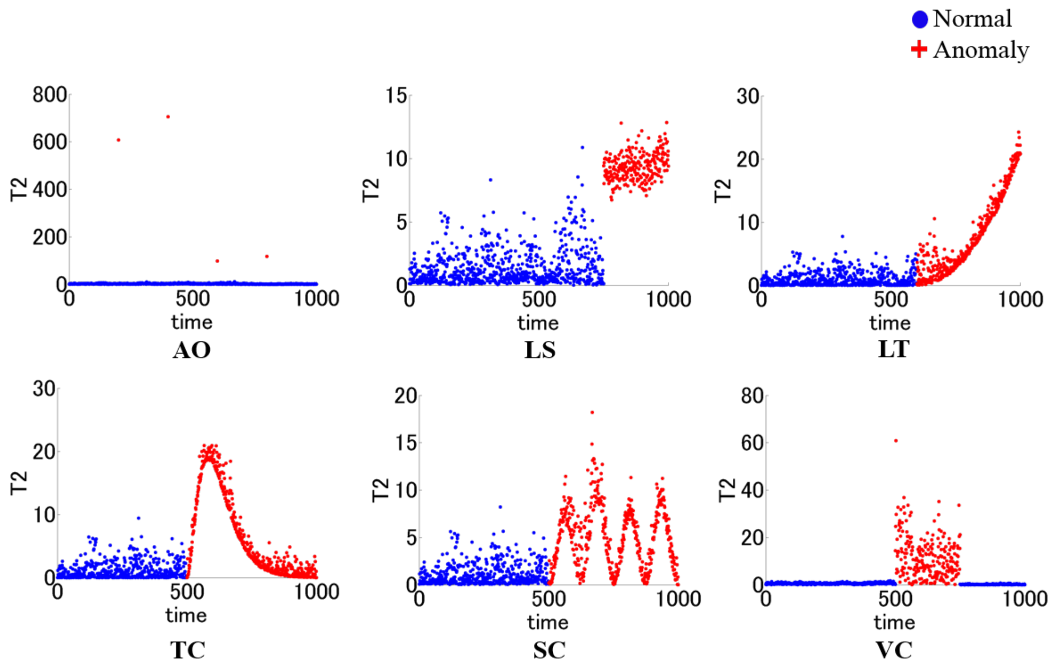

Fig. 2.17 Supervised PCA T^2 plots for single anomaly scenarios.

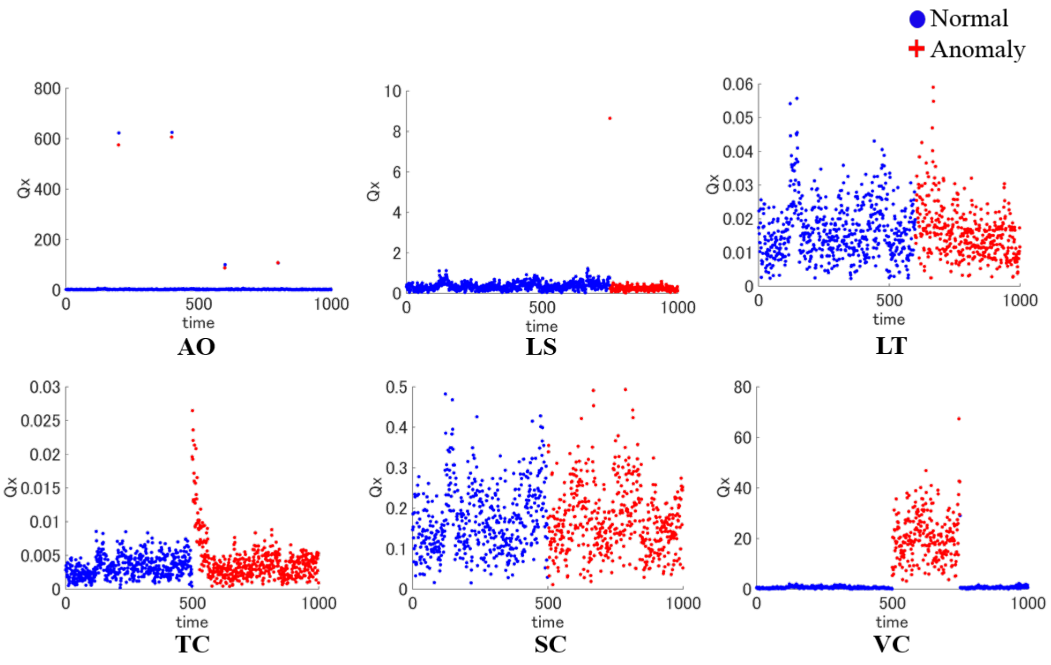

Fig. 2.18 Unsupervised DPCA Q plots for single anomaly scenarios.

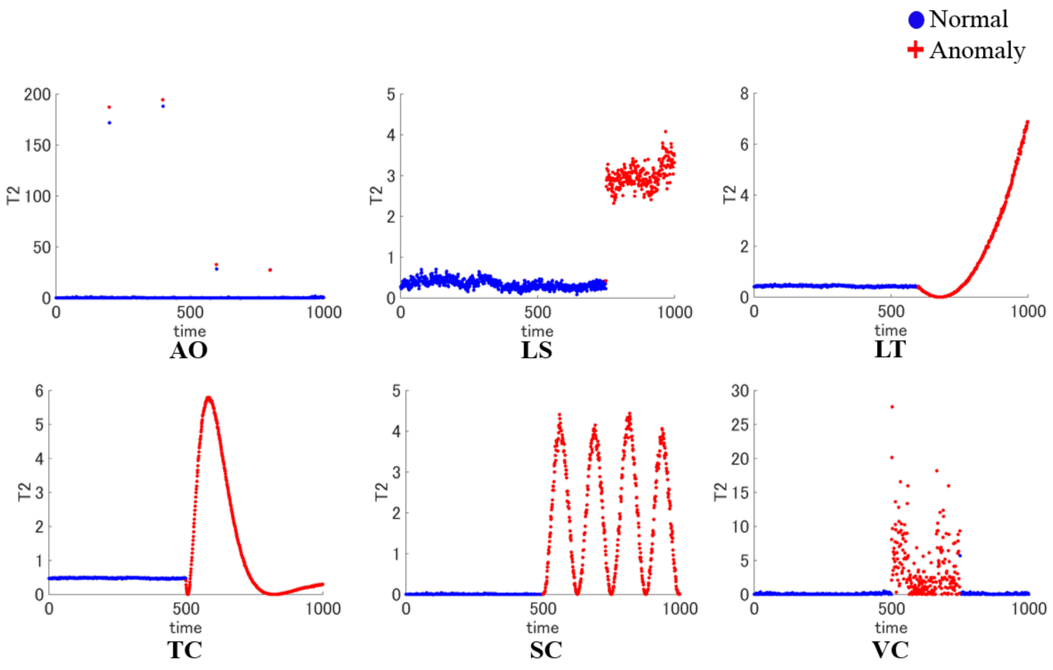

Fig. 2.19 Unsupervised DPCA T^2 plots for single anomaly scenarios.

By relying on dynamic information, discrimination was improved, particularly for LS, TC and VC. Discrimination is not perfect for all samples, but a clear visualization of two or more clusters with distinct characteristics is possible. Evidently, the linear nature of the methodology is still present, which hinders performance. Supervised DPCA, as shown in Figures 2.20 and 2.21, leads to even better discrimination, as expected. For all scenarios, anomaly detection was satisfactory, even if not all samples are detected as such.

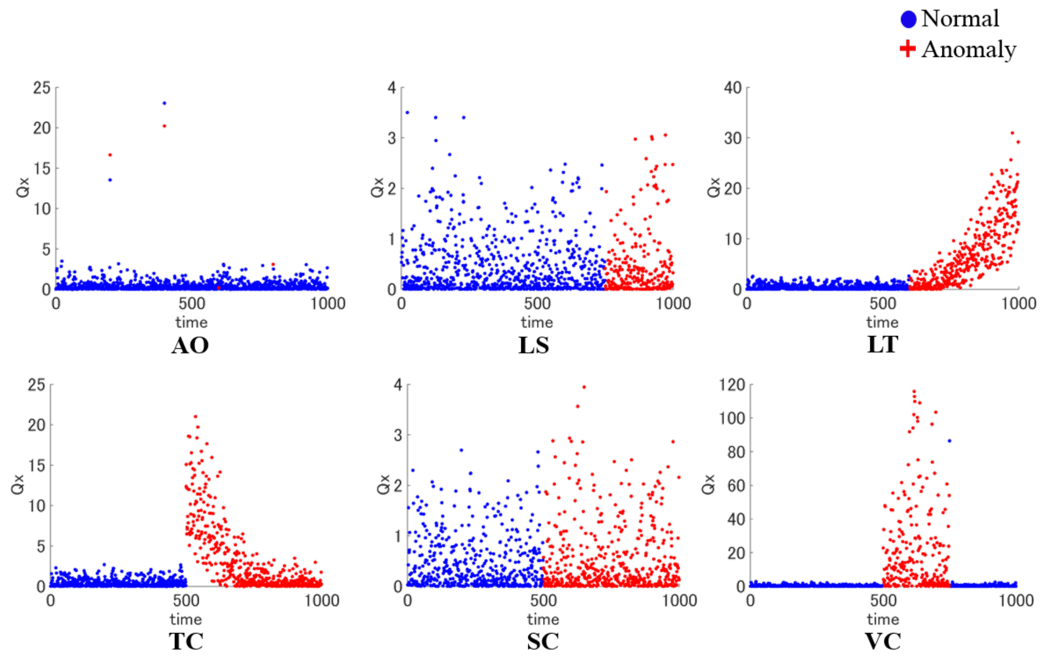

Fig. 2.20 Supervised DPCA Q plots for single anomaly scenarios.

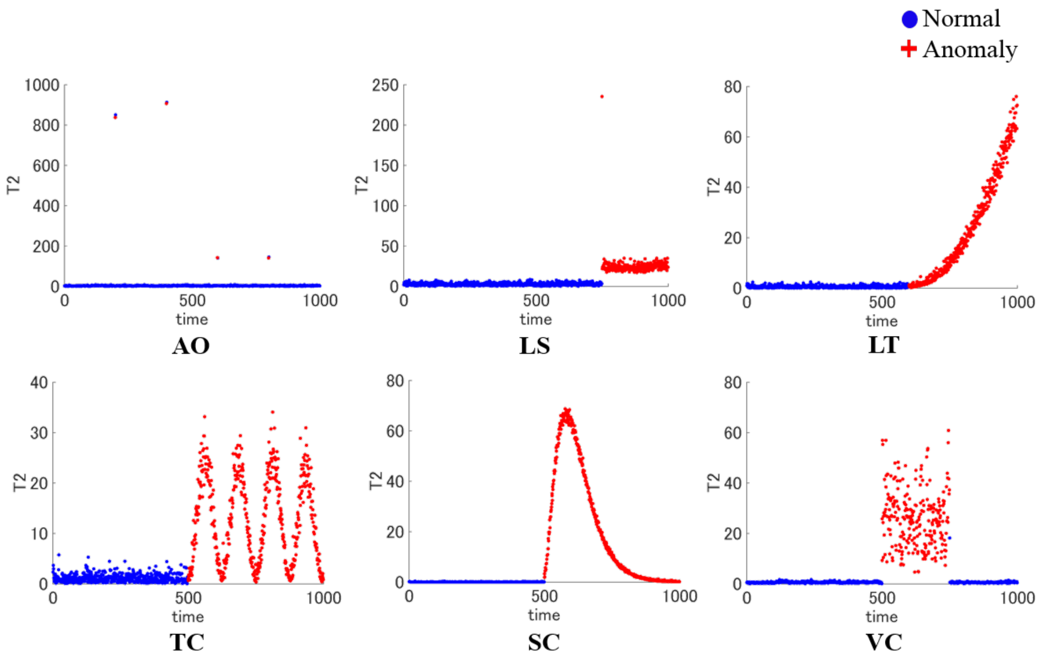

Fig. 2.21 Supervised DPCA T^2 plots for single anomaly scenarios.

DPCA's performance considers evaluating not only variables' range, but also their dynamics. The approach, however, is still linear and, therefore, other aspects related to the inherent nonlinearity of the data might be compromised. From this premise, Figures 2.22 and 2.23 show, initially, unsupervised kPCA Q and T^2 plots. Compared to the other methodologies, a small improvement could be hinted, especially if one considers Q performance as a monitoring index. Even so, still is not satisfactory enough to confirm its applicability as a fully unsupervised methodology. Figure 2.24 and 2.25, on the other hand, show supervised kPCA Q and T^2 results with a rather improved performance. Except for a few samples in SC and VC, most detections were near perfect. The use of kernels for transforming the data had a significant impact in the final discrimination performance, confirming its potential as a supervised methodology.

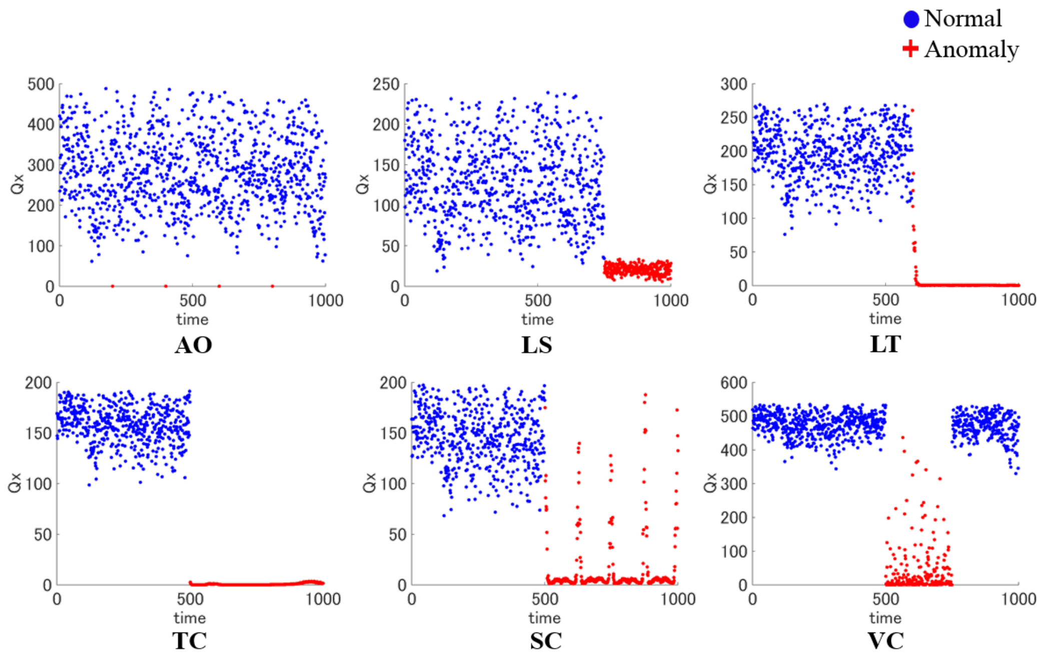

Fig. 2.22 Supervised kPCA Q plots for single anomaly scenarios.

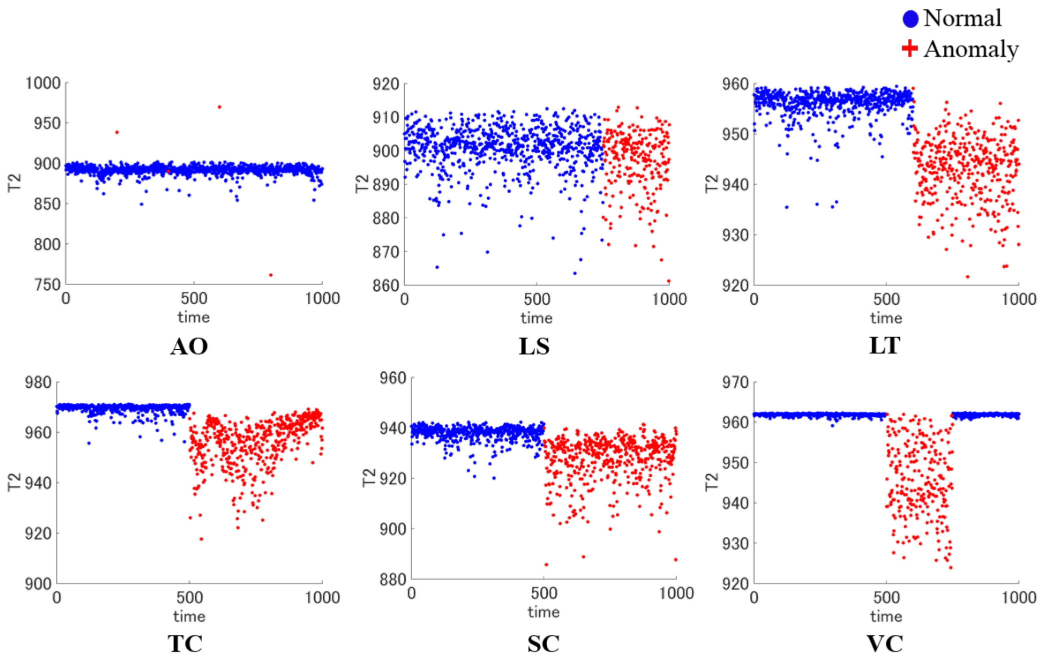

Fig. 2.23 Supervised kPCA T^2 plots for single anomaly scenarios.

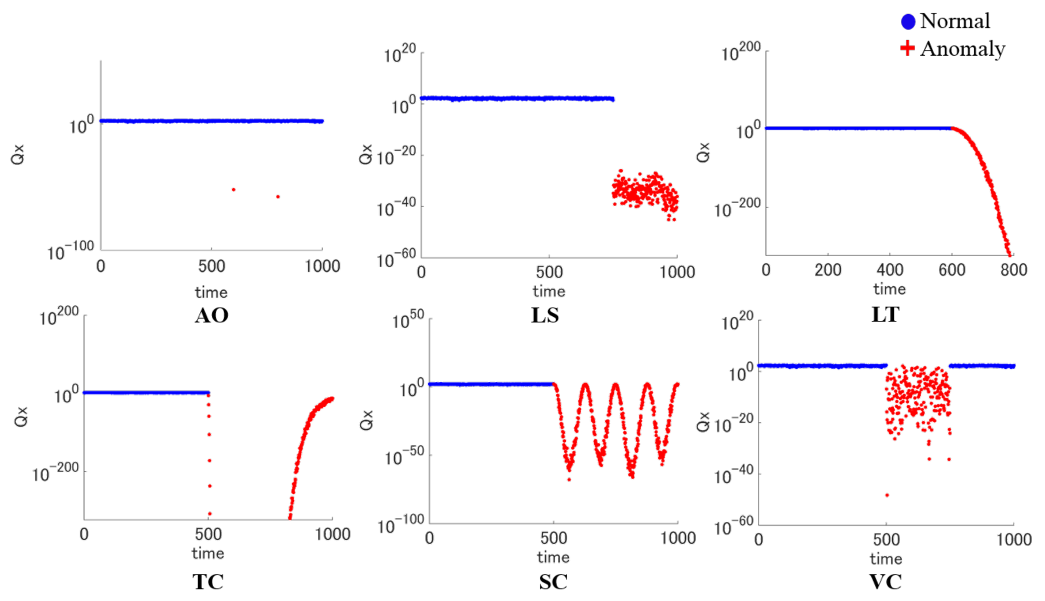

Fig. 2.24 Supervised kPCA Q plots for single anomaly scenarios.

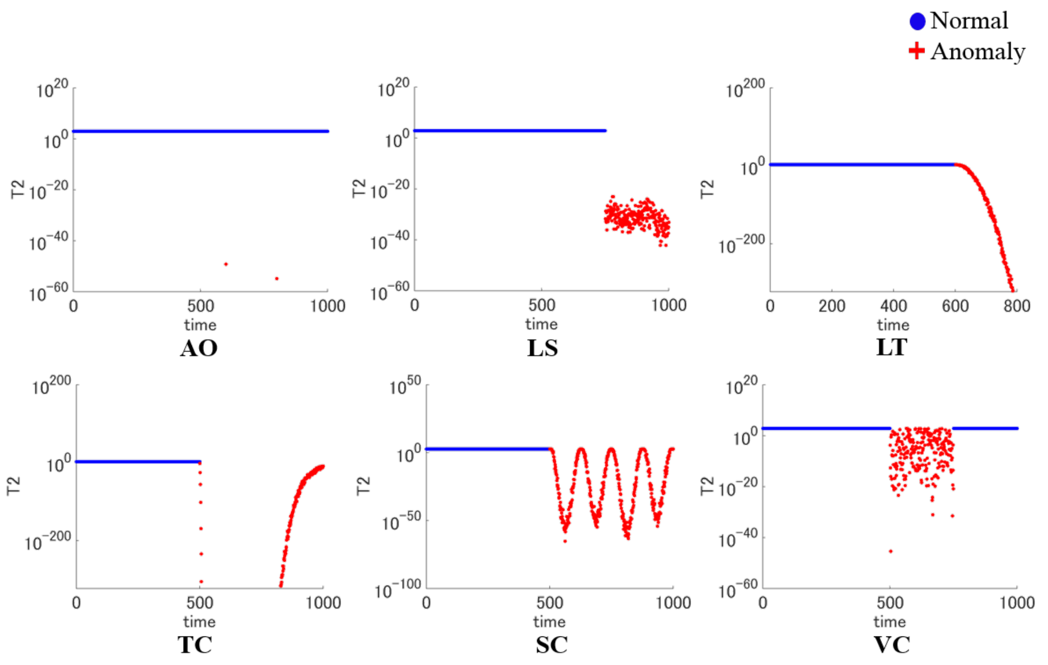

Fig. 2.25 Supervised kPCA T^2 plots for single anomaly scenarios.

As for GTM as a methodology for process monitoring, without considering Graph Theory, the distinction between unsupervised and supervised approaches is far more evident. The latent map is dependent on which samples are being used for training. From a RE, remapping error, perspective, if all samples present in the system are used for training, both normal and anomalous samples will adhere similarly to the non-Euclidean manifold created. Little to no distinction therefore will be attested, as seen in Figure 2.26. As for the similarity assessment index, slightly better discrimination can be seen, but with little improvement, as shown in 2.27. Supervised GTM, on the other hand, shows a clear evolution on discrimination, since outliers now hardly adhere to the map, as shown in Figures 2.28 and 2.29.

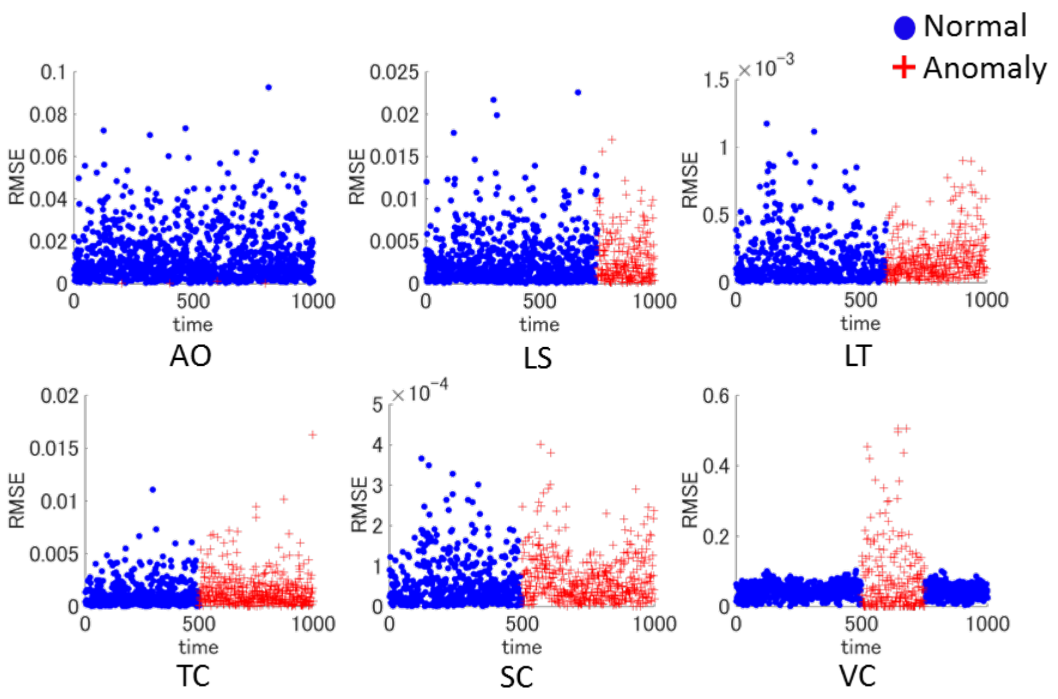

Fig. 2.26 Unsupervised GTM *RE* plots for single anomaly scenarios.

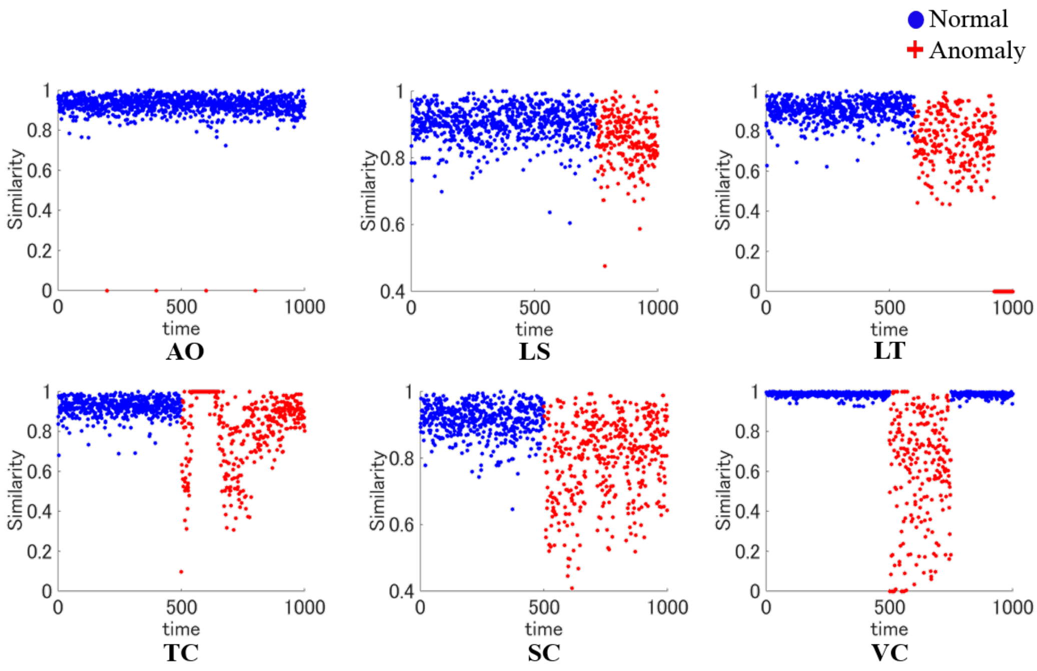

Fig. 2.27 Unsupervised GTM Similarity index plots for single anomaly scenarios.

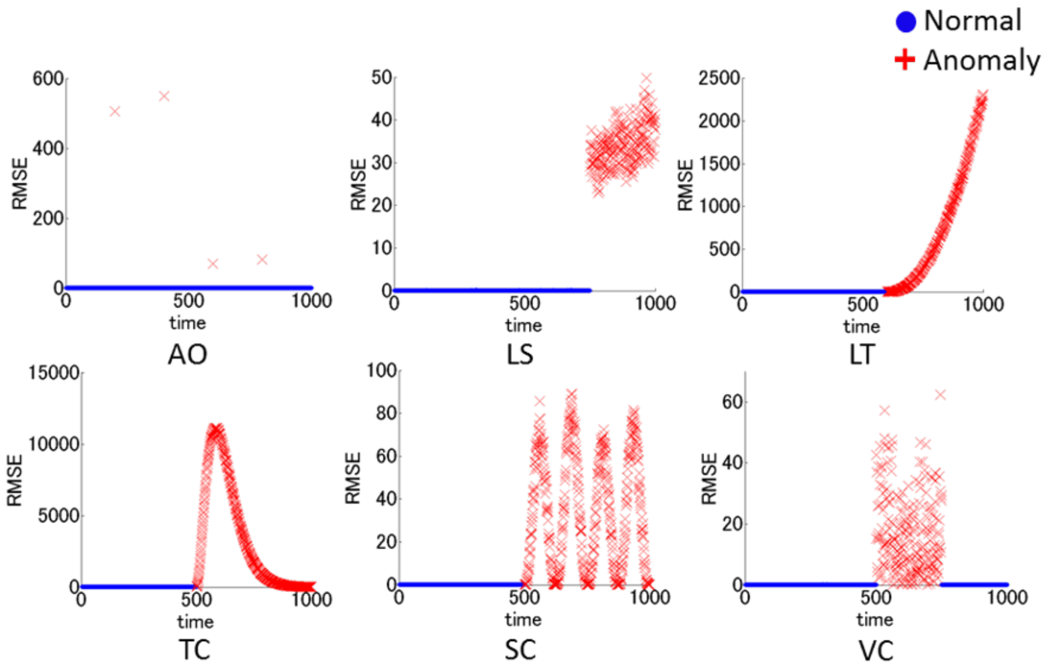

Fig. 2.28 Supervised GTM *RE* plots for single anomaly scenarios.

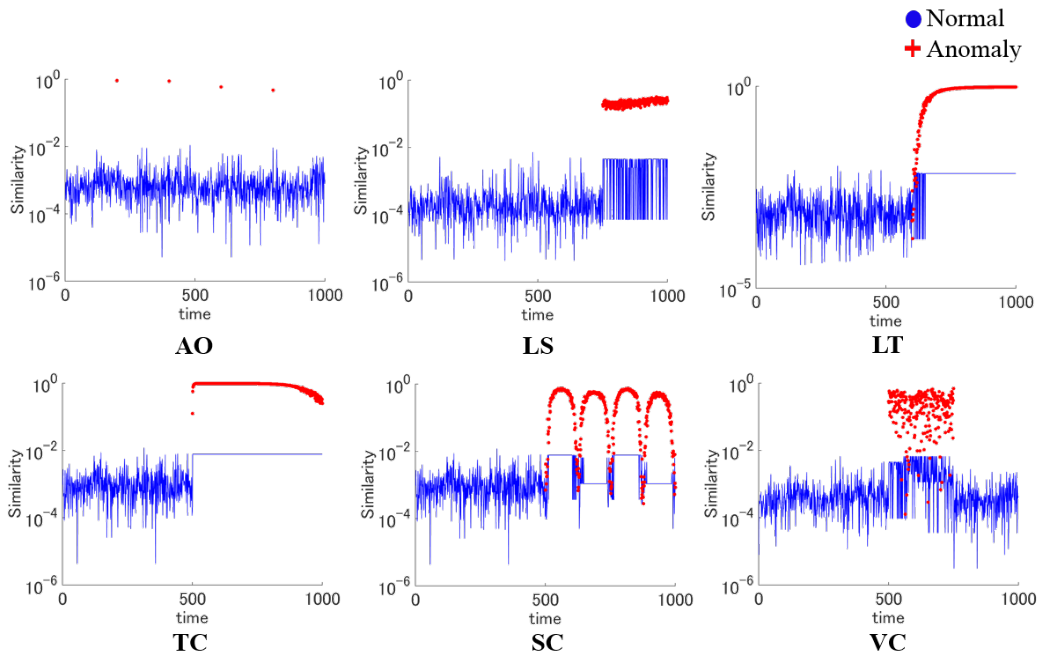

Fig. 2.29 Supervised GTM Similarity index plots for single anomaly scenarios.

Independent Graph Theory analysis, where the AM is constructed by assessing similarity between the original samples using simple correlation, also had poor performance. The similarity matrixes could not detect any differences between normal and faulty data, what compromises the structure of the network generated. Figure 2.30 shows the discrepancy between Graph Theory AM and GTM/Graph Theory AM for the LT scenario, even though the same behavior was apparent for all scenarios. GTM highlights relevant information regarding different states in the plant, where the influence of redundant and unnecessary information is minimized. If the AM cannot recognize any patterns, the network has similar connection density everywhere, resulting in one single cluster.

As states before, the combined GTM and Graph Theory has two key features: on one hand GTM extracts the system relevant information while minimizing the impact of redundant information, where Graph Theory uses this knowledge for visualization and clustering of different states. Figure 2.31 and Figure 2.32 show such behavior. Figure 2.31 networks present all samples without connection, so to ease the visualization of different clusters. Figure 2.32 networks show all nodes and edges for each network.

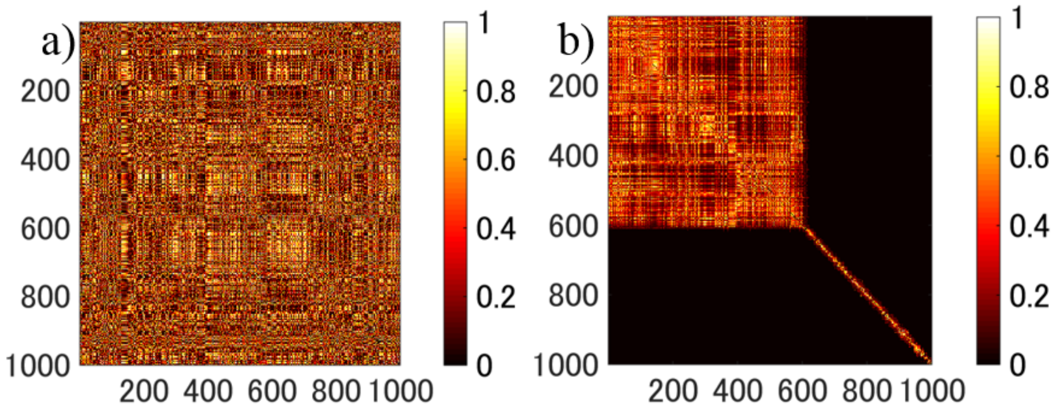

Fig. 2.30 AM matrix for a) Graph Theory and b) combined GTM and Graph Theory approaches, considering the LT scenario.

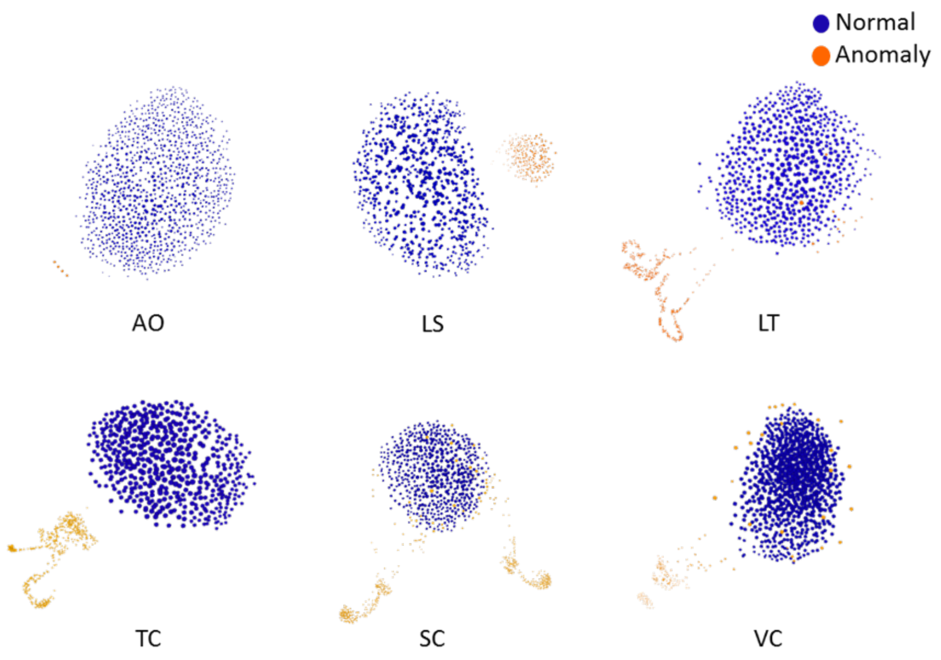

Fig. 2.31 GTM + Graph Theory networks for single anomaly scenarios represented without any edges.

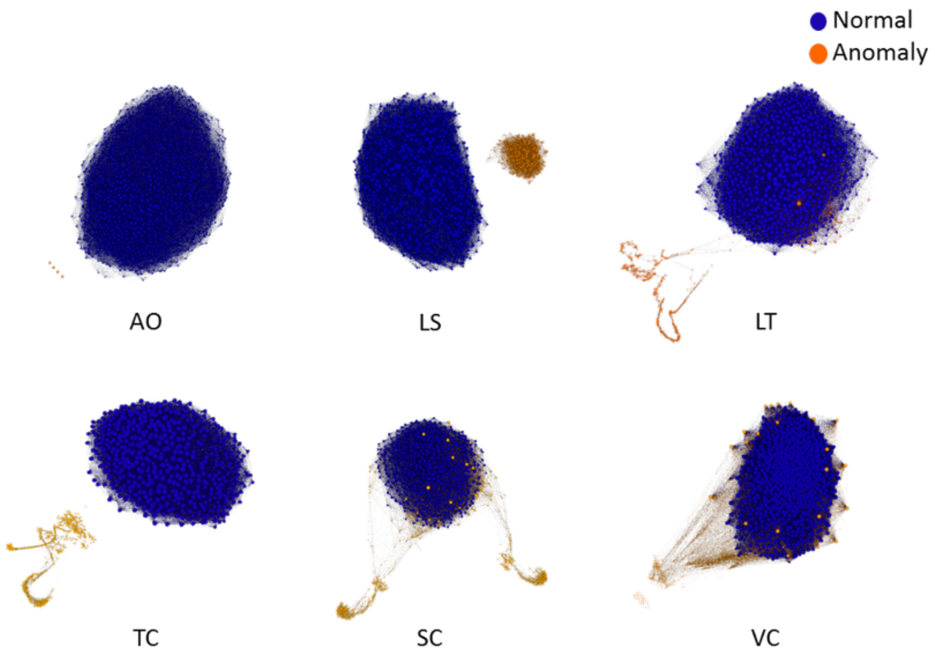

Fig. 2.32 GTM+Graph Theory networks for single anomaly scenarios represented with all available connections.

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. Nonetheless, these results motivate the use of the proposed approach for process monitoring. One important aspect to be considered is that this approach is fully unsupervised. Compared to the other unsupervised approaches, the proposed approach outperformed all techniques presented. Furthermore, it performed as well as or close to the supervised approaches, revealing how the analysis of the relationship between variables might be enough for data discrimination.

The idea behind using unsupervised analysis can be misinterpreted as ignoring all knowledge available. It is important to clarify the tone of the analysis proposed. Using unsupervised approaches for data analyzing does not intend to ignore important information, but rather to support methodologies already implemented by offering a perspective free of biases, relying only on the relationship between variables and samples. What is offered is the analysis before the analysis, *i.e.*, to not rely on the preconceived notion that all existent information is reliable.

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 information and incorporate it to the normal data set, 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.

2.5.2 Multiple Anomaly Scenario

This scenario considers multiple anomalies in the same data set, where not only normal and anomalous data have to be discriminated, but also that within anomalies, it should be possible to discriminate between different types of faults. For this step, the whole methodology is considered, including normal data refinement, so to compare the evolution of the discrimination and motivate the application to the Tennessee Eastman Process case study. The simulation consists of three variables, where two of them are highly correlated. In order to mimic such behavior, the same Brownian motion sequence was associated to two variables, with distinct Gaussian noise for each variable. The remaining variable has both independent Brownian motion and Gaussian noise.

The normal data benchmark was created using Equation 33, with baseline $\mathbf{x_b} = (0.1, 0.2, 0.3)$. Variables $\mathbf{x_1}$ and $\mathbf{x_2}$ are correlated. 700 samples were generated, where the first 300 are associated to normal data and the remaining 400 are split in four different outliers, each with 100 samples. Differently from what was devised in section 2.5.1, the anomalies are now local, *i.e.*, they do not affect all variables simultaneously, but the structure of the faults still follows Equation 34. Figure 2.33 shows the variable plot and Table 2.2 details the parameters involved for each anomaly, including which variables are affected by each anomaly.

Table 2.2 Parameters for multiple anomalies scenario generation.

	Variables Affected	$\mathbf{t_j}$	$\mathbf{o_j}$
O1	$\mathbf{x_1}$	$\mathbf{t_{O1}} = [301\ 302\ \cdots\ 400]$	$\mathbf{o_{O1}} = 0.0005(\mathbf{t_{O1}} - 300)$
O2	$\mathbf{x_1}$ & $\mathbf{x_2}$	$\mathbf{t_{O2}} = [401\ 402\ \cdots\ 500]$	$\mathbf{o_{O2}} = 0.05$
O3	$\mathbf{x_1}$ & $\mathbf{x_2}$	$\mathbf{t_{O3}} = [501\ 502\ \cdots\ 600]$	$\mathbf{o_{O3}} = 0.089 + 0.005\mathbf{b_{O3}}$
O4	$\mathbf{x_3}$	$\mathbf{t_{O4}} = [601\ 602\ \cdots\ 700]$	$\mathbf{o_{O4}} = 0.1\mathbf{b_{O4}}$

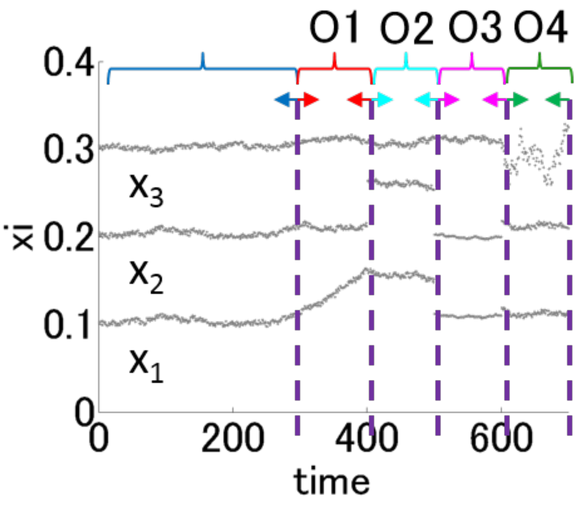

Fig. 2.33 Multiple anomalies simulation data set, where each color delimits different states.

Each anomaly presented here has a different characteristic. O1 is a slow drift acting on variable $\mathbf{x_1}$. O2 is a sudden sustained change in variable $\mathbf{x_2}$, accompanied by variable $\mathbf{x_1}$. O3 is a change in correlation values for both $\mathbf{x_1}$ and $\mathbf{x_2}$, where the range of the variables does not change, only the relationship between them. This type of anomaly is particularly difficult to be detected. Finally, O4 expresses random variation in variable $\mathbf{x_3}$, result most likely from some equipment malfunctioning. The delay d chosen for defining time delayed variables is 1 and the initial λ is 1/3.

Figures 2.34 and 2.35 shows the comparison between PCA, DPCA and KPCA for both unsupervised and supervised approaches. Figure 2.36 shows RE and Similarity assessment results for GTM.

The case presented now is more complex than the previous one for two reasons. First, the presence of multiple anomalies itself leads to a more complex representation of data, where now more than two clusters are necessary for representing the data set. In addition, now faults are local and not global anymore, what also penalizes how much difference there is between each normal and anomalous states.

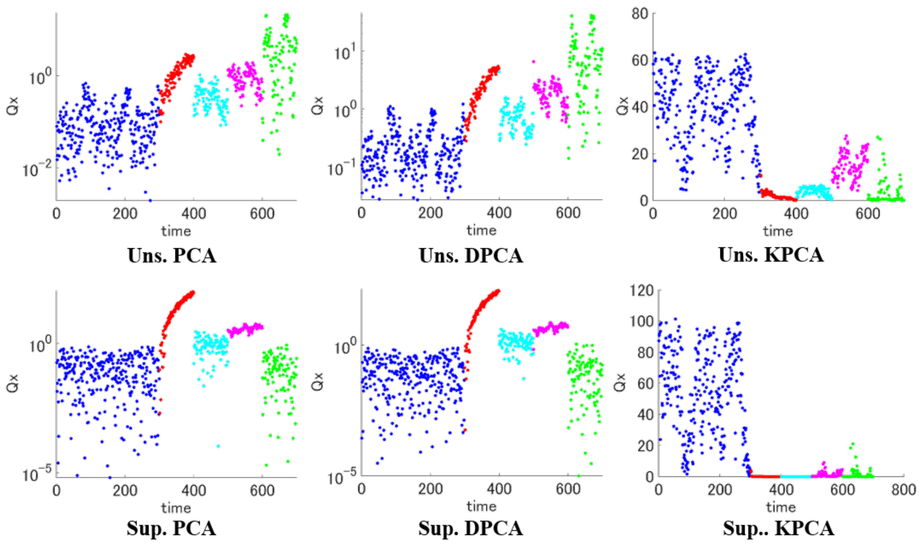

Fig. 2.34 PCA, DPCA and KPCA Q monitoring results for both unsupervised and supervised results, following the color scheme depicted in Figure 2.33.

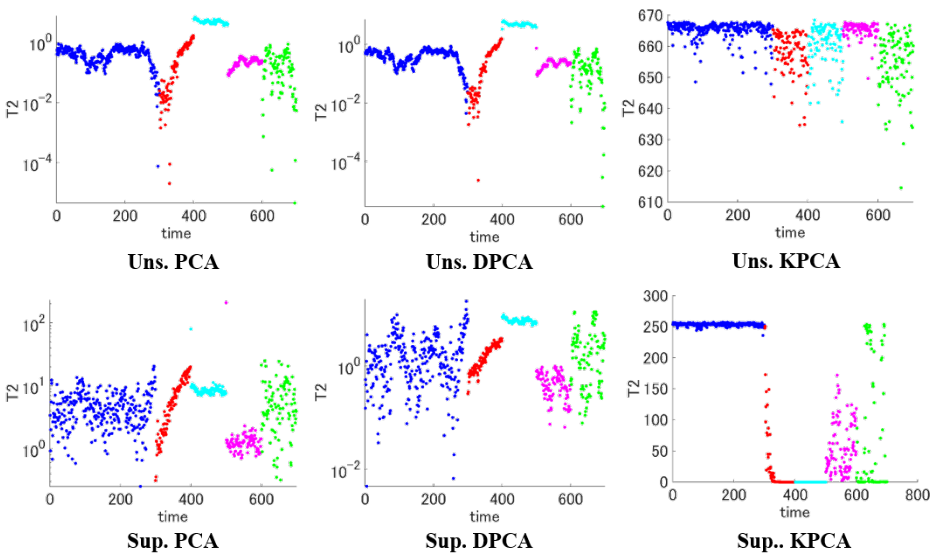

Fig. 2.35 PCA, DPCA and KPCA T^2 monitoring results for both unsupervised and supervised results, following the color scheme depicted in Figure 2.33.

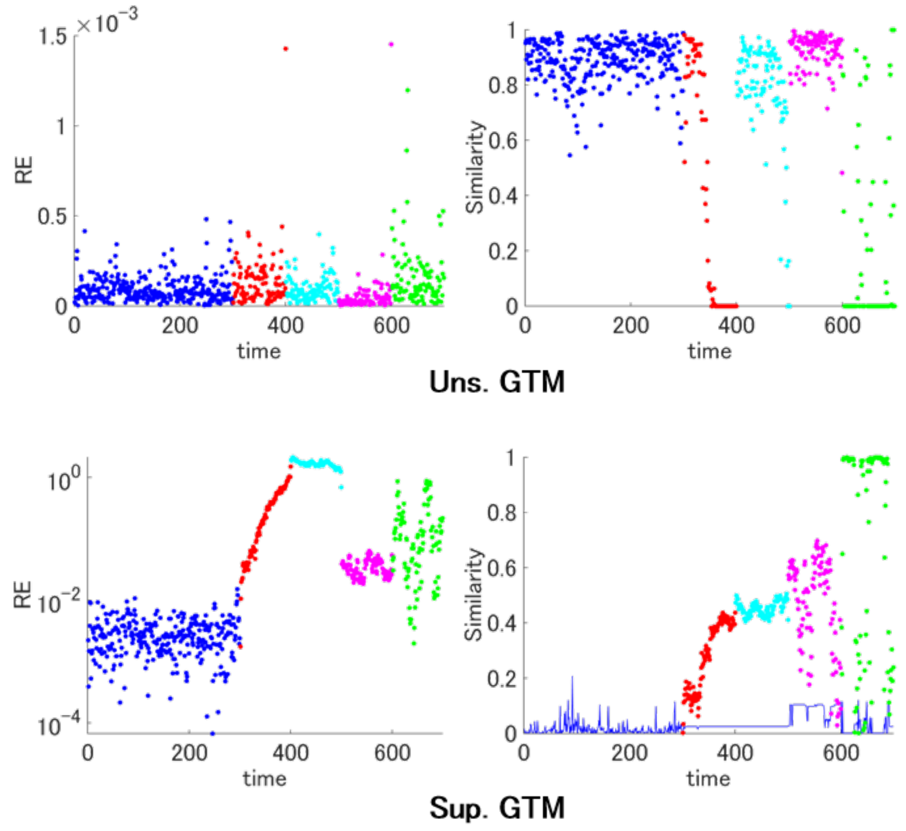

Fig. 2.36 GTM monitoring results for both unsupervised and supervised results, following the color scheme depicted in Figure 2.33.

PCA, DPCA and KPCA have similar performance, where for both unsupervised and supervised approaches, finding clean normal and anomalous clusters is not feasible. For the unsupervised approach, the overlap between different states is small, even though clear differentiation between them is not possible. For the supervised results, O4 and O2 anomalies overlap with the normal data set, which is not an indication of good fault detection.

GTM, as expected, is ineffective as an unsupervised methodology, since all data is used for map training. The supervised approach, however, has a good performance, where most anomalies are soon detected, even though for anomalies O1, O3 and O4 some samples are falsely distinguished as normal. For the supervised approach, it should be noticed that the training data threshold changes for each abnormal sample tested, since it represents the similarity associated to the maximum similarity value of those samples.

For this case study, the combined approach is fully explored, including refinement of the NC, according to all steps described in section 2.4.2. Only two cycles were necessary for finding the optimal NC, where Figure 2.37 shows the evolution of the NC from the first network.

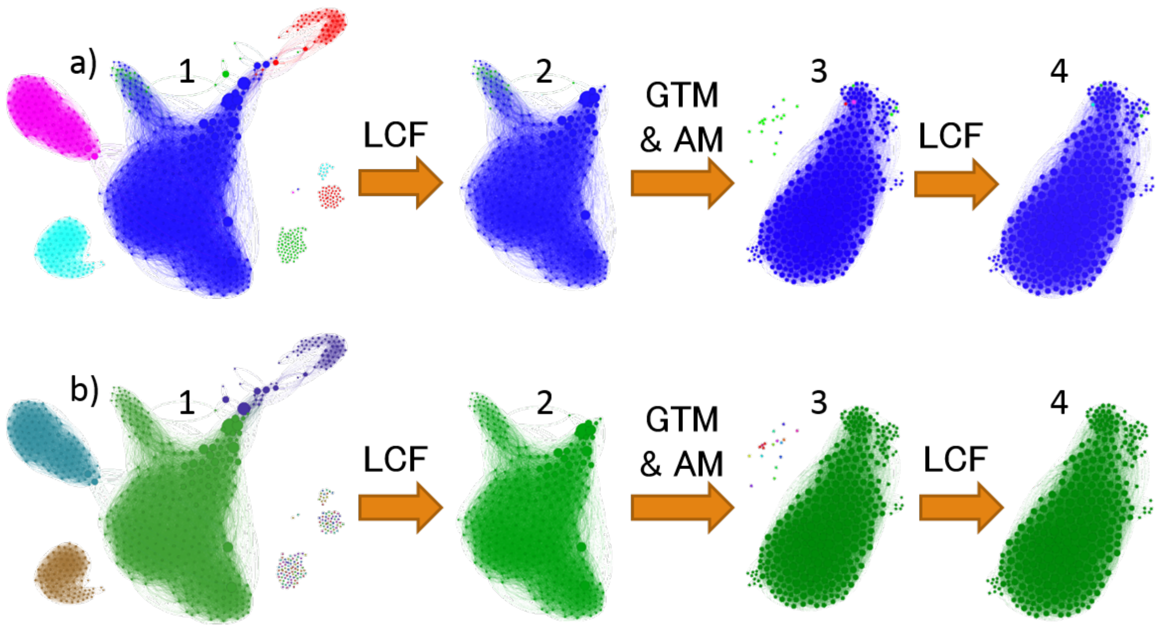

Fig. 2.37 Evolution of NC where a) shows data label for normal and anomalous states and b) shows clustering results.

Initially, all samples are trained, resulting in the first network marked by 1. NC is then extracted based on the biggest cluster available, showed in 2. A new GTM map is obtained by training only the NC, which followed by AM calculation results in the new network depicted in 3. After another NC extraction, showed in 4, further GTM maps and network generation led to one single cluster, satisfying the criterion established. Once the process is finished, all networks connections in each step are fed back to the original network, resulting in what can be seen in Figure 2.38, which shows the clustering results and actual labels of the final network found.

From the NC perspective, the discrimination obtained is near perfect, where only a few abnormal samples are not detected. Once more, the key aspect to highlight here is that the methodology is unsupervised from start, not relying on any labels. Even when compared with supervised techniques, the proposed methodology outperformed them. It is true that the methodology was only initially unsupervised, since in later cycles labels were assumed, similar to a self-learning process. Even so, there was no external knowledge available other than the relationship between variables and samples. Furthermore, one could also argue on the nature of the data presented. According to the characteristics of the methodology presented, it is fair to say that the contaminated data in the NC is similar to the NC itself.

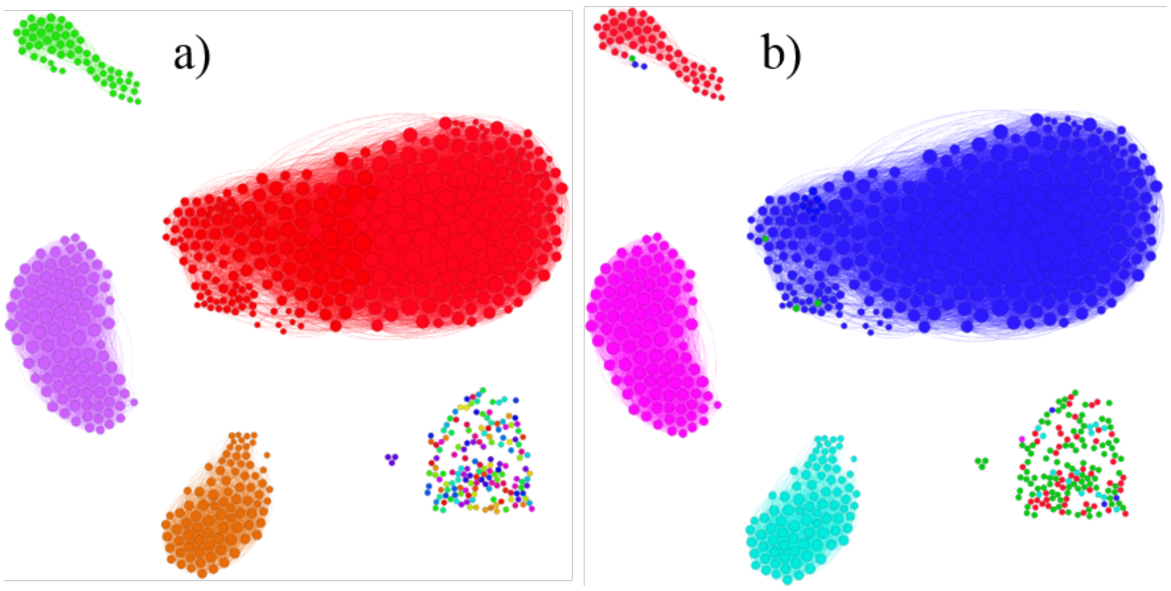

Fig. 2.38 Final network for the multiple anomalies scenario showing a) clustering results, where different colors are different clusters, and b) network labeled for reference.

Taking into account the timeline of data, it is common sense to assume that those samples are anomalous, since all samples around them are. If similarity with the NC is high, however, considering them as normal has little to no impact on the data set. On a bigger scale, how reliable was assigning those samples as outliers is the issue.

Besides data discrimination itself, data visualization aspects can also be discussed, related to the nature of each fault. The structure of the network and the clusters hints on the structure of the normal and faulty data. The NC obtained is a cluster with high connection density, *i.e.*, there is a high ratio between the number of nodes and edges. This is an indication of a stable state, since a lot of samples relate to each other. This is not a characteristic of normal data only, though. Both O1 and O3 represent data that, while anomalous, are stable. Understanding the characteristic of those faults, one can investigate further their origin on a more oriented basis. O2 cluster has a lower density, implying that only a few samples are similar to each other. This is an indication of constant transition, where samples at the beginning of the fault might not relate to samples at the end of the fault. Such structure matches the slow drift structure of O2. Finally, when the behavior is erratic, such as the one portrayed by O4, each sample has a low likelihood of relating to any other faulty sample, resulting in isolated nodes, with no connection to other ones whatsoever. Almost all data belonging to O4 exhibit this behavior. By analyzing not only which clusters were found, but also what their characteristics are, further insight can be gained on the nature of outliers.

On Online Monitoring

In order to apply the proposed methodology in a real case study, evaluating the methodology beyond its training data is fundamental. With that mind, a test data set is presented, so to motivate the use of GTM and Graph Theory for online monitoring. Figure 2.39 shows the structure of the test data. The structure of normal and faulty data follows the one shown in Table 2.2, with different noise and Brownian Motion. In addition, the order of the anomalies is different.

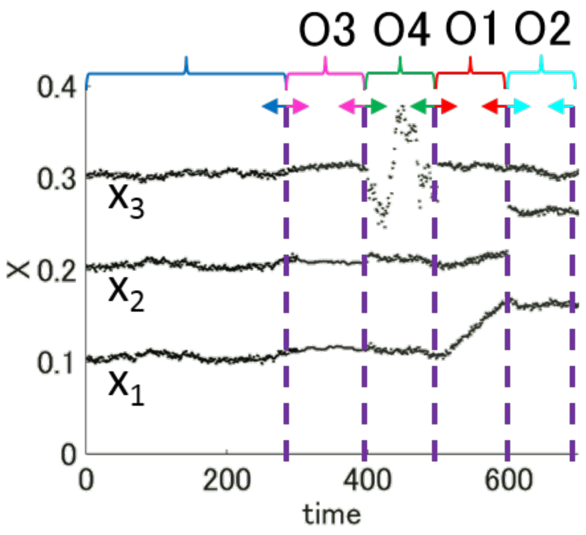

Fig. 2.39 Multiple anomalies simulation test data set, where each color delimits different states.

When evaluating test data, one can assume that the proposed strategy (GTM+GT) has already detected within the training database which data is normal and which data is anomalous. From that point on, the analysis is no longer unsupervised. From exclusively a fault detection perspective, GTM+GT could evaluate test data, however other supervised methods can be used instead, saving processing time. GTM was the methodology with the best performance and, therefore, it is used for this assessment.

Figure 2.40 shows supervised *RE* and Similarity index results for test data, respectively, using training data's threshold. Figure 2.41 depicts better the inferred labels according to GTM, so to highlight the detection performance. Overall, the methodology can detect fairly well normal and anomalous data, except for a few instances. More importantly, however, is knowing that there was no detection delay in this scenario, *i.e.*, anomalies were detected as soon as they occurred.

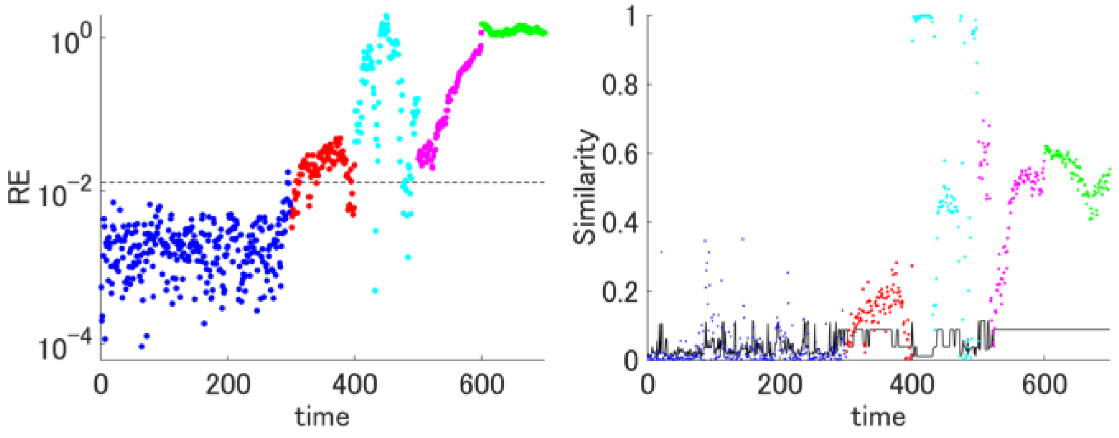

Fig. 2.40 GTM monitoring results for both unsupervised and supervised results with the detection threshold highlighted in black, following the color scheme depicted in Figure 2.39.

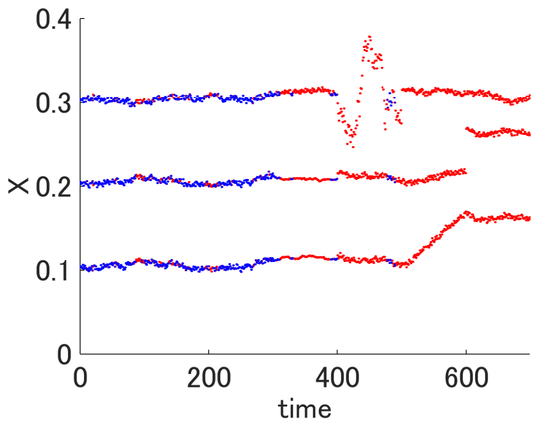

Fig. 2.41 GTM normal and anomalous inferred labels for the test data set.

The normal instances undetected or the anomalous ones wrongly detected as normal reveal important characteristics, and limitations, regarding the approach taken in this work. In order to select the NC, GTM+GT relies heavily on similarity, since it depends heavily on the AM obtained. In the test data, some normal samples deviate from the original NC obtained, indicating their lack of similarity with that cluster. This limitation hints at applicability domain issues, where the NC obtained should be broad enough to encapsulate a fair range of normal states, and it should be considered more thoroughly for future works.

When it comes to anomalies being detected as normal data, however, a different aspect of similarity can be considered. Knowing that time-delayed variables were used for GTM+GT evaluation, both variable range and dynamics are being considered for similarity assessment.

The samples mislabeled as normal, therefore, are indeed similar to the NC in both aspects. Despite their anomalous nature, they affect little the existent pool of normal data, since their values are within boundaries of normal data variation. This behavior assumes, evidently, that normal clusters in the training data set have little contamination. Despite those issues, however, most of the detection was successful, motivating its application for more complex scenarios presented in the following section.

2.6 Case Study - Tennessee Eastman Process

2.6.1 Data Characterization

TEP is a realistic virtual industrial process whose data sets are valuable for process control and monitoring evaluation. The schematics of the plant can be seen in Figure 2.42. The system consists of eight compounds (A,..., H), 12 manipulated variables and 41 process (measured) variables. In order to evaluate fault detection capabilities, 21 preprogrammed faults are available, consisting of both training and test data sets. Each training data set consists of 500 initial normal samples followed by 480 anomalies.

As for the impact of the outliers on the chemical plant, TEP differs from the scenarios presented previously. For the simulation data sets, the system was an open loop and each variable channel was, from an outlier perspective, independent from one another. Each anomaly, therefore, had no impact on the other variables. TEP variables, on the other hand, are affected in two different ways. First, depending on the physical structure of the plant, anomalies in one variable can affect other ones in the system. If the A feed depicted in Figure 2.42 is affected, for example, variables related to the reactor will probably be affected as well. Moreover, TEP is a closed loop, where several PID controllers are constantly reacting to changes in the plant. This results in seemingly unrelated variables changing to compensate the anomaly. TEP is a far more realistic approach to what is experienced in real plants and it is a more challenging case study to analyze. All manipulated variables and all, but one, process variables are used as input, giving a total of 52 variables for assessing relationship between samples. From the literature[57], TEP's delay d is defined as 2, since a more complex system needs a bigger delay for dynamic information to be collected.

For the work presented here, not all faults are being considered for analysis. Table 2.3 shows the nine faults chosen for evaluation, considering anomalies with different characteristics. Figure 2.43 shows all nine plots for all TEP variables already centered and scaled.

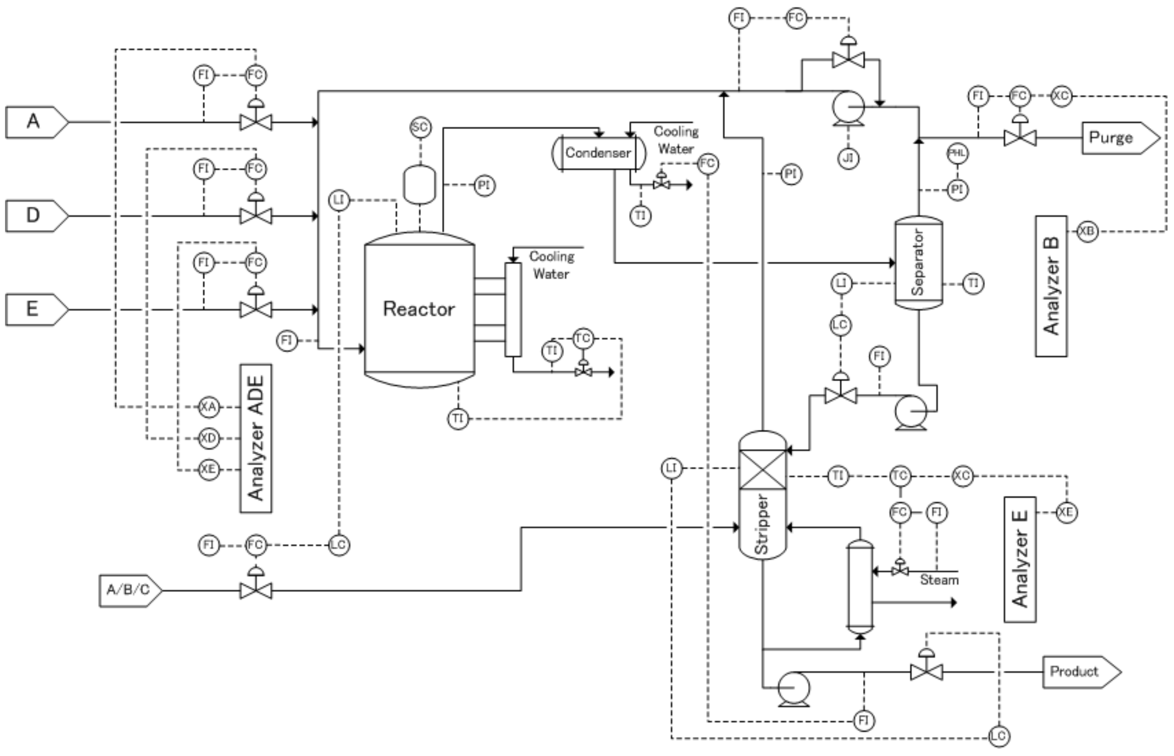

Fig. 2.42 Tennessee Eastman Process flow sheet.

2.6.2 On GTM and Graph Theory Combined Approach

The discussion presented so far evaluated the distinction in performance from a qualitative point to view, using monitoring plots and networks to assess how well each methodology could discriminate between normal and anomalous data. From a more quantitative point of view, however, the following results try to go beyond the visual cue given by those approaches, even though specially from a network analysis perspective, those cues bring the entire system to an easier representation. For the proposed approach, outlier assessment relies on the final network obtained for each scenario, by checking normal and anomalous clusters. Figure 2.44 and Figure 2.45 show the networks obtained, with and without connections.

Table 2.4 shows the confusion indexes for each scenario. Confusion indexes consists of evaluating four different, yet related, values for assessing outlier detection performance. True positive (TP) shows how many samples were correctly detected as outliers. True negative (TN) shows how many samples were correctly identified as normal. False positive (FP) shows the ratio of normal samples falsely detected as anomalies. False negative (FN) shows the ratio of abnormal samples wrongly detected as normal samples. True indexes should be close to one and false indexes should be close to zero.

Table 2.3 Preprogrammed faults in TEP process.

Fault ID	Description	Type
F1	A/C feed ratio, B composition constant	Step
F2	B composition, A/C ratio constant	Step
F5	Condenser cooling water inlet temperature	Step
F6	A feed loss	Step
F7	C header pressure loss - Reduced Availability	Variation
F8	A,B,C feed composition	Variation
F12	Condenser colling water inlet temperature	Variation
F13	Reaction kinetics	Slow Drift
F17	Unknown	?

Table 2.4 Confusion matrix for all scenarios.

	F1	F2	F5	F6	F7	F8	F12	F13	F17
TP	99.4	100	100	100	100	96.6	93.6	97.0	94.4
FP	0.6	13.0	11.4	13.6	0.2	0	0	0.8	14.2
TN	99.4	87.4	88.6	86.4	99.8	100	100	99.2	85.8
FN	0.6	0	0	0	0	3.4	6.4	3.0	5.6

When GTM and Graph Theory are combined for detection, there is overall good anomaly detection. For all step faults, detection is near perfect and even for the other faults, contamination of the normal database is small. There is a trade-off associated to this detection, however, as it can be seen by the FP results. Depending on the scenario, a significant percentage of data is considered anomalous. According to the similarity principle considered, it is fair to say that those samples do not relate much with the existent normal data set. Knowing that the normal samples are essentially the same for all scenarios, the outliers have an influence on the process. GTM training involves initially all samples, which led to different initial maps for each scenario, as one would expect. This discrepancy can be augmented over the cycles, leading to different final maps for the NC where certain samples are more likely to detach than others do. F2, F5, F6 and F17 networks have a substantial part of normal samples detached, where most of them are recurrent in at least two of those clusters, corroborating the fact that certain samples are more prone to detachment due to their dissimilarity. When creating the network, it is undesirable to lose too many samples, since the inner variation of the data set can be compromised. This effect, however, is not big, does not affect all scenarios and it helps with the faulty detection itself.

Along with the discrimination itself, certain aspects of visualization can also be taken into account. The structure of anomalous clusters reveal important information about the

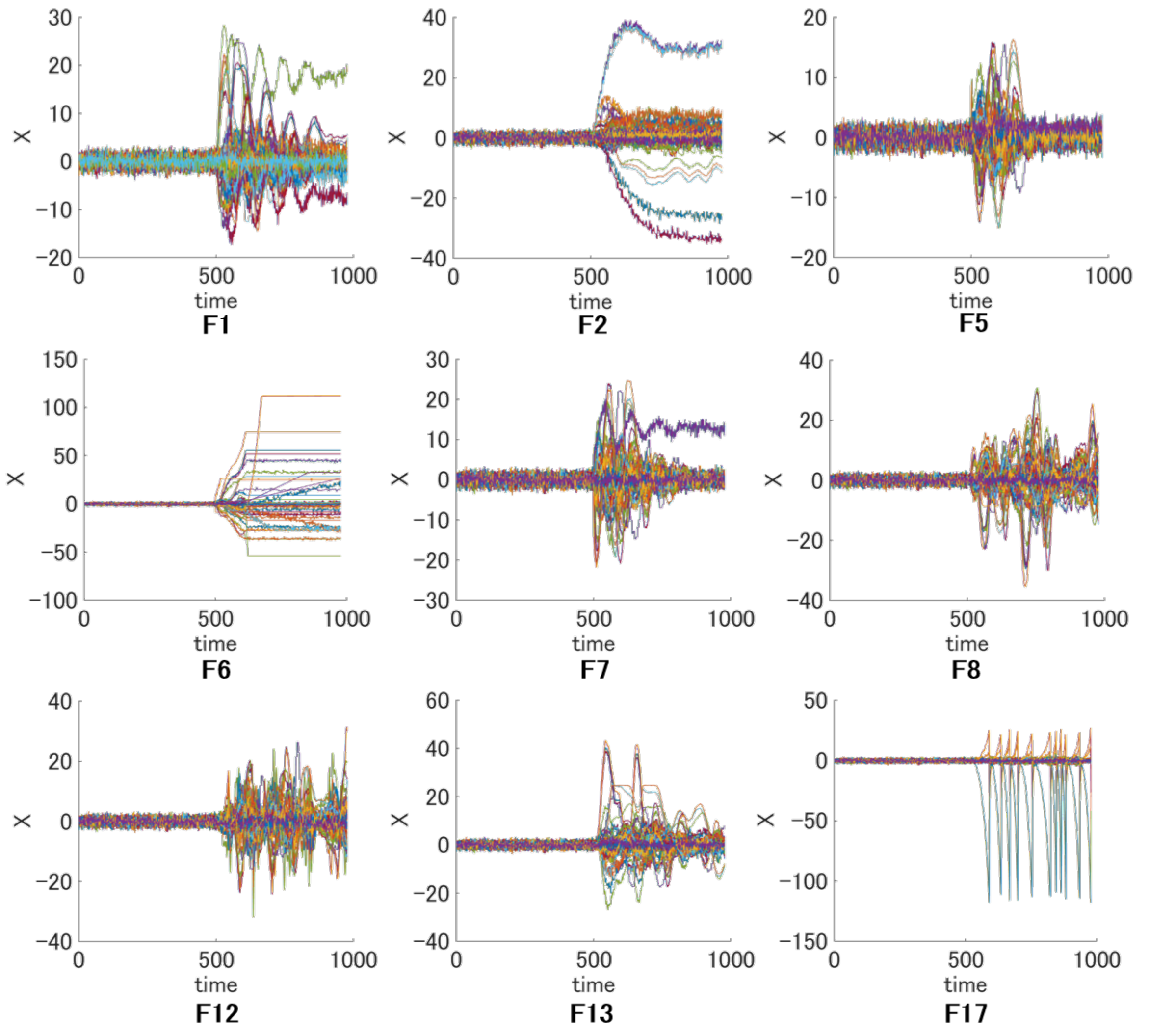

Fig. 2.43 Input variable plot for TEP preprogrammed faults.

nature of the faults. For F1 and F2, for example, the step faults eventually led to a second stable state, which can be identified by the formation of a second big, densely connected cluster. For both cases, however, the initial transition is characterized by disconnected or poorly connected clusters. For other scenarios, clusters as sparse as their anomalies indicate. F8, for example, is a random variation fault, which is associated to almost no connection between anomalous samples. Analyzing the visual aspect of the formed clusters, therefore, can lead to different insights related to the process.

On Normal Cluster Refinement

One of the key aspects of the methodology presented is how to clean the NC obtained in the first GTM + Graph Theory cycle, by retraining that data. The impact of this assessment is worthy of analysis. Figure 2.46 shows the evolution of the NC size, along with the amount of normal samples contained in it. It is important to recall that 500 samples were said to be normal.

For some scenarios, such as F1, F2 and F6, discrimination on the first cycle is already good, where further refinement eliminates aim to remove outliers whose similarity with the normal data is high. For other scenarios, the initial NC has far too many outliers, which are gradually discarded as the methodology evolves. F5, F7 and F17 in particular start with at least 800 samples in total, where gradually the NC is brought to a size where not only normal samples are still the majority, but also the number of outliers is reduced drastically.

The reason why refinement is necessary is directly connected with similarity assessment and GTM training. Considering F5, for example, the first cycle of the GTM+GT methodology failed to discriminate between normal and anomalous samples, but it managed to remove those samples whose variation was big compared to the remaining samples. By reducing the overall variation in the following sub-data sets, samples that were similar before can be now be better discriminated. Figure 2.47 shows the first and last networks for F5.

When data has significant inner variation, dissimilar groups of data might be brought together to a single cluster misguidedly, reaffirming the need of a refinement approach, which can handle better such dissimilarity assessment.

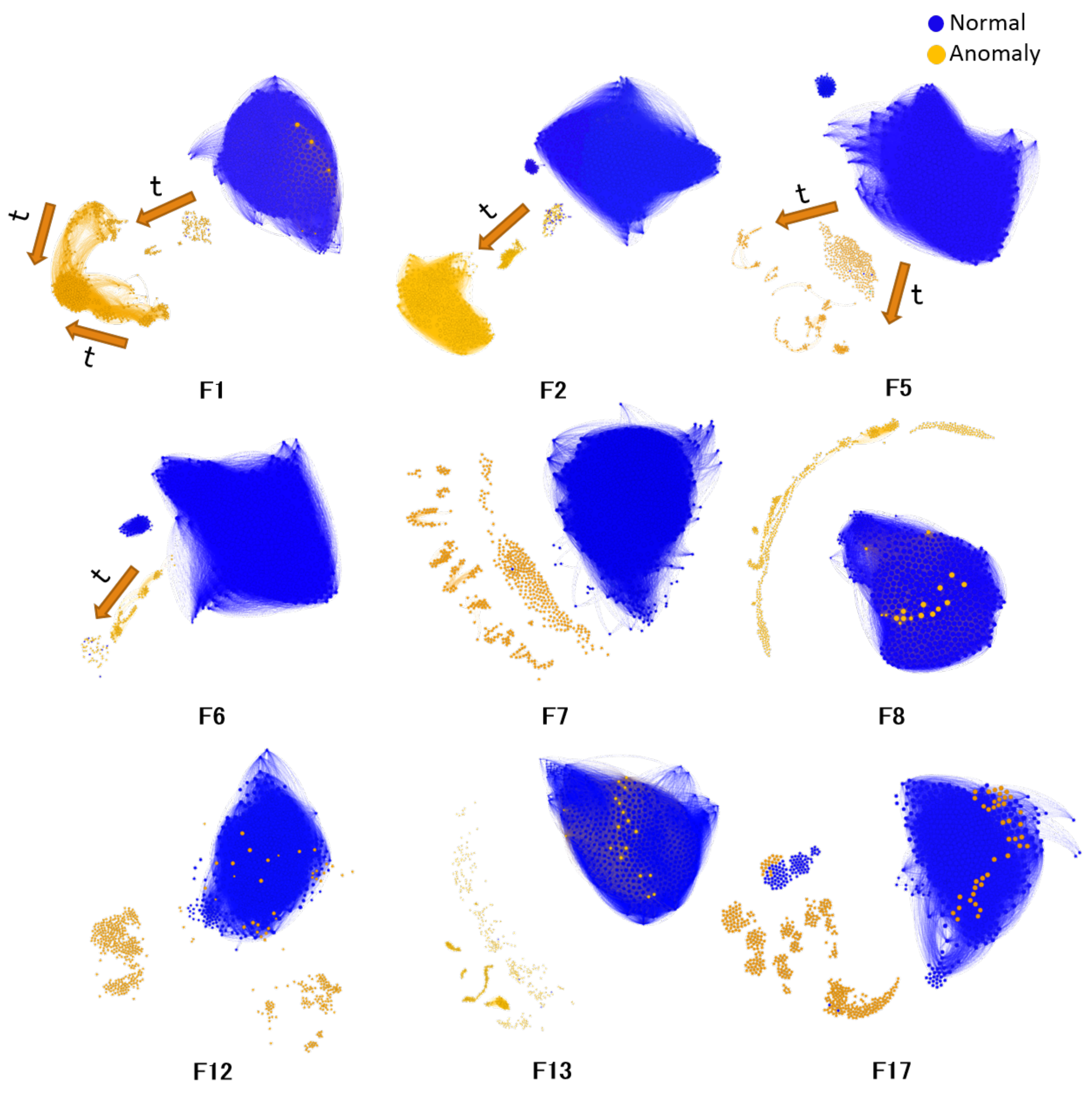

Fig. 2.44 GTM + Graph Theory networks for TEP preprogrammed faults, showing all available connections.

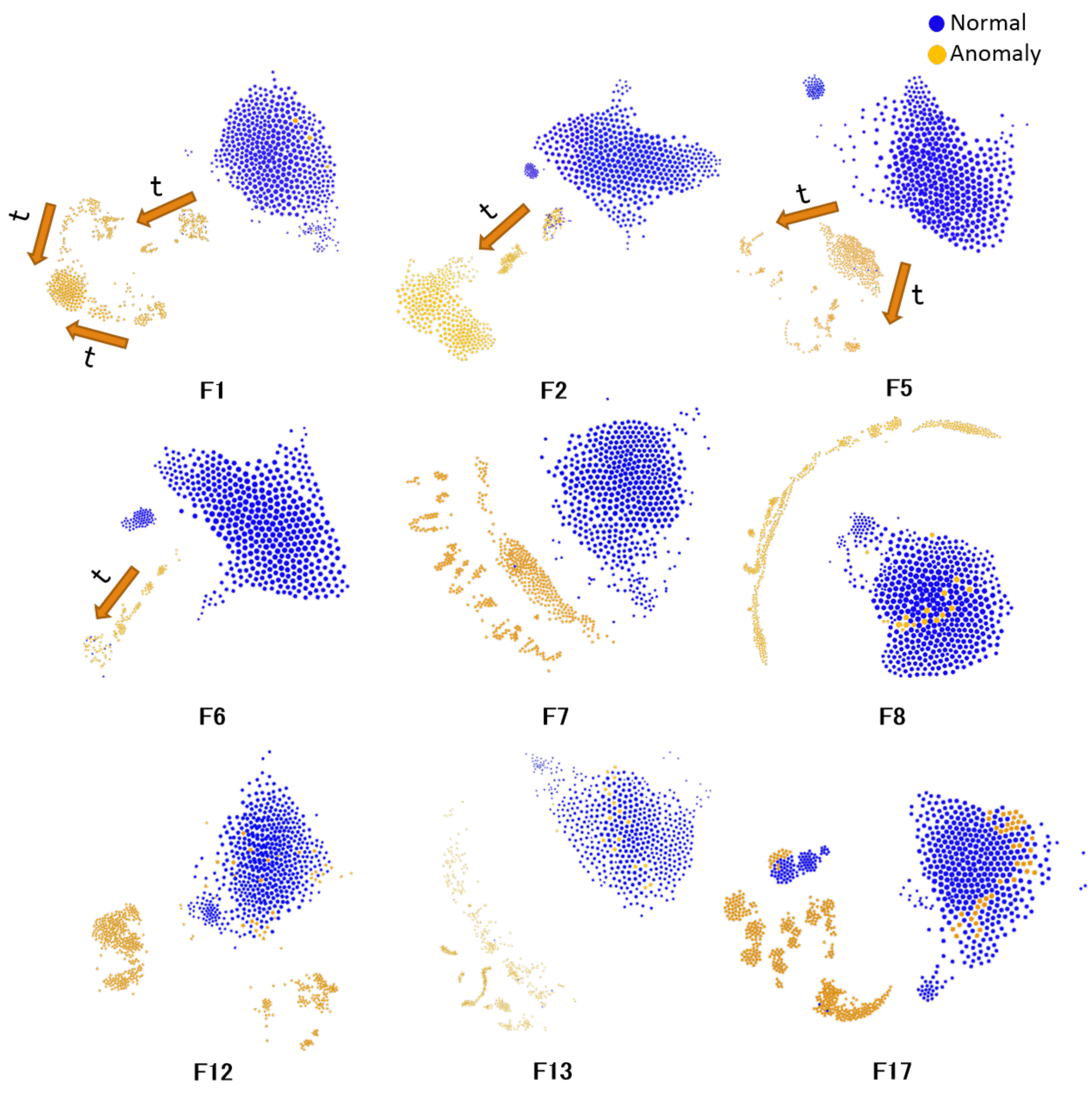

Fig. 2.45 GTM + Graph Theory networks for TEP preprogrammed faults, without edges.

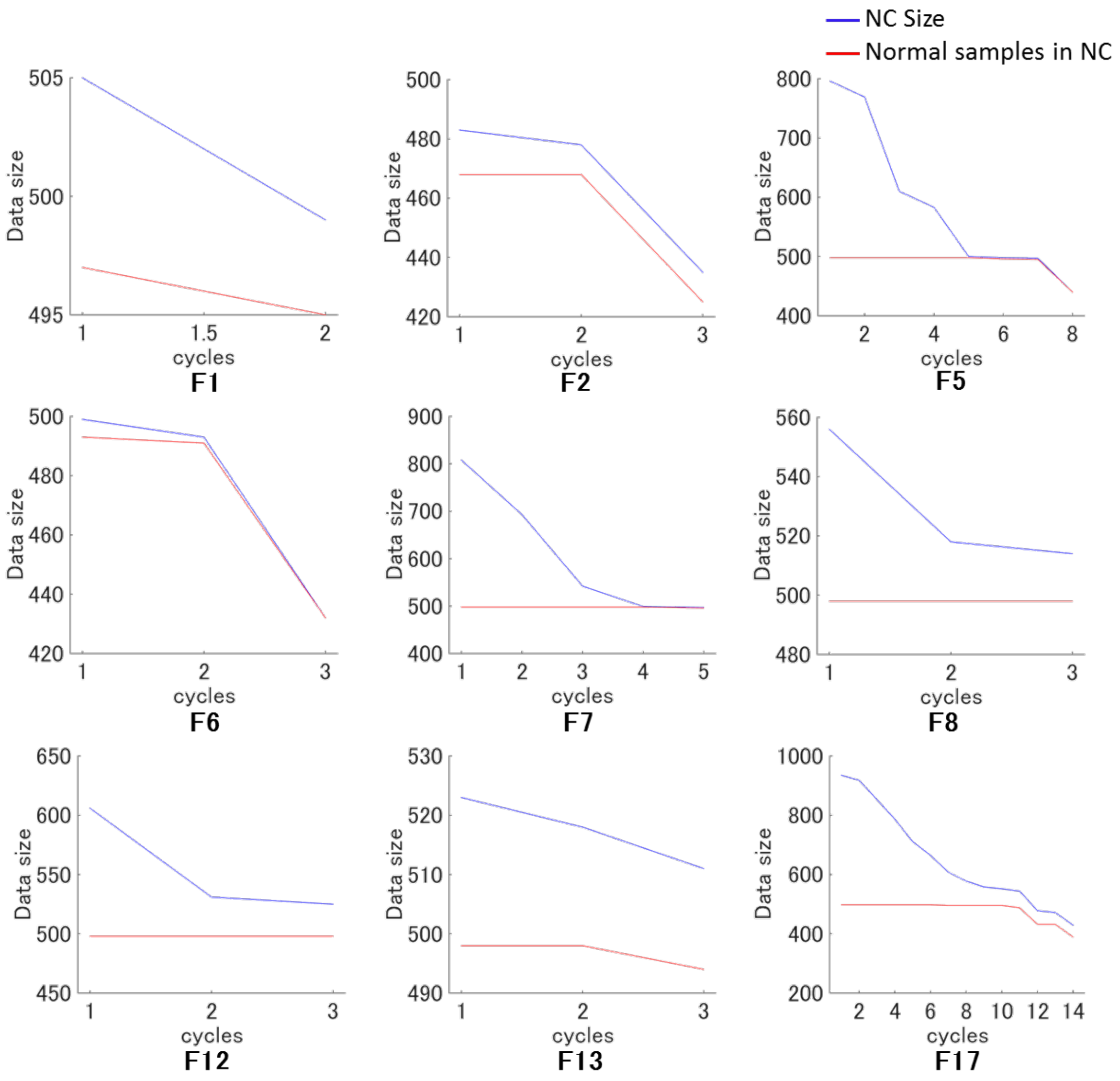

Fig. 2.46 Evolution of NC size and normal data in NC over refinement cycles

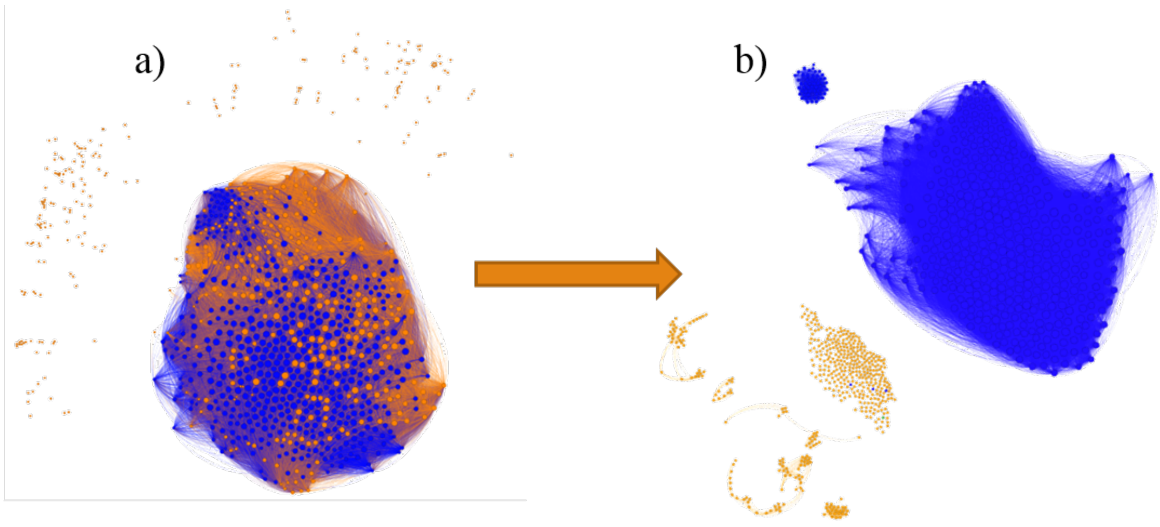

Fig. 2.47 a) First and b) last network obtained using GTM + GT proposed method for F5

2.6.3 On Data Labeling and Fault Detection

Knowing the unsupervised nature of the proposed methodology, along with the common understanding that supervised approaches should be inherently superior, the TEP case study tries to challenge this notion, by presenting results from both ends of the spectrum.

For unsupervised PCA and DPCA, Q and T^2 thresholds alone do not lead to good results, since there is no clear distinction between different data groups. Instead, k-means clustering algorithm of the PCA subspace is used for evaluation, aiming to find two distinct clusters, one normal and one anomalous. The optimal number of PC is determined when the cumulative component contribution reaches 99%. Unsupervised GTM, as mentioned before, has no potential for discrimination, since all data is used for the map training and, thus, it is not being considered. For Supervised PCA and DPCA Q and T^2 thresholds are being used for delimiting normal data regions, where values exceeding those thresholds are considered anomalous. Supervised GTM uses RE for assessing faulty samples, also relying on a RE threshold.

Initially, for the unsupervised approaches, Table 2.5 and Table 2.6 shows how PCA, DPCA and the proposed approach perform when trying to discriminate clusters with different characteristics, by showing TN and FN results. Figure 2.48 to 2.53 show Q and T^2 plots for PCA, DPCA and KPCA.

When evaluating the performance of data discrimination for unsupervised anomaly detection, two points are important. First, the majority of normal samples, used as reference since samples' true label is unknown, should belong to the NC. Second, as few anomalies as possible should be contaminating the NC. Table 2.5 and Table 2.6 summarize those results,

Table 2.5 TN in percentage for unsupervised approaches.

	F1	F2	F5	F6	F7	F8	F12	F13	F17
GTM+GT	99.4	87.4	88.6	86.4	99.8	100	100	99.2	85.8
Uns.PCA	100	100	100	100	100	100	100	100	80.8
Uns.DPCA	100	100	100	100	100	100	100	100	81.9
Uns.KPCA	100	100	100	100	100	100	100	100	82.3
Uns.GTM	100	100	100	100	100	100	100	100	100

Table 2.6 FN in percentage for unsupervised approaches.

	F1	F2	F5	F6	F7	F8	F12	F13	F17
GTM+GT	0.6	0	0	0	0	3.4	6.4	3.0	5.6
Uns.PCA	71.2	22.9	85.2	16.0	81.0	64.0	67.7	75.6	58.3
Uns.DPCA	50.8	23.1	85.4	16.2	80.6	74.4	67.3	79.2	58.1
Uns.KPCA	40.3	18.7	60.4	10.7	81.3	55.7	59.3	62.4	57.6
Uns.GTM	$\sim$100	$\sim$100	$\sim$100	$\sim$100	$\sim$100	$\sim$100	$\sim$100	$\sim$100	$\sim$100

revealing that the proposed methodology outperforms PCA and DPCA for all scenarios. As an unsupervised methodology for discrimination, relying on k-means is at best naive, since knowing with certainty the number of clusters existent in the system is unlikely. Even so, for F2 and F6, two step faults, discrimination was fair, with roughly 20% of the data contaminating the samples. The proposed approach, on the other hand detected pretty well step faults, with close to perfect discrimination. For all other anomalies, discrimination was still high, even with some contamination.

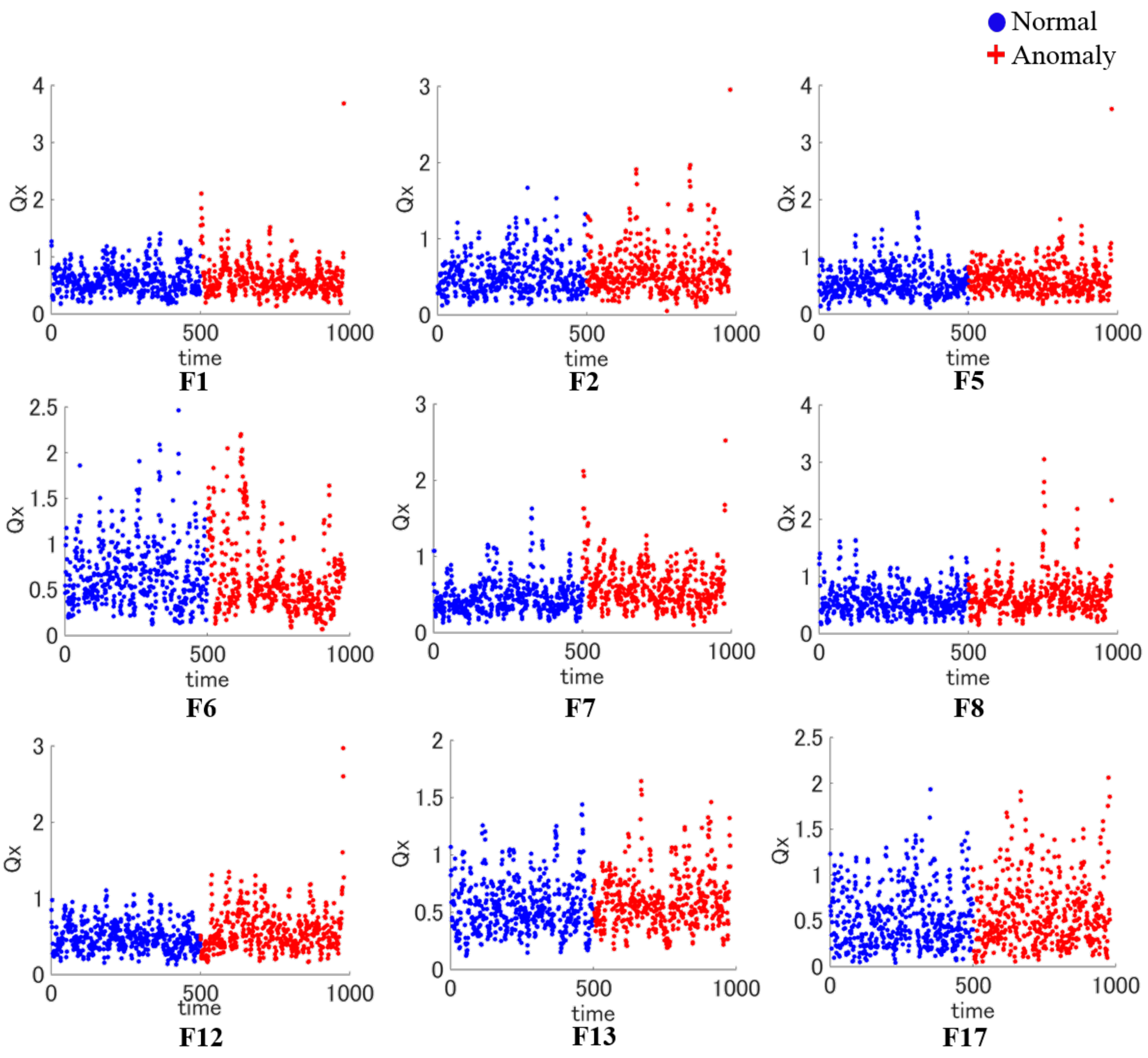

Fig. 2.48 Unsupervised PCA Q plots for TEP.

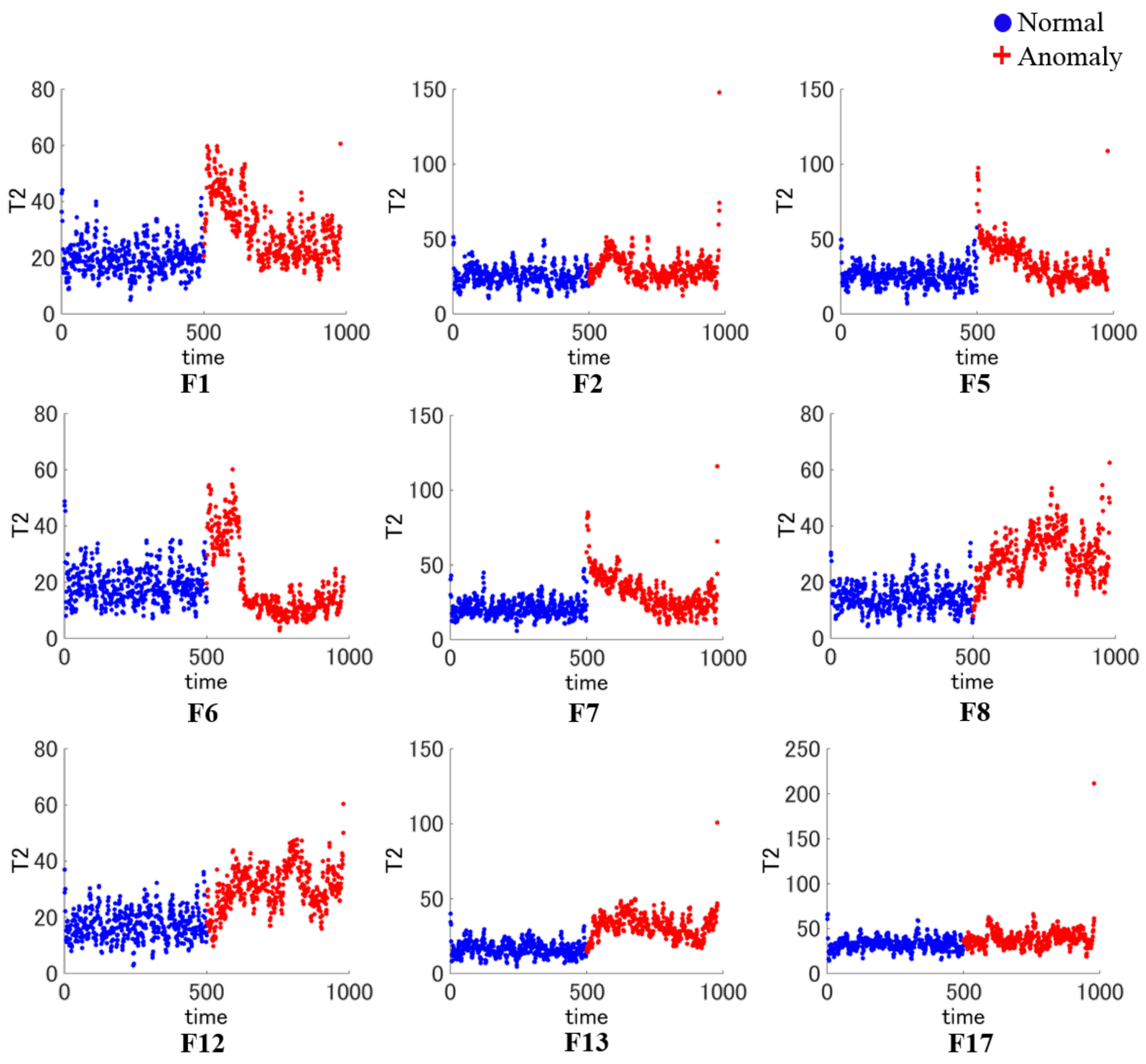

Fig. 2.49 Unsupervised PCA T^2 plots for TEP.

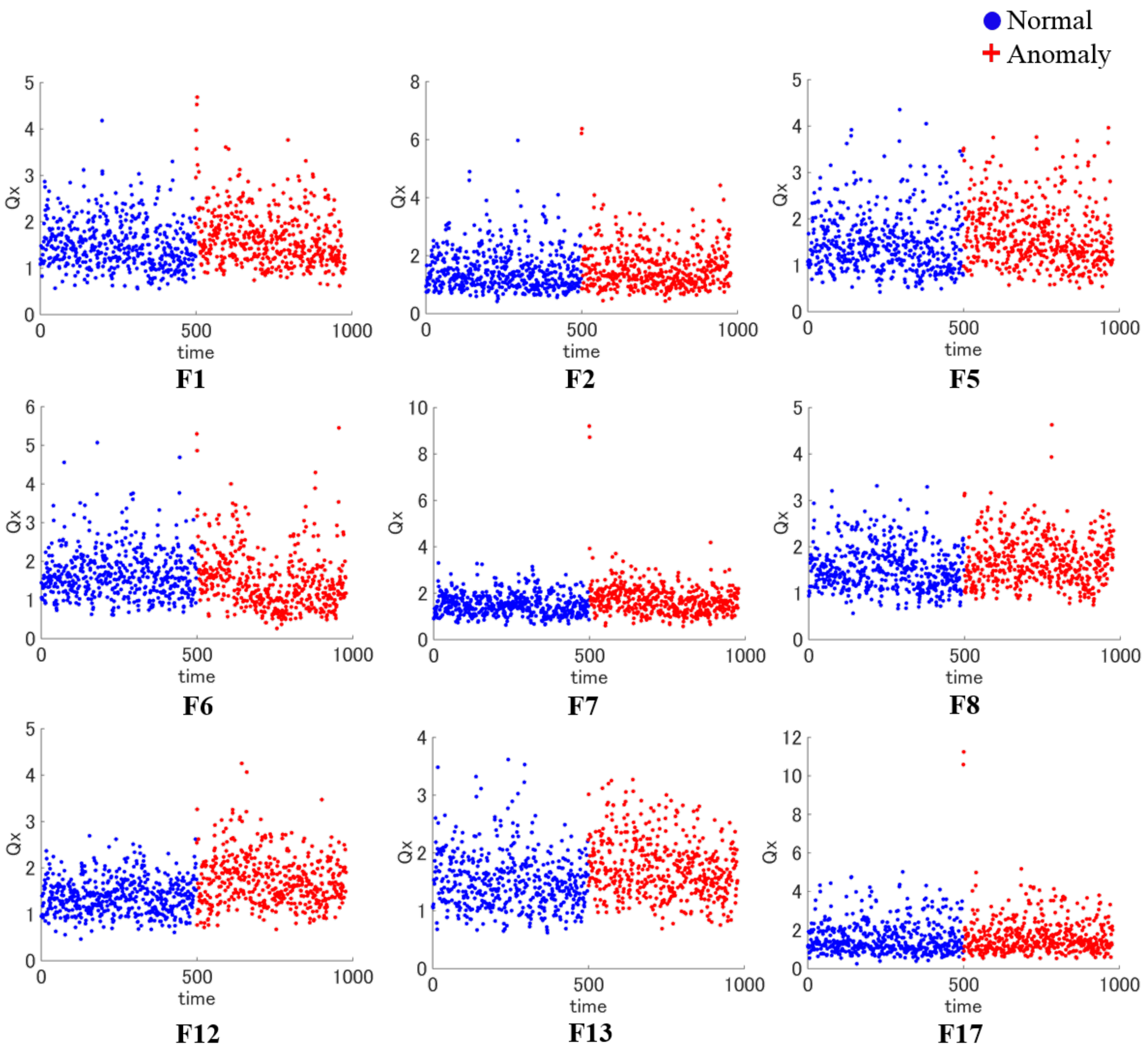

Fig. 2.50 Unsupervised DPCA Q plots for TEP.

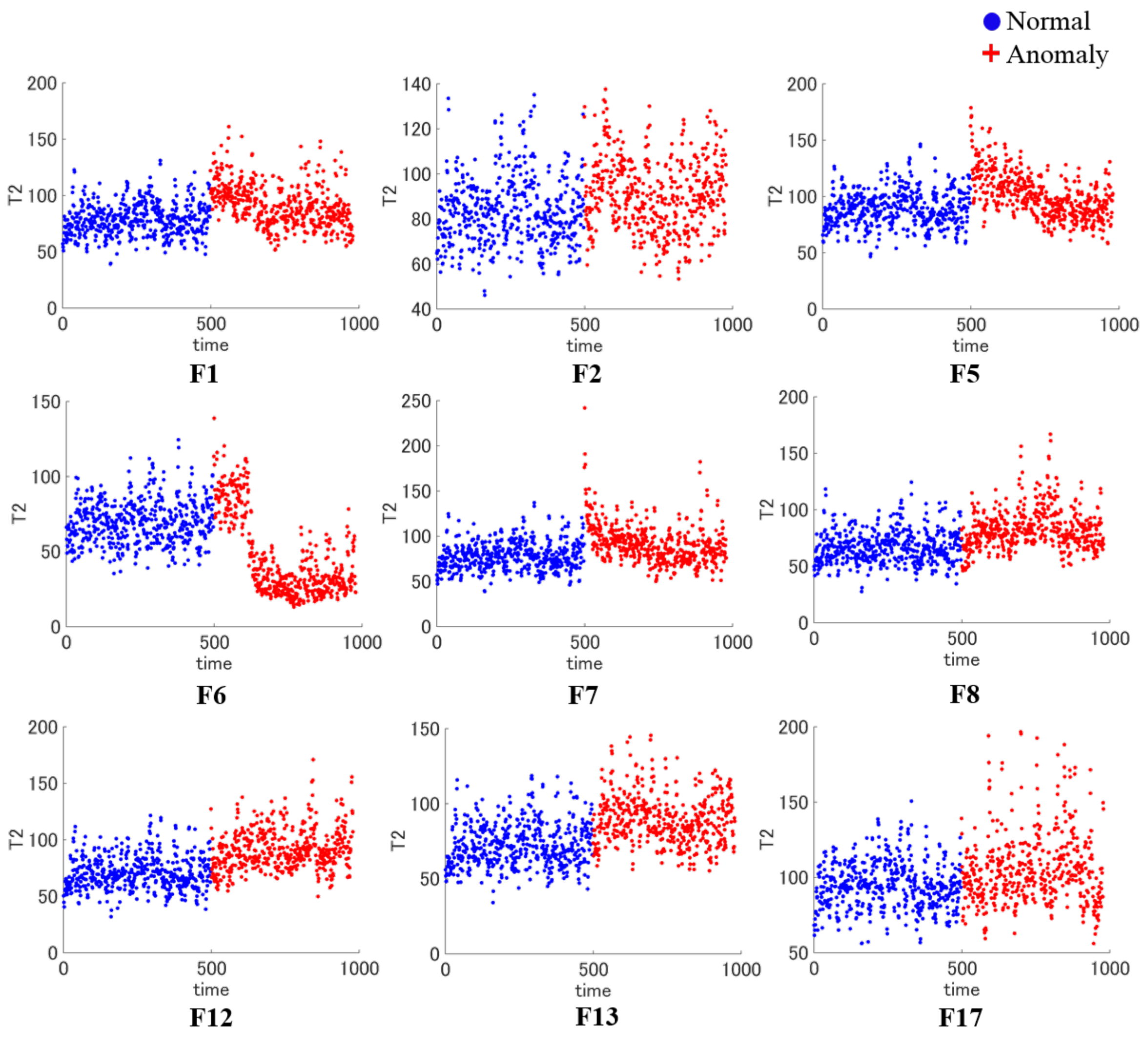

Fig. 2.51 Unsupervised DPCA T^2 plots for TEP.

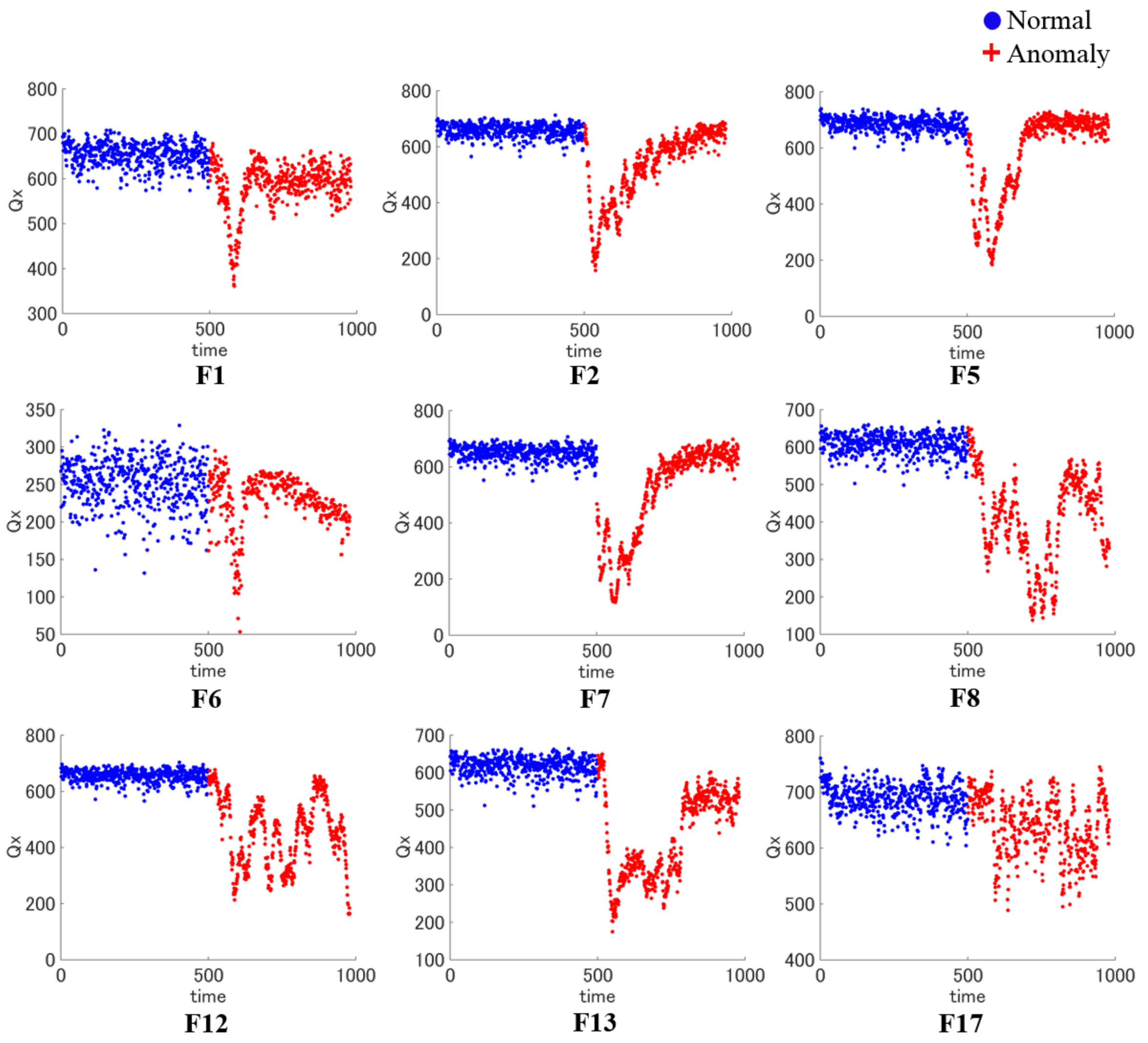

Fig. 2.52 Unsupervised KPCA Q plots for TEP.

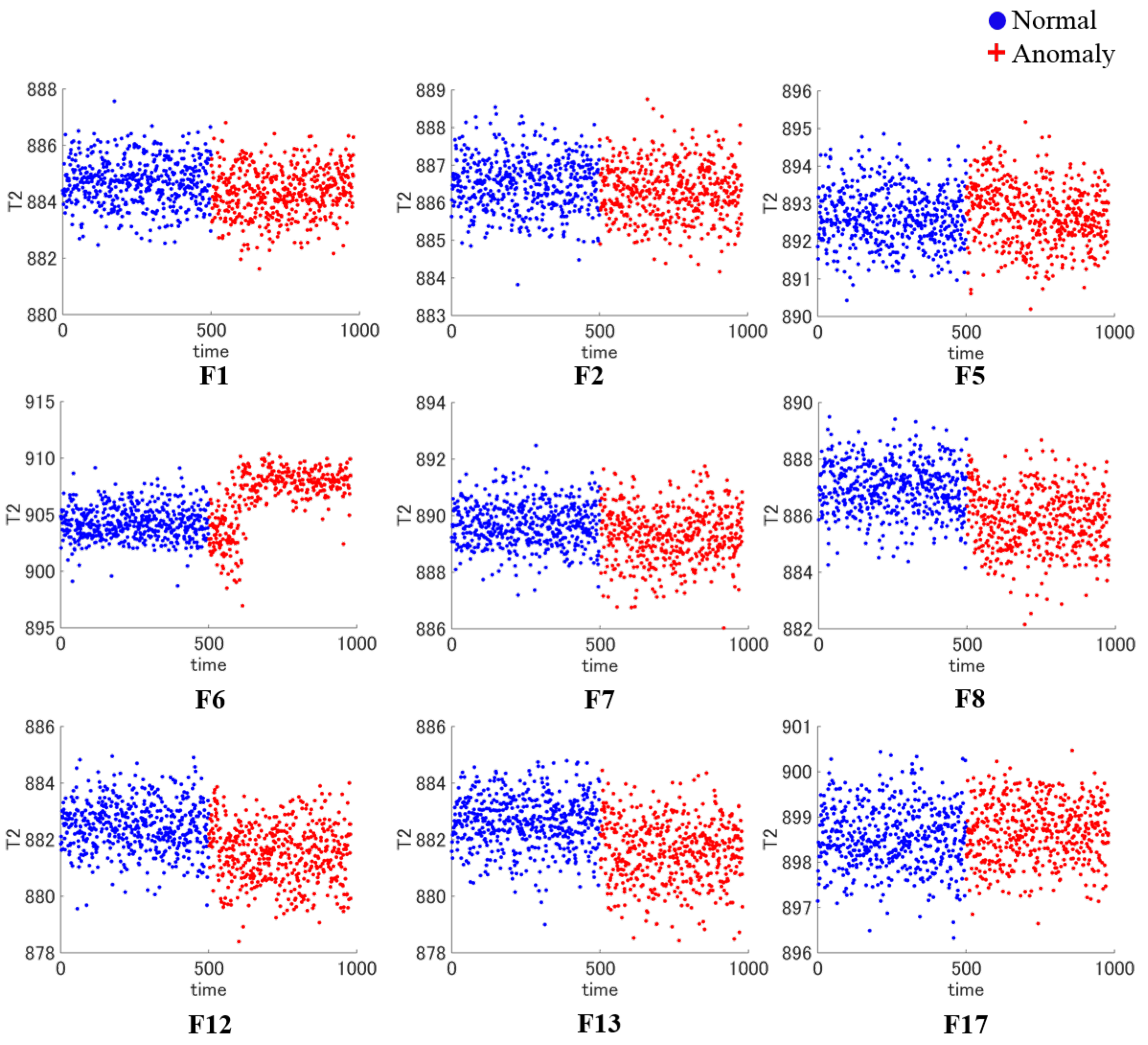

Fig. 2.53 Unsupervised KPCA T^2 plots for TEP.

Table 2.7 shows the incidence of undetected outliers for all supervised approaches tested and the proposed unsupervised approach. Since normal data is known beforehand, there is no need to assess the amount of normal samples undetected. Figures 2.54 to 2.59 show Q and T^2 plots for supervised PCA, DPCA and KPCA and Figures 2.60 and 2.61 shows RE and similarity plots for supervised GTM.

Among the supervised techniques, DPCA had the best performance overall, where despite poor detection on F17, the remaining scenarios had good detection performance. PCA, KPCA and GTM struggled to discriminate correctly F5. The RE plot for this fault shows that part of the anomalous data adheres well to the map, which is a sign of similarity with the normal data. As mentioned in the previous section, the proposed methodology can handle such issue, by retraining NC data until the inner variation of the data set is small enough to discriminate seemingly similar, yet dissimilar, data groups.

The unsupervised proposed methodology here performed as well as the best other methodologies presented. This reinforces the use of GTM+GT as a tool for both data visualization and fault detection.

Table 2.7 FN in percentage for supervised approaches and the proposed methodology.

	F1	F2	F5	F6	F7	F8	F12	F13	F17
GTM+GT First	1.6	2.2	58.4	0.8	40.2	1.4	7.6	3.0	32.2
GTM+GT Last	0.6	0	0	0	0	3.4	6.4	3.0	5.6
Sup. PCA	0.42	2.50	49.2	0	0	2.5	5	3.12	11.25
Sup. DPCA	0	2.08	0	0	0	1.87	1.87	2.08	12.92
Sup. KPCA	0.80	1.81	41.6	1.20	0	2.81	4.22	3.01	12.05
Sup. GTM	0	2.08	51.7	0	0	2.71	3.75	3.12	10

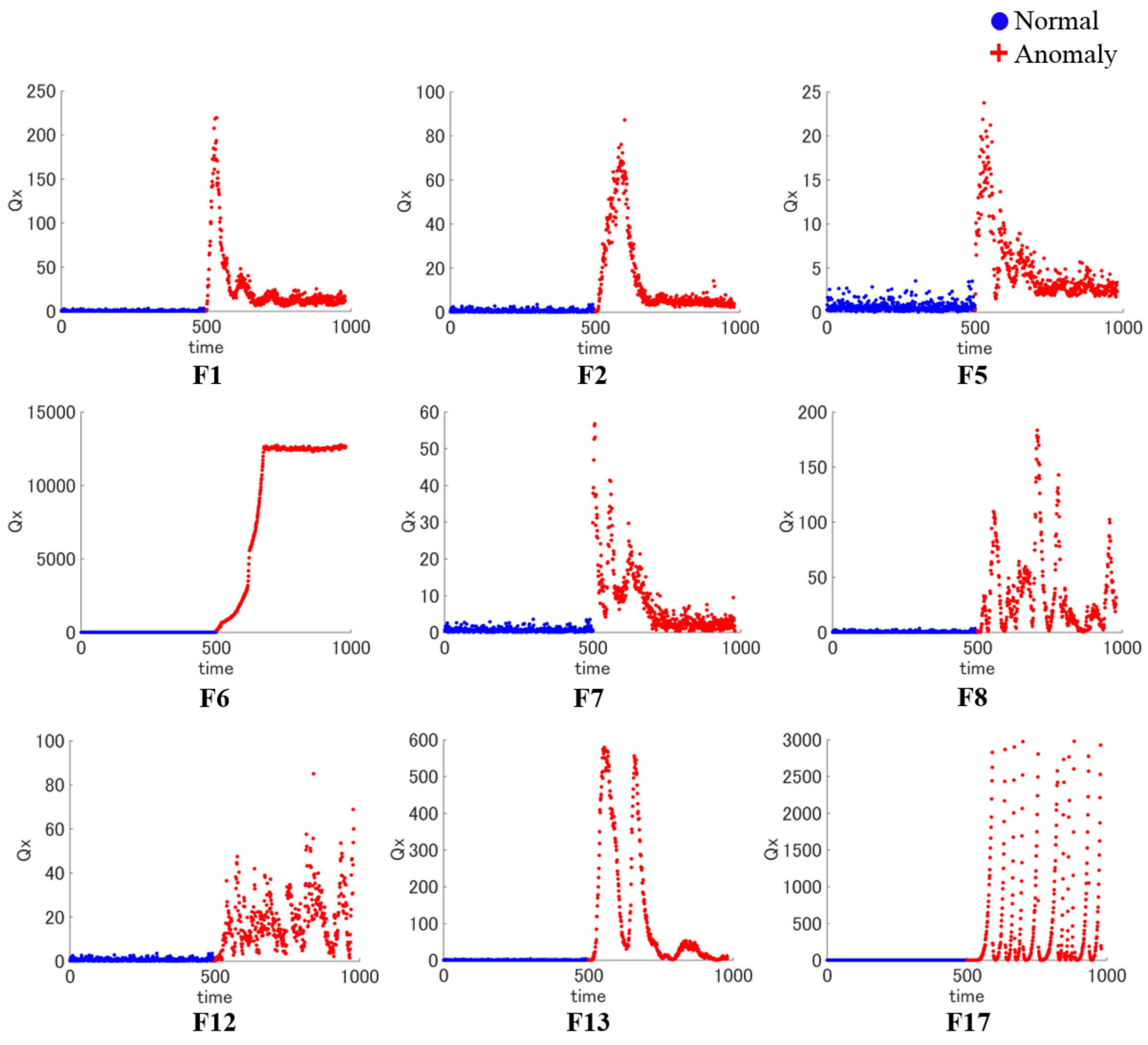

Fig. 2.54 Supervised PCA Q plots for TEP.

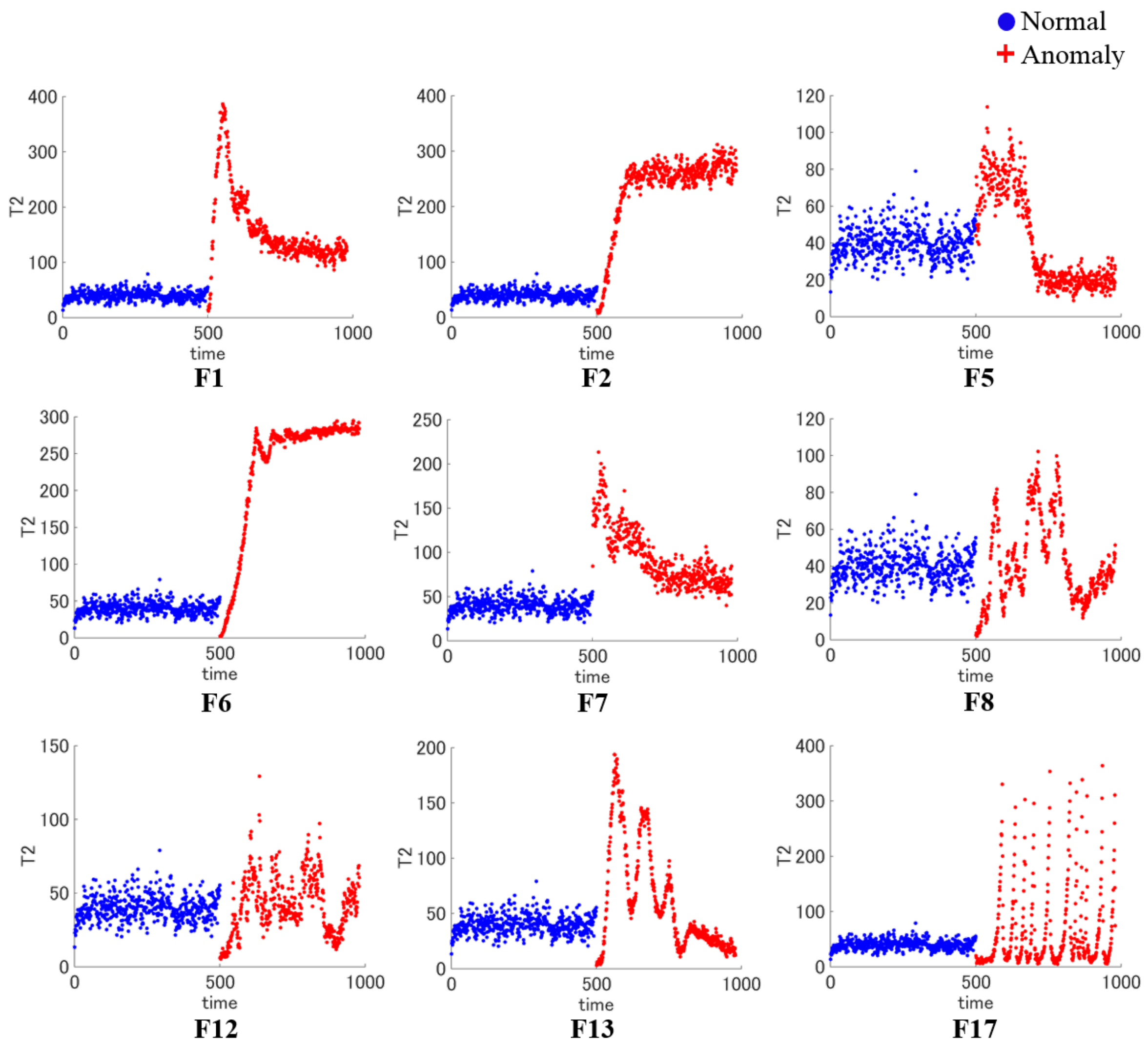

Fig. 2.55 Supervised PCA T^2 plots for TEP.

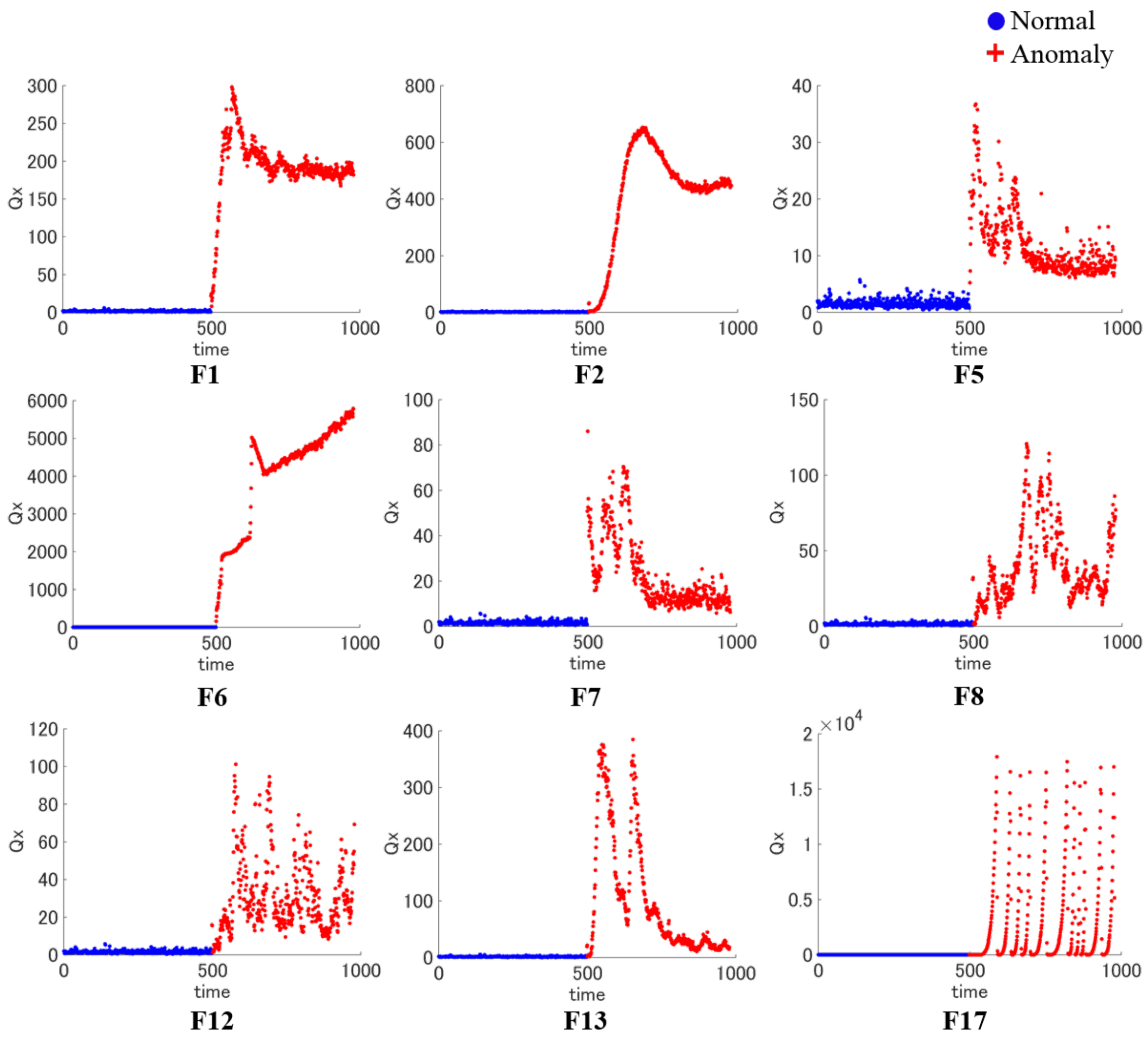

Fig. 2.56 Supervised DPCA Q plots for TEP.

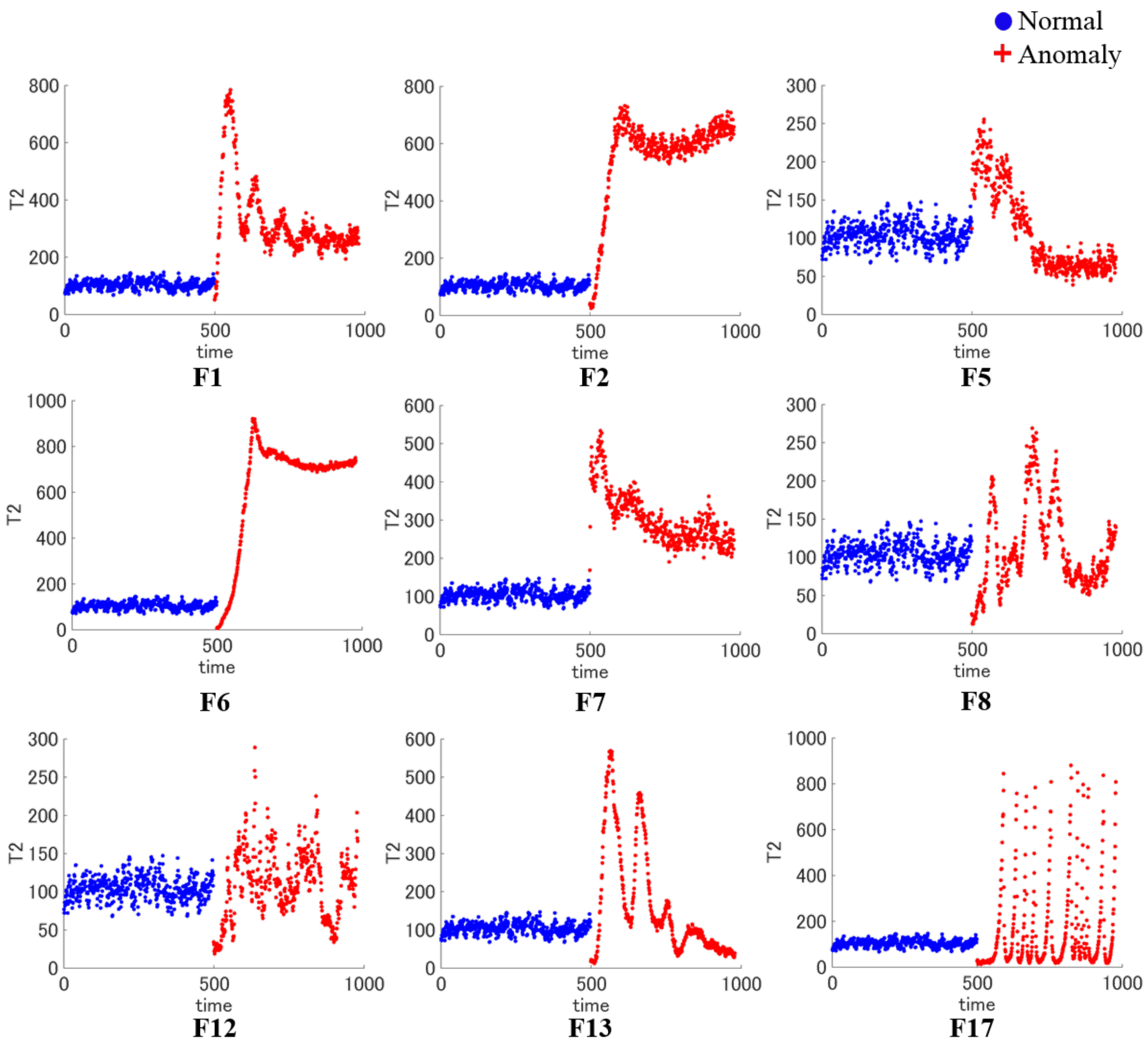

Fig. 2.57 Supervised DPCA T^2 plots for TEP.

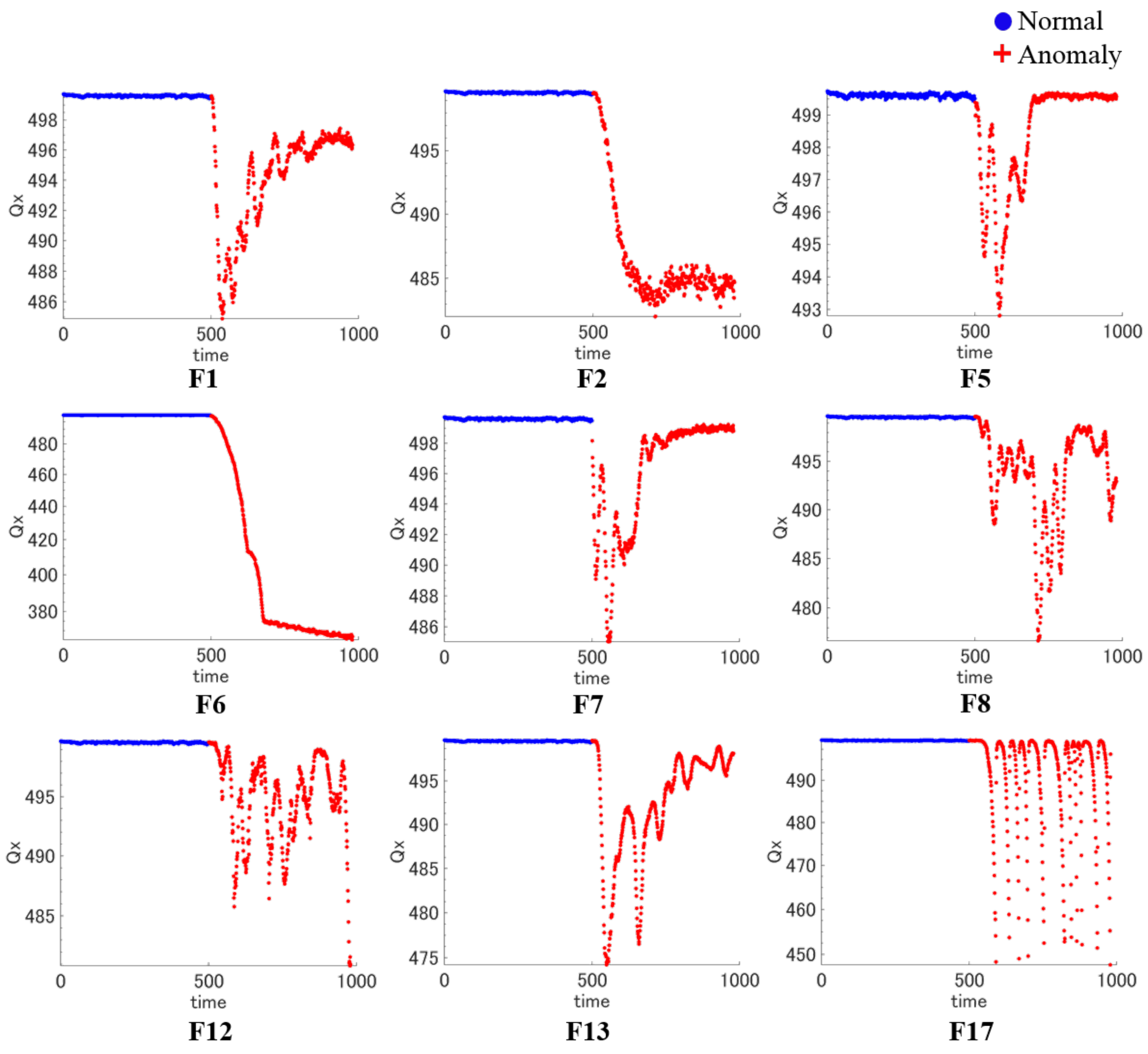

Fig. 2.58 Supervised KPCA Q plots for TEP.

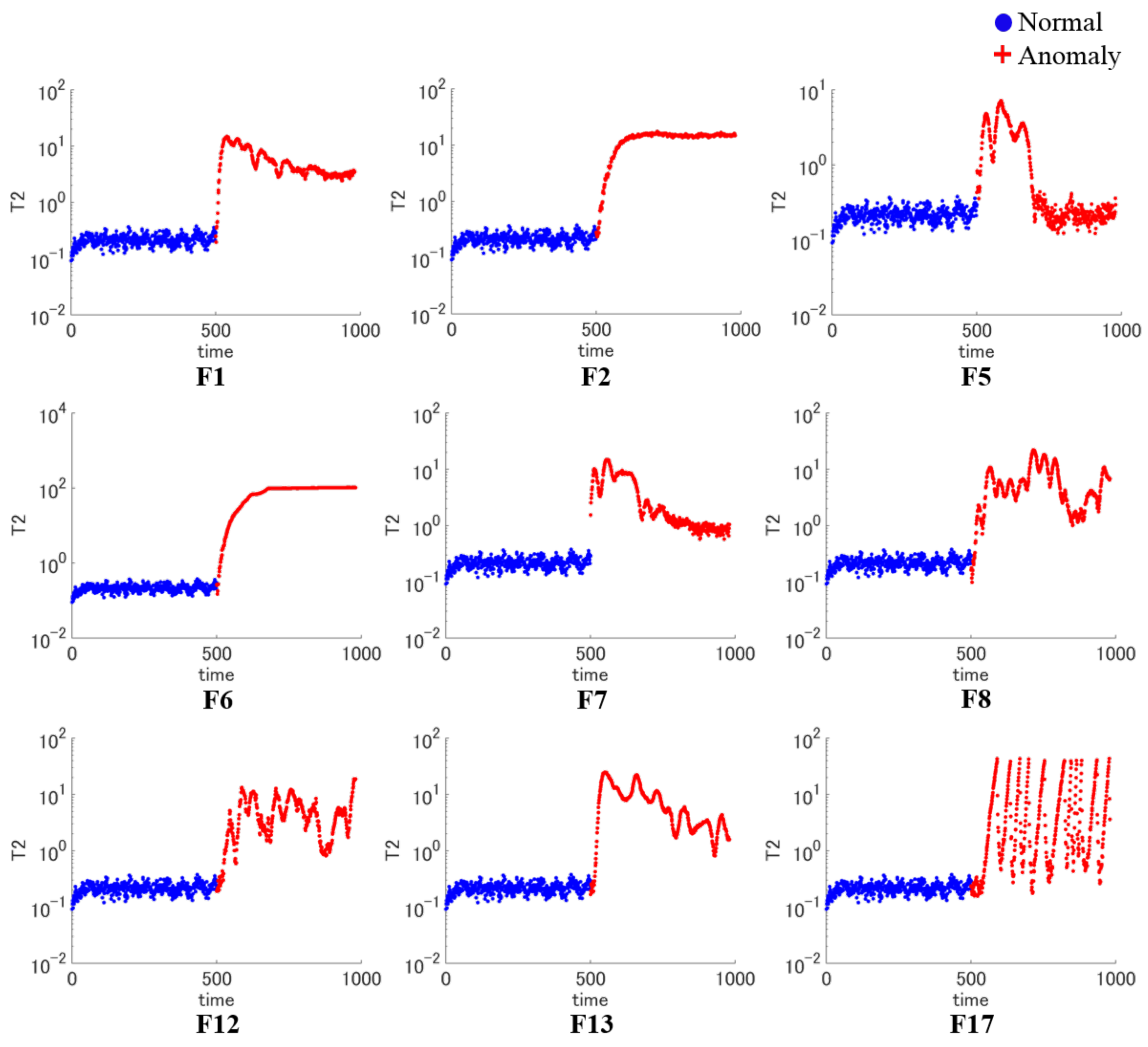

Fig. 2.59 Supervised KPCA T^2 plots for TEP.

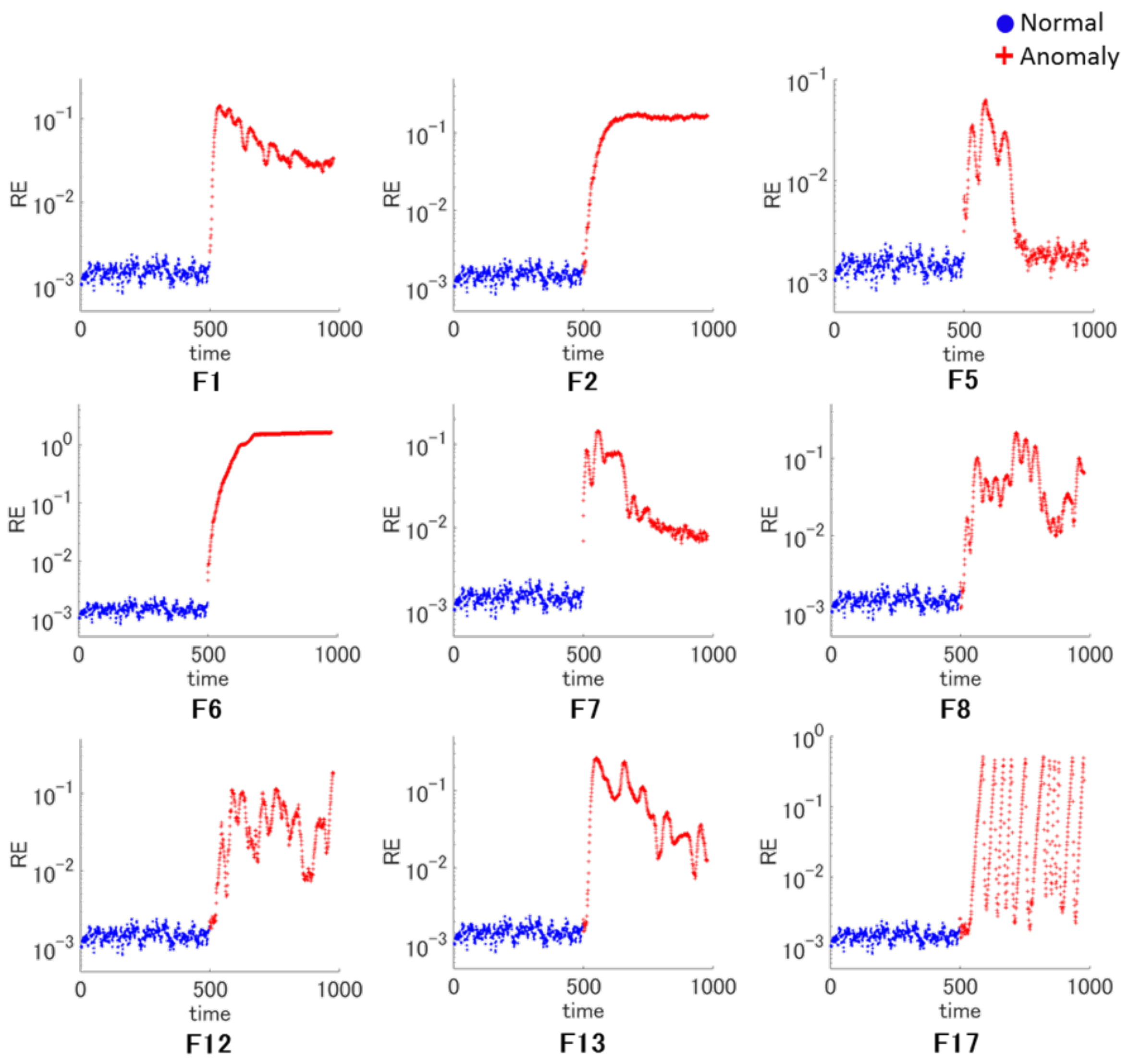

Fig. 2.60 RE plots for supervised GTM.

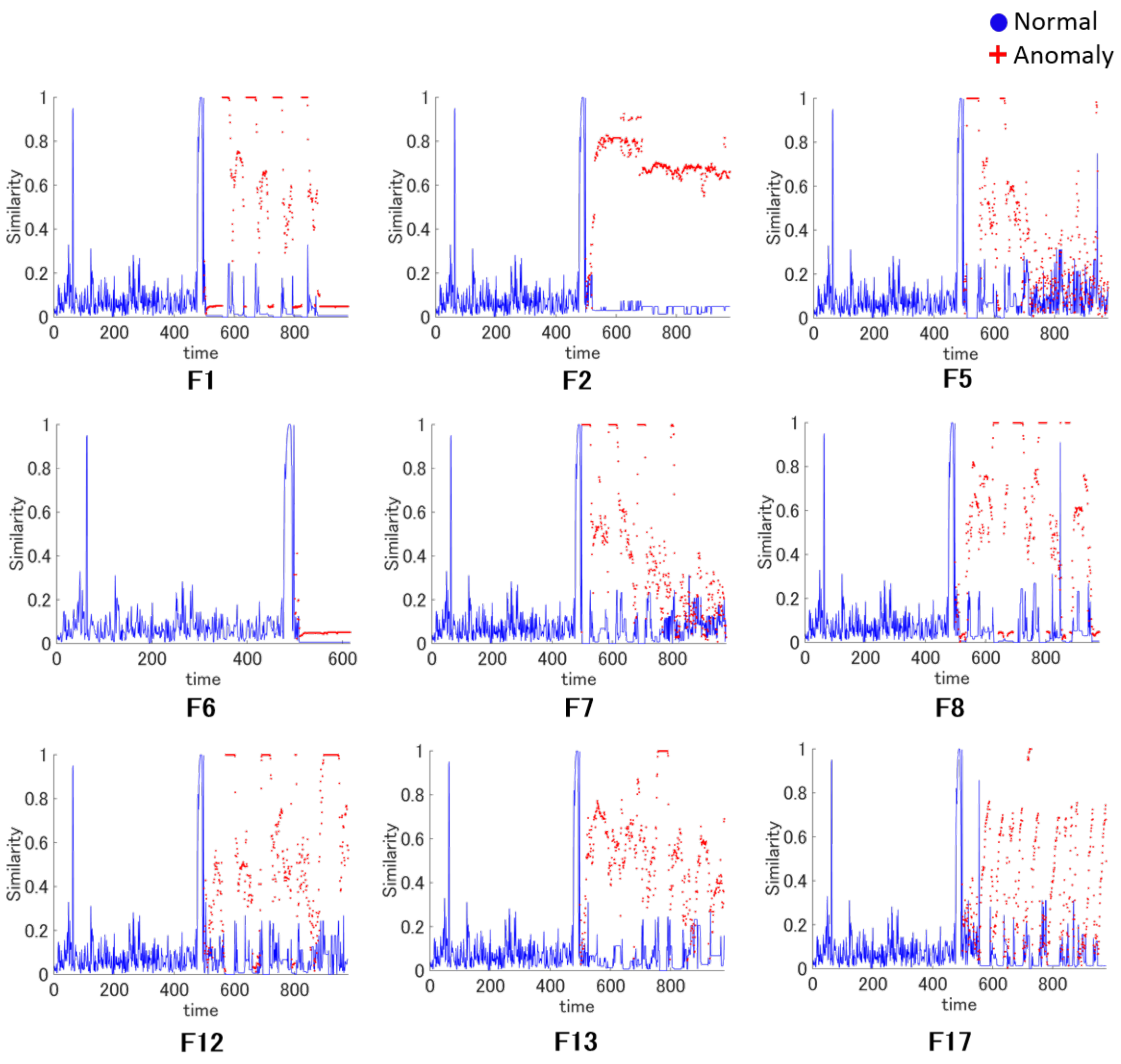

Fig. 2.61 Similarity plots for supervised GTM.

2.6.4 On Online Monitoring

Similarly to what was presented in section 2.5.2, discussing the proposed strategy is important for assessing its main benefits and drawbacks. Based on the data obtained in the literature[47], test data sets are available for all preprogrammed faults. The source of each fault is the same for training and test data sets. 160 normal samples are initially displayed, followed by 960 anomalous instances. Based on the results presented in section 2.6.3, DPCA was the supervised methodology with the best performance. Figures 2.62 and 2.63, therefore, show DPCA results for each fault. Figure 2.64 shows the inferred labels using DPCA.

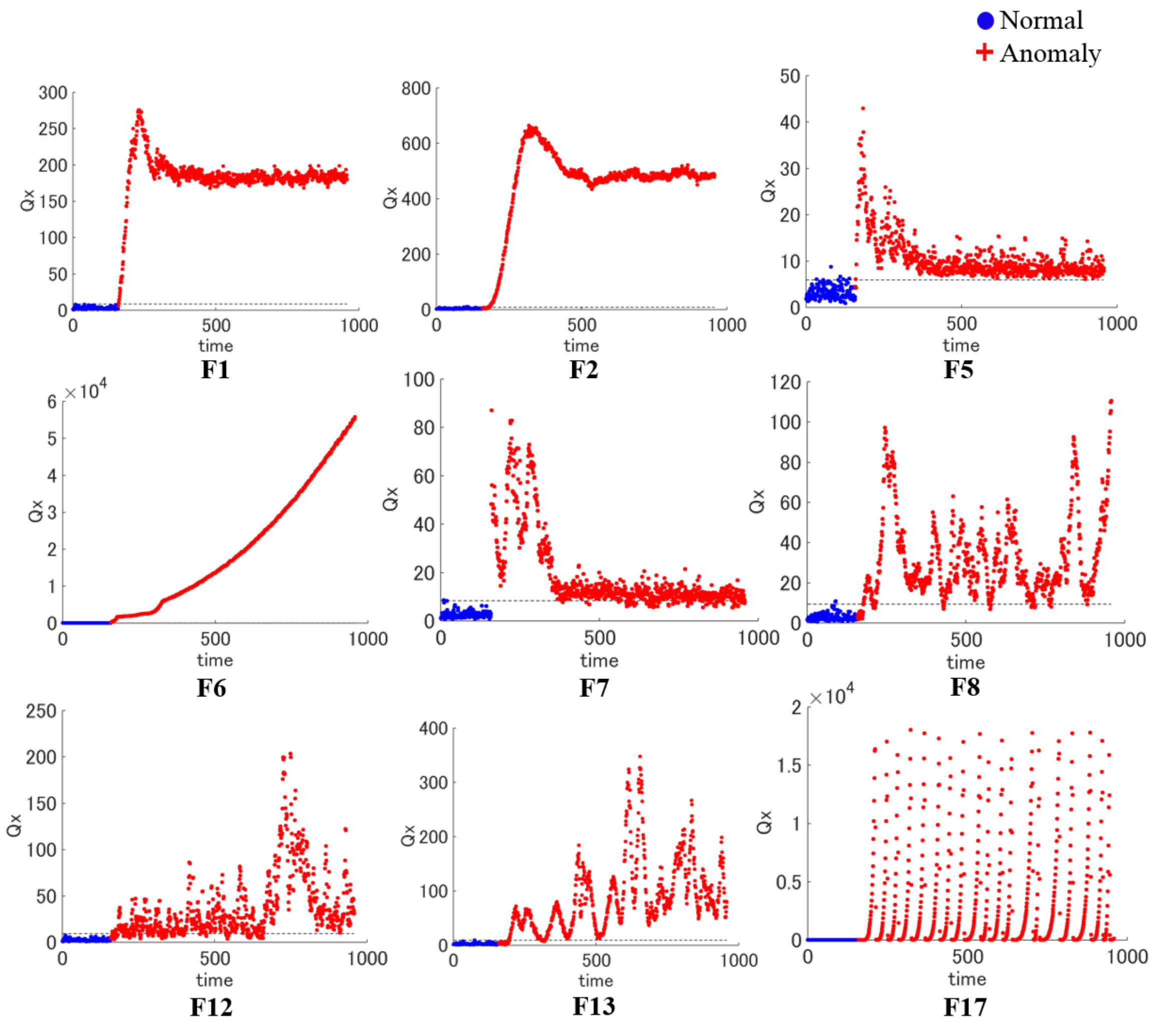

Fig. 2.62 DPCA *Q* monitoring results for test data.

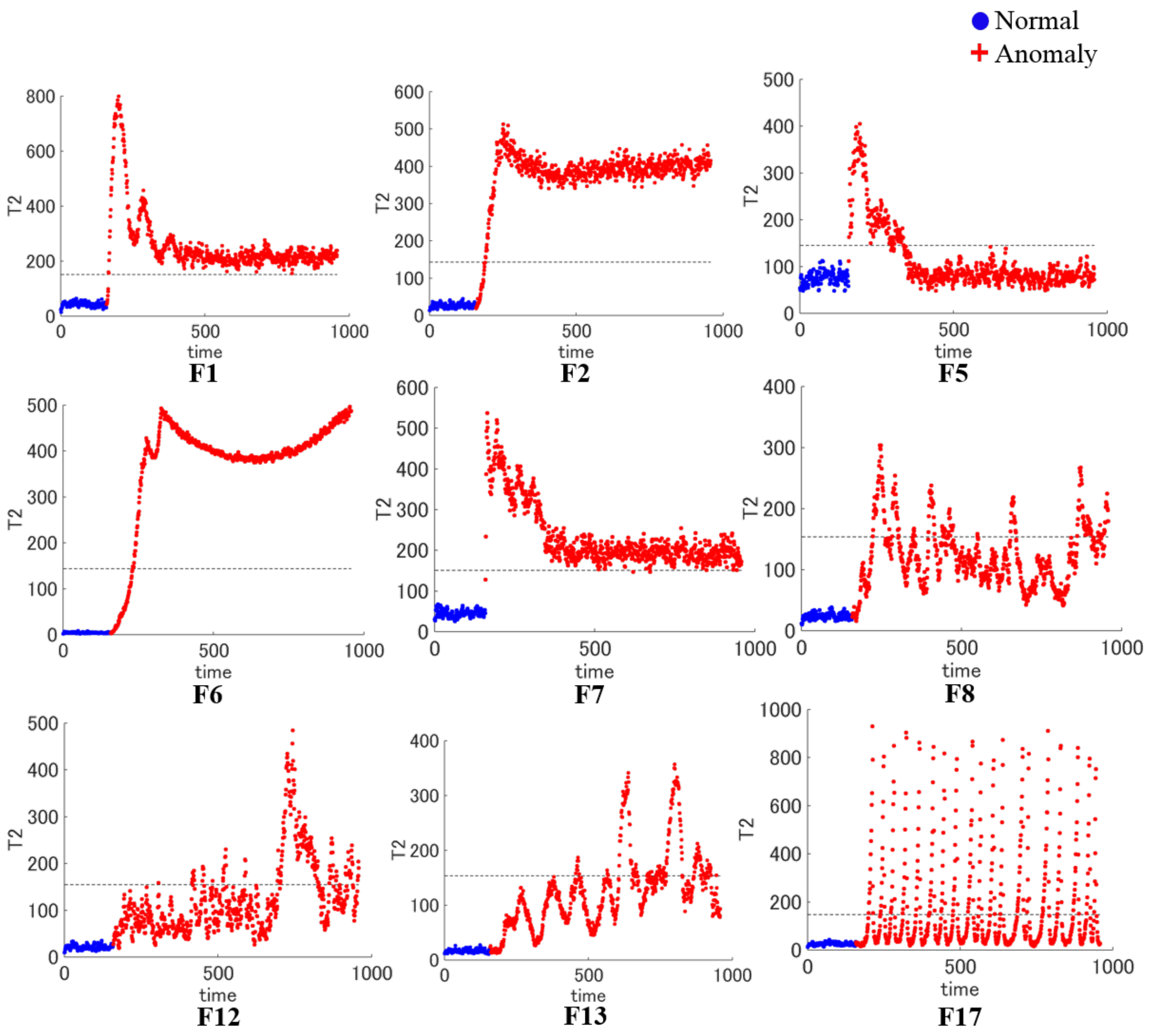

Fig. 2.63 DPCA *Q* monitoring results for test data.

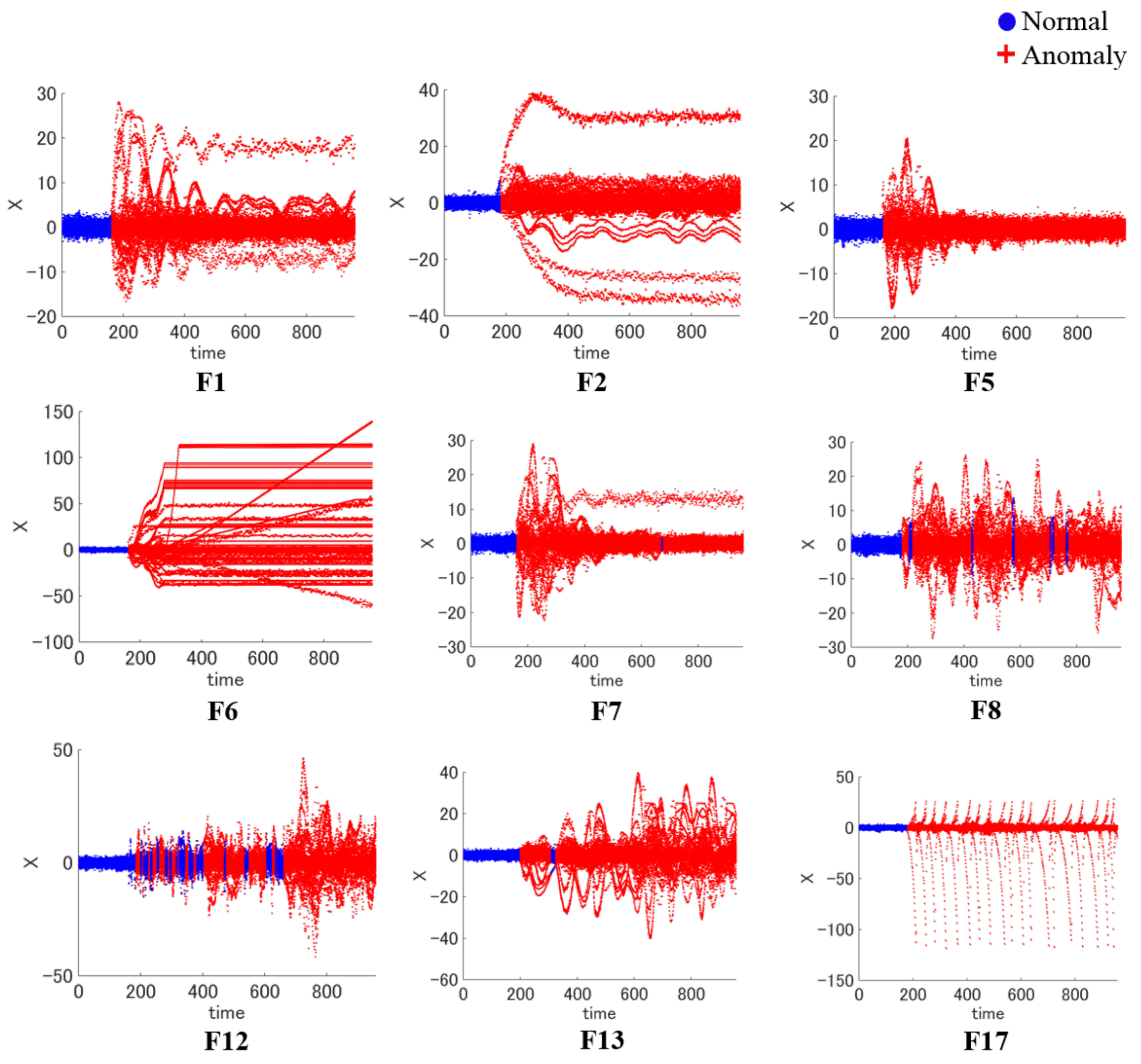

Fig. 2.64 DPCA normal and anomalous inferred labels for test data.

For most scenarios, detection was good, with little delay or no delay at all. Table 2.8 shows the detection delay for all cases, keeping in mind that FN and FP results are presented in percentages. Since the amount of normal is small, FN values are significantly larger than the ones presented in Table 2.7 Despite the fact that for F2 and F13, for example, there was some delay involved, those delays are small when compared to the overall evolution of anomalies.

Table 2.8 DPCA Detection delay, FN and FP for test data.

	F1	F2	F5	F6	F7	F8	F12	F13	F17
Detection Delay (Samples)	1	20	1	1	0	19	5	35	17
FN (%)	0.62	12.5	0.62	0.62	0	11.87	3.12	21.87	10.62
FP (%)	0	1.89	3.80	10.13	0.12	1.26	0.60	0.63	13.29

F8 and F12 presented poor detection, where lots of anomalies were mislabeled as normal instances, especially because mislabeling was recurrent. F8 and F12 are random faults, meaning that their behavior is erratic, and as such it can come back to the normal state momentarily. Random behavior can be very difficult to predict and if both range and dynamics match the one in the normal state, mislabeling will occur.

Another aspect to be considered is that ideally one would assume that the NC contains only significant normal samples, but contamination is a reality, affecting the similarity assessment of future samples. Considering the results presented in Table 2.7, F8 and F12 have a fair amount of contamination in their NC. The presence of such anomalous instances might lead to other anomalies correlating to those particular samples. It is important to notice, however, that due to the nature of those faults and GTM+GT's methodology, it is more likely that those samples have little influence in the overall results. In the end, the application of GTM+GT as a tool for online process monitoring is viable and it showed good performance, even though depending on the specific fault nature, detection performance might suffer some degradation.

2.7 Case Study - Exhaust Gas Denitration Process

The final case study explores an Exhaust Gas Denitration Process at Mitsui Chemicals, Inc. NH_3 is injected into a denitration reactor, where exhaust gas passes through a catalytic layer, and NO_x is decomposed into N_2 and water vapor. The reactions are established as described below:

$$4NO + 4NH_3 + 2O_2 \rightarrow 4N_2 + 6H_2O \tag{2.38}$$

$$NO + NO_2 + 2NH_3 \rightarrow 2N_2 + 3H_2O \tag{2.39}$$

The interest (output) variables in this system are the denitration outlet NH_3, denitration outlet NO_x and outlet gas duct NO_x. Among all variables available for measurement, 37 were used for assessing normal and anomalous states. Two distinct case studies were considered, each with 2000 samples, where both normal and anomalous data coexist. The labeling was provided by operators *in locu*, who judged which samples would be normal and anomalous. By referring to those labels and the results obtained from different methodologies, we aim to assess, besides fault detection itself, the reliability of such labeling. Table 2.9 presents extra information on each case study.

Table 2.9 Case study description.

	#Normal Samples	#Anomalous samples
Case A	1200	800
Case B	1430	570

Figures 2.65 to 2.68 shows Case A and B results, respectively, for PCA, DPCA, Kernel PCA and GTM. For Case A, regardless of the technique used, it is clear that even for supervised approaches discrimination is poor if one assumes correct labeling. Supervised DPCA, Kernel PCA and GTM all fail to exclude the beginning of the anomaly, which might be an indication that the label defined by operators is only roughly discriminating between distinct states. Case B also shows a poor discrimination, where only supervised GTM has slightly better results. Unsupervised approaches still perform poorly, as expected, but GTM can discriminate to a certain extent. Similarity, however, is not really effective on normal and anomalous data, where Q shows a more promising potential.

For unsupervised GTM, using similarity for discrimination is difficult, where the inner similarities for each data group results in similar maximum values. For the supervised approach, even though the visualization of the concept is difficult, its logic is more coherent. A varying threshold reflects how different samples can relate to the normal data set, even if their similarity is not as high.

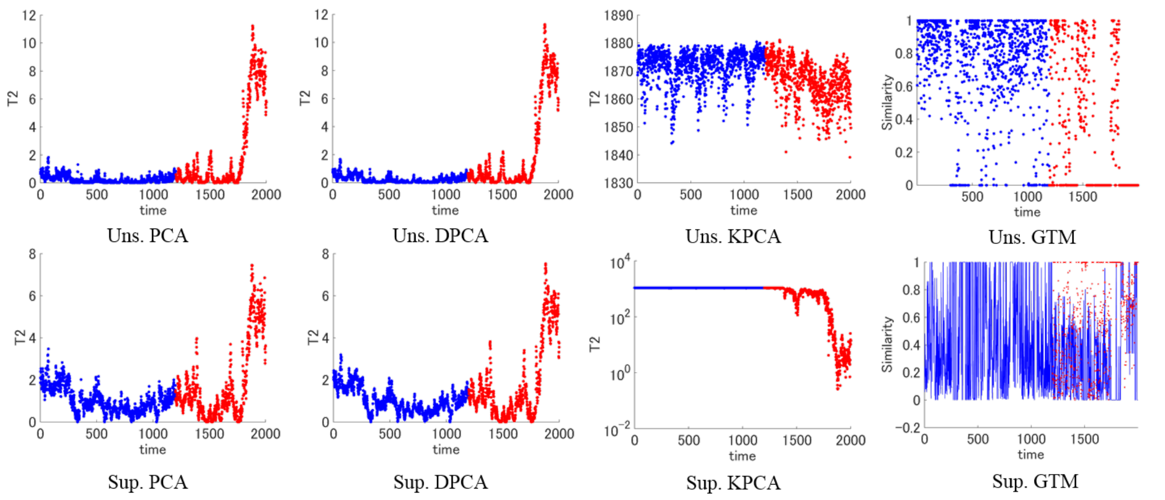

Fig. 2.65 Case A PCA, DPCA, KPCA T^2 and GTM similarity monitoring results for both unsupervised and supervised results, where blue samples are normal and red, anomalous.

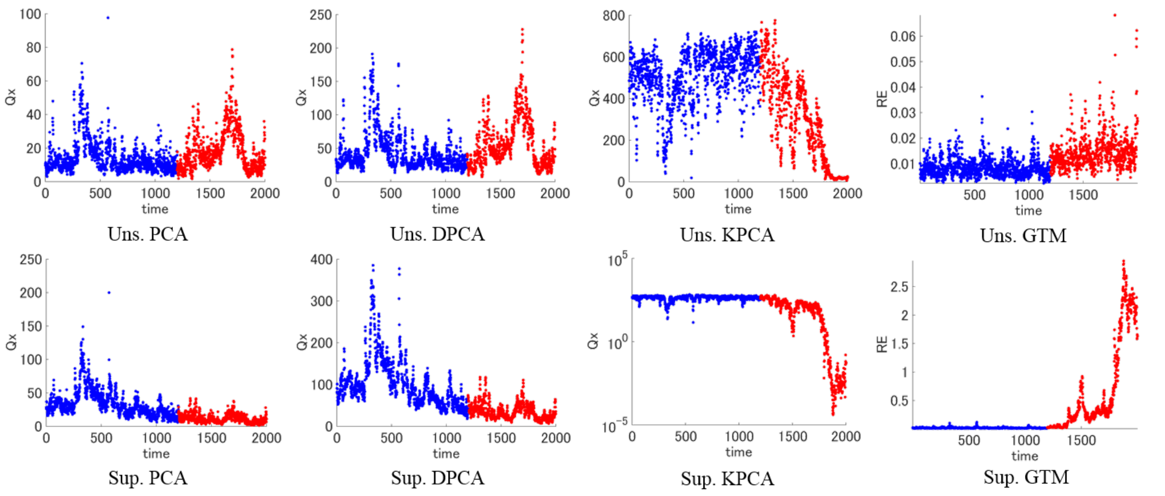

Fig. 2.66 Case A PCA, DPCA, KPCA Q and GTM RE monitoring results for both unsupervised and supervised results, where blue samples are normal and red, anomalous.

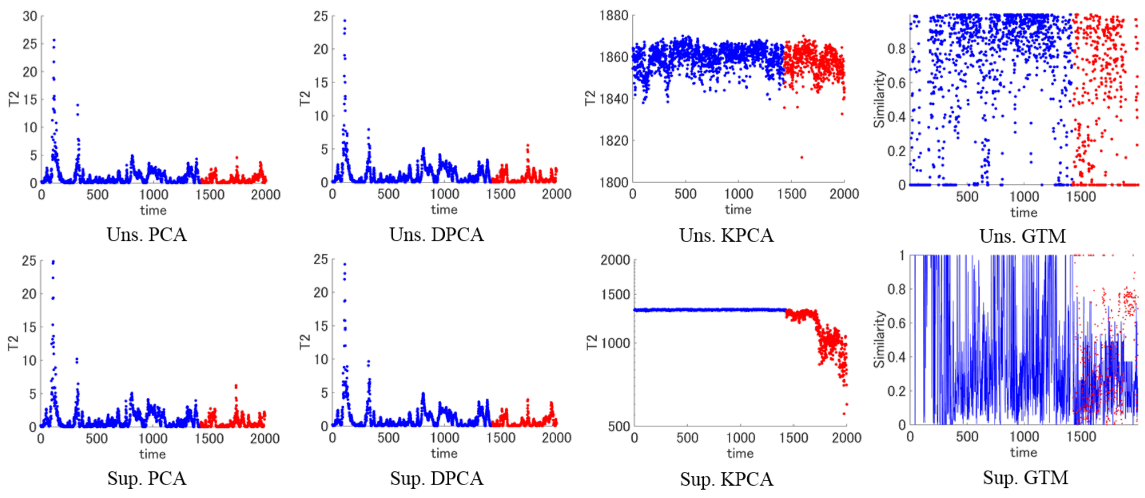

Fig. 2.67 Case B PCA, DPCA, KPCA T^2 and GTM similarity monitoring results for both unsupervised and supervised results, where blue samples are normal and red, anomalous.

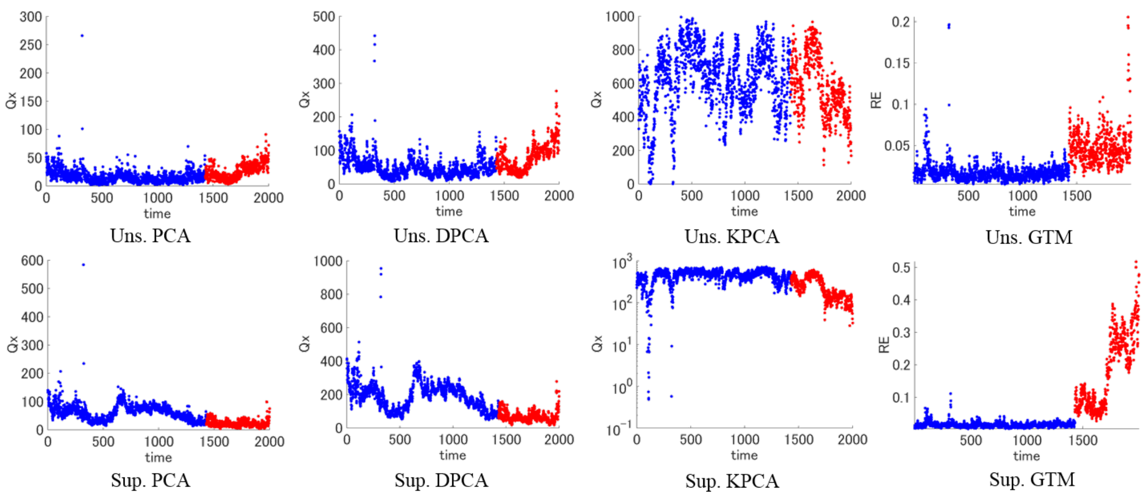

Fig. 2.68 Case B PCA, DPCA, KPCA Q and GTM RE monitoring results for both unsupervised and supervised results, where blue samples are normal and red, anomalous.

From Q, T^2, RE and similarity perspective alone, however, it is difficult to say which is the reason for samples to belong to one or another cluster. Figure 2.69 shows the detection results for GTM+GT on both Case A and Case B. Complementary, the network presented in Figure 2.70 takes GTM information and highlights it, showing clearly how some outliers are actually rather similar to the NC, to the point of correlating with lots of samples. The proposed approach not only aims to discriminate data, but also gives unique insights on the nature of distinct anomalies. What was complicated to assess using the aforementioned indexes becomes much easier to grasp.

As for the reliability of the labeling presented, this example motivates the use of GTM+GT as an unsupervised methodology, since normal data was probably mislabeled as anomaly by the operators. From PCA-based methodologies and GTM alone, one could even argue that part of the anomalies mislabeled as normal are indeed normal, according to a consensus. GTM+GT, however, can identify to what extent are those samples similar to the NC, giving a more reliable tool for judgment. Being able to locate the borderline between normal and anomalous states, especially the instances close to the true beginning of the anomaly, is fundamental to better understand the origin of such deviation.

Case B, on the other hand, due to its higher complexity did not perform as well as expected. GTM's similarity assessment and LCF's clustering narrowed the final NC encountered. In addition, since the anomaly behaves similarly to the normal state in many instants, some of its samples were mislabeled as normal. Even with those drawbacks, however, the network shows that part of the normal data undetected is very similar to the NC, only suffering from LCF's restriction. From a practical point of view, that data could be incorporated to the NC, once some expert knowledge can confirm its nature. Once again, the assessment proposed here does not ignore expert knowledge, but rather uses it as support for validating and confirming true states in the chemical plant.

Despite the results presented, these scenarios show one key limitation to be explored in the future. The notion of unstable normal states is, perhaps, the biggest challenge faced by GTM+GT. Expert knowledge and statistical analysis need to be merged in order to overcome this issue. A semi-supervised strategy would be desirable, combining bits of truly reliable information with unsupervised data clustering. Operators could focus on assessing important moments of the process and use this information to evaluate the entire process span.

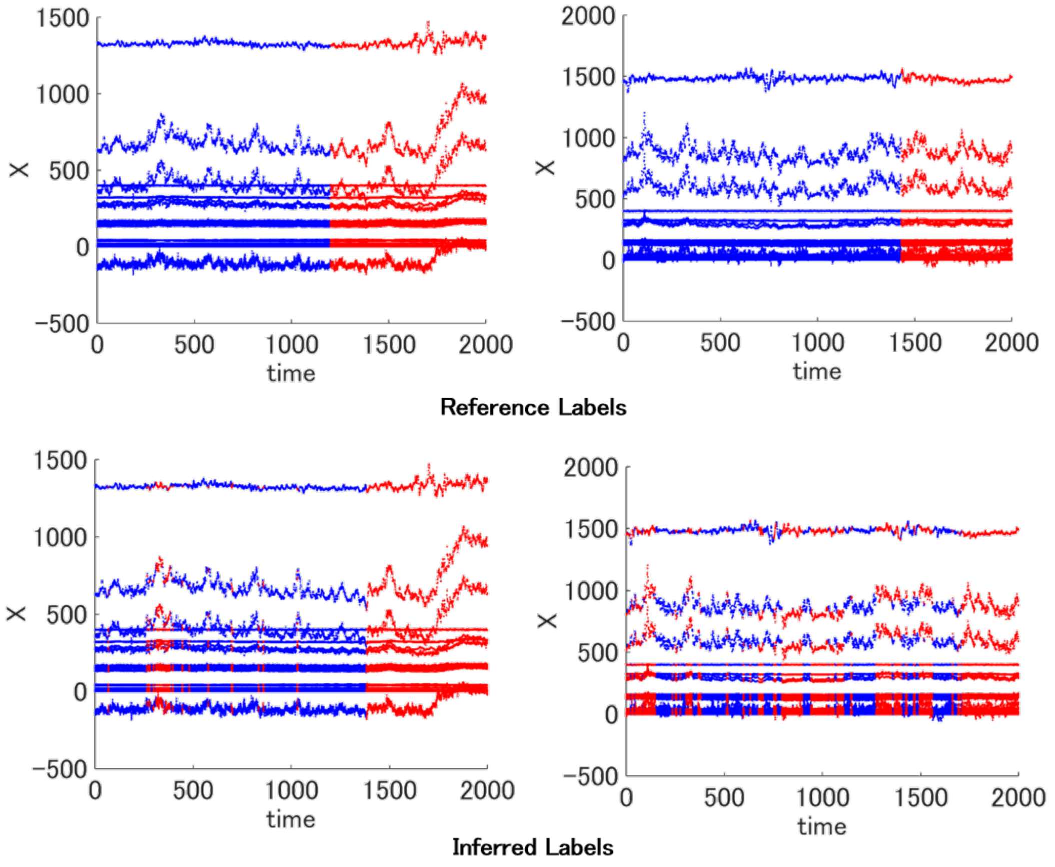

Fig. 2.69 Detection results for cases A and B. Normal samples are blue and anomalous samples are red

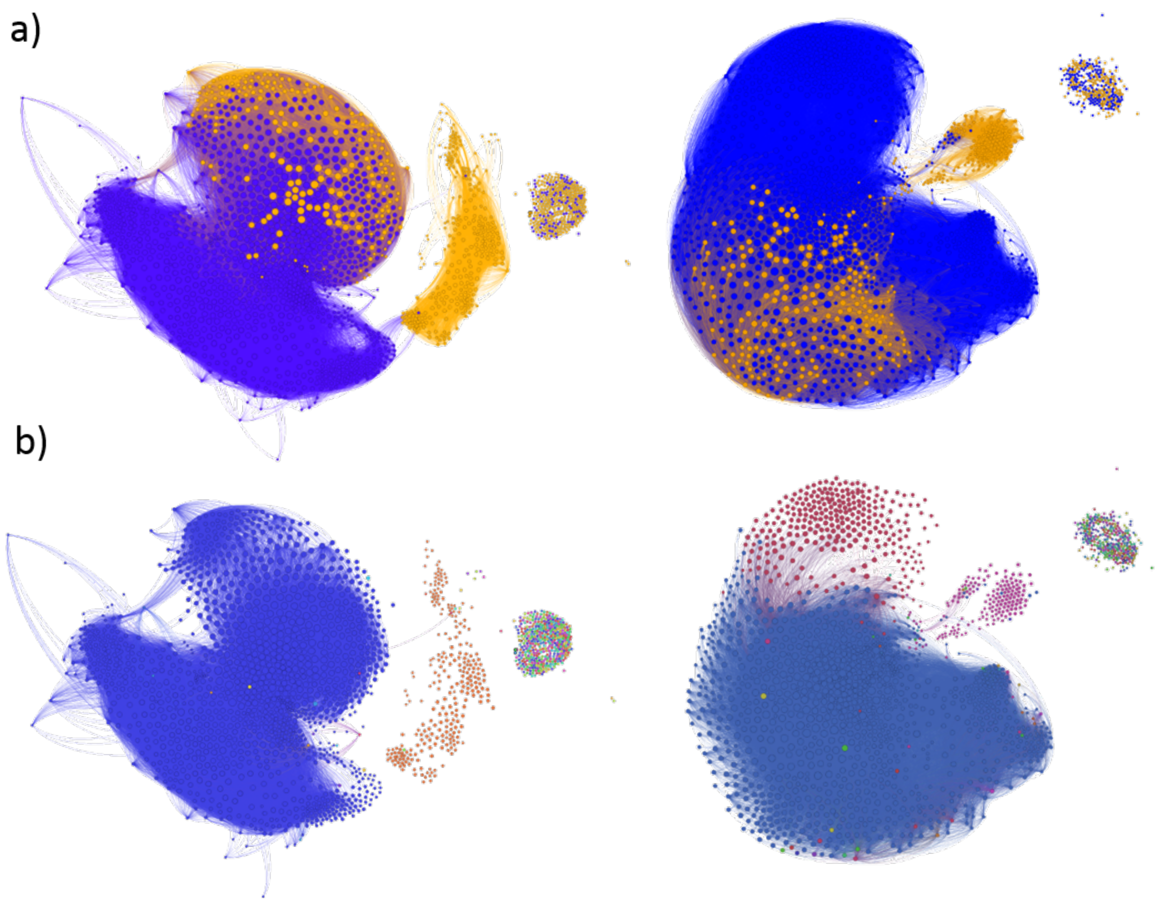

Fig. 2.70 GTM+GT results for cases A and B, showing reference labels and clustering results.

Chapter 3

Applicability Domain & Fault Detection

3.1 Applicability Domain

AD represents to what extent a model can be used to predict other data, being developed originally to deal with quantitative structure activity relationships (QSAR)[102, 103] and quantitative structure property relationships (QSPR)[104, 105]. Analyzing AD aspects can lead to multivariate statistical monitoring, where model accuracy can be monitored through its AD over time3,17,18[3, 31, 32].

Here, the concept of prediction trustworthiness is particularly important. How much one trusts in the prediction is a key element of AD. There are rather distinct approaches which tackle this concept, however the essence of the analysis is to pursue this notion of reliable predictions[106].

Convex-hull methods, for example, try to encapsulate reliable data within a region in space, where data beyond that hull exceeds its AD, considered faulty[107, 108]. These methods establish this hull with concrete and well-defined boundaries. Such boundaries might lead to complications, however, since it assumes complete reliability on the training data used in the first place. Furthermore, how the boundaries are established is a further complication if complex, nonlinear boundaries are required, since most classic convex-hull methods have a geometrical approach[109].

Density estimation based approaches create fuzzy probability clouds to define the variable space[110, 111]. Aside from mere outlier detection, density estimation can be used for support of other applications, such as soft sensors[112]. Defining boundaries, however, may require certain tuning of several empirical parameters, which constrains how easily can one apply it for meaningful AD analysis.

In the realm of distance-based methodologies usually lead to less robust, yet more simple, approaches. T^2 and Q statistics are one of the most widespread techniques for assessing applicability domain[33]. Both indexes can be applied for PCA and PLS assessing related, yet different aspects of data. For PCA, T^2 evaluates how far samples are from the data distribution based on data variance. Q evaluates pure distance from the PCs hyperplane, using residuals for calculating projection error. For PLS, the analysis is similar, but it relies on the reduced PLS matrix. The interpretation is different, where now distance from the model is assessed. T^2 indicates input variation discrepancies in data and Q distance from the model from prediction residuals.

Another alternative relies on exploring the standard deviation of prediction errors, by evaluating the same sample with different models coming from the same methodological source. This can be resulting from ensemble learning[113] or even repeated runs of a particular non-deterministic methodology[30]. For the study in this work, genetic algorithms are used for this assessment, even though any non-deterministic strategy can be considered.

The use of standard deviation allows the evaluation of prediction anomalies. Low average prediction errors should lead to small standard deviations and high prediction errors, to big standard deviations. When high prediction errors have low standard deviation, for example, values are being consistently poorly predicted, which is an indication of measurement error in Y-variables.

3.2 Proposed Strategy

AD is intrinsically related to soft sensor modeling, which can deviate from the general perspective taken so far. Instead of considering the whole process, one can focus on key variables in the system and monitor them specifically.

One way to evaluate AD relies on average prediction values. For the work presented in this thesis, GAPLS and GAWLS are used for the generation of several predictions for each sample. Standard deviation of output variable prediction errors against the average prediction error plots can, thus, be easily obtained for all samples, as expressed in Equations 3.1 to 3.3.

$$\bar{y}_{pred_i} = \sum_{k=1}^{L} \frac{y_{pred_{ik}}}{P} \tag{3.1}$$

$$\sigma_i = \sqrt{\left[\frac{1}{P-1}\right] \sum_{k=1}^{P} \left(y_{pred_{ik}} - \bar{y}_{pred_i}\right)^2} \tag{3.2}$$

$$\Delta y_i = |\bar{y}_{pred_i} - y_{test_i}| \tag{3.3}$$

where P is the number of available predictions, whether it comes from ensemble or different runs. y_{pred_i} is the i^{th} predicted output value, y_{test_i} is the i^{th} output reference value and σ_i is the standard deviation of the i^{th} sample prediction errors. σ_i shows how consistent the prediction is for recurrent models being tested against the test data set. Ideally, small prediction errors lead to small prediction variance, where big prediction errors result in bigger prediction variance. If the methodology can predict a certain output sample with small error, it is expected that such prediction is consistent. Alternatively, poorly predicted samples will have rather different predicted values for different models.

By analyzing this and assuming that the model is reliable, irregularities in predictions can be assessed, leading to detection of samples with anomalies in the measured Y-values (output).

This analysis, though, is not the only one available for AD analysis. Other indexes can be used where the relationship between inputs and outputs can be assessed. Analogous to PCA, T^2 and Q indexes can be applied for model prediction, particularly PLS[33]. The equations are essentially the same ones presented in Equations 4 and 5, but applied to the reduced PLS matrix.

Similarly to the supervised approach for PCA, T^2 and Q are calculated for training data first, but now focusing on prediction values rather than data distribution itself, so to determine a threshold. New test data then, is tested against the model obtained, trying to relate differences in prediction error with exceeding T^2 and Q values. This approach is applied in this thesis for AD assessment and, as a result, anomaly detection, as well.

3.2.1 Genetic Algorithm Partial Least Squares

Genetic algorithms (GA) are presented in this thesis as a methodology for soft sensor modeling, which through ensemble prediction leads to faulty data assessment. Specially, but not limited to, systems with many features, some of those might be redundant or unnecessary for PLS modeling. From this premise, the best variable set should be optimally selected. GA can be used for this goal, being based on a direct analogy to evolutionary biology. A random population with different characteristics will be naturally selected according to their adaptive potential, being subject to an environment where certain key features lead to reproduction and survival. Those more fit will reproduce and propagate their features along several generations[114]. GA-based PLS is an extrapolation of this idea for modeling[34], whose conceptual representation can be seen in Figure 3.1.

Initially, a population is created with a predetermined number of random chromosomes whose nonlinear optimization solutions are feasible. To express the suppression (non-selection) or expression (selection) of a gene (variable) in the chromosome, a continuous value from 0 to 1 is used. Assuming a threshold, which in this study is 0.5, values below it are not selected. If any value is higher, however, the gene is selected. For GAPLS, the genes are the input features of the system, where PLS models are generated for each chromosome for the prediction of an output vector $\mathbf{y}$. In order to find the fittest chromosomes, the predictability of a model given a training data set is evaluated, according to the index q^2, shown in Equation 3.4.

$$q^2 = \sum_{i=1}^{N} \frac{(y_i - y_{CV_i})^2}{(y_i - \bar{\mathbf{y}})^2} \tag{3.4}$$

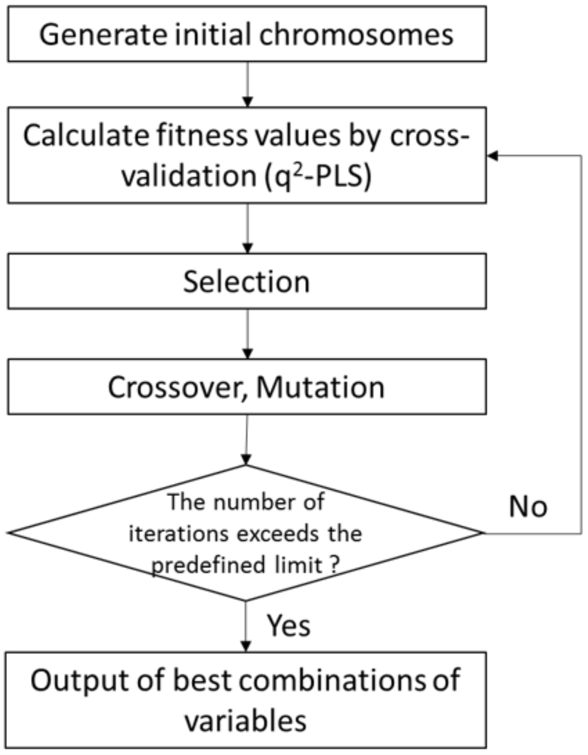
Generate initial chromosomes
Calculate fitness values by cross-validation (q²-PLS)
Selection
Crossover, Mutation
The number of iterations exceeds the predefined limit ?
No
Yes
Output of best combinations of variables

Fig. 3.1 Schematic representation of GAPLS procedure.

where y_i is the i^{th} output value, y_{CV_i} is the i^{th} value predicted by cross-validation and $\bar{\mathbf{y}}$ is the average output value. Once all fitness indexes are calculated, the best ones are selected for cross-over and mutation. Cross-over is the key element for evolution, where chromosomes interact with each other to create new chromosomes that might be more fit than the previous ones. Mutation stands for the randomness of a change in one or more genes in a chromosome, also aiming for better variable sets. Both cross-over and mutation rates are defined in advance, therefore only a portion of the genes and chromosomes is changed. These steps are repeated for a preset number of generations. In the end, the best variable combination (chromosome) in the population is selected and presented as a final solution. Due to its nondeterministic nature, it is important to notice that for each new GAPLS run, the final solution might be different, which is interesting for ensemble prediction purposes, leading to anomaly detection capabilities.

3.2.2 Genetic Algorithm-based Wavelength Selection

The concept behind GAWLS is similar to GAPLS, since PLS is still being used for modeling. What changes, however, is how variables are selected[35], since GAWLS selects wavelength regions instead of individual features. The difference in strategy can be seen in Figure 3.2. The structure of the chromosome is now different. Two parameters are used for generation: number of windows, which determines the number of regions to be optimized, and frame size, indicating the maximum interval of each region. Each two genes represent one region, where the first value indicates initial wavelength and the second value shows length. The overall procedure is exactly the same as GAPLS, but the chromosome's structure is different, guaranteeing regions to be found and not scattered data.

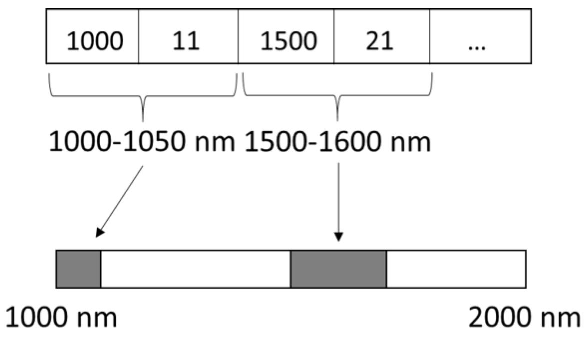

Fig. 3.2 GAWLS chromosome coding rule for a data set with 5nm sampling.

3.3 Case Study - Flour & Protein Content Data Set

AD is discussed via a GAPLS and GAWLS approach, trying to relate soft sensor performance, data splitting and Y-anomaly detection, based on an article previously published[30]. The case study used a flour and protein content data set for evaluation, to be presented in next section.

3.3.1 Data Characterization

The data analyzed consists of 34 samples data set from different brands and types of flour, where the protein content was measured in triplicate using two distinct techniques: Fluorescence and Near-Infrared (NIR) spectroscopy. There are, then, 102 values in total for each method.

All samples were characterized off-line by a Farinograph analysis, using a Brabender GmbH & Co. KG, Duisburg, Germany, model FD0234H, and the protein content was determined by a Hach, Dusseldorf, Germany, Digesdahl Digestion Apparatus. Parallel to that, online characterization was performed by Fluorescence and NIR spectroscopy. NIR data was collected via a Bruker Optik GmbH Ettlingen, Germany, Multi Purpose NIR Analyser and Fluorescence data was collected via a BioViewR sensor (DELTA Light & Optics, Denmark).

NIR spectroscopy has 1150 variables, referring to near infrared reflectance for each wavelength, uniformly distributed between 800nm and 2758 nm. Fluorescence has 150 variables, represented by the light intensity for several pairs excitation (270 - 550 nm) and emission (310 - 590 nm) wavelengths. A summary of all this information can be seen in Table 3.1.

Table 3.1 Fluorescence and NIR spectroscopical summary.

Date	Samples	Inputs	Inferences(Outputs)
Fluorescence	34x3	150	1
NIR	34x3	1150	1

Two interesting aspects of this case study are worth noting. Firstly, this application differs from usual online process monitoring. Samples are not time related, even though online NIR and Fluorescence measurements are possible. Secondly, the number of samples is far smaller than thousands of data points available in most processes. The techniques applied for online measurement are explained in the Appendix II and III.

3.3.2 On Applicability Domain and Fault Detection

The analysis portrayed here has an intimate connection with a basic pre-processing philosophy that is ignored by many researchers: knowing how to define your training and test data for soft sensor modeling. When it comes to developing a soft sensor of any sort, the way data is selected has a great impact in the final accuracy and predictive capability of the model developed. More specifically, the way a data set is split between training and test groups reflects directly on how consistent and reliable the analysis will be. Ignoring it may lead to model misunderstanding, which affects any posterior application using that model, such as soft sensor prediction and anomaly detection.

Regardless of the application, however, the methodology for assessing how adequate the splitting is does not change. AD is in this section the bridge connecting data splitting with anomaly detection, using soft sensor modeling results for evaluation. Soft sensor models can be interpreted from both data splitting and fault detection perspectives, each with its own significance and complementary information, where GAPLS and GAWLS, given their non-deterministic nature, help in the AD analysis, providing different prediction models.

Data Splitting

Different data split set-ups were devised, following a training:test ratio 0.7:0.3, commonly used. As described in section 3.3.1, the data is composed of 34 flour samples, where for each sample triplicate experiments were performed, giving a total of 102 values.

Initially two big groups were considered, according to the nature of their splitting: Sample splitting (SS), where data is split according to the samples, *i.e.*, experiments of the same sample are kept together, and Experiment splitting (ES), where the samples have no restriction to be split. Three different strategies for splitting were assumed for each group, resulting in the six scenarios in total, as shown in Figure 3.3.

Initially, two strategies were applied for both SS and ES. The first method, the most widely used one, is called Random Selection (RS), where data is randomly split in two groups, according to a ratio defined in advance. For practical purposes, one random splitting was chosen and kept this way, allowing the reproducibility of results. For SS, the method was called Sample Random for differentiation (SR).

The second approach is called Y-ranking (YR), where data is initially sorted from lowest to highest Y-values. Test data is space evenly along the Y interval, following a given ratio. For SS, the same approach is Sample Y-ranking (SYR).

Knowing that each sample has three experiments, another splitting was also selected in the ES group, by choosing one experiment (OE) as test data and two experiments as training data. Finally, one last splitting in SS was determined, Y-splitting (YS), by selecting a threshold in Y ($Y = 0.85$) and splitting the data above and below in training and test data, respectively. Table 3.2 presents a summarized description.

Table 3.2 Data splitting description summary

Experiment Splitting (ES)	Random Selection (RS)	Data is randomly split in training / test data
	Y-Ranking (YR)	Data is sorted according to Y, so that test data can be selected evenly along Y range
	One Experiment (OE)	One experiment from each sample is selected as test data
Sample Splitting (SS)	Y-Splitting (YS)	Data is split based on a Y threshold (Y = 0.85)
	Sample Random (SR)	Data is grouped in samples and randomly split in training / test data
	Sample Y-Ranking(SYR)	Data is grouped in samples and sorted according to Y, so that test data can be selected evenly along Y range

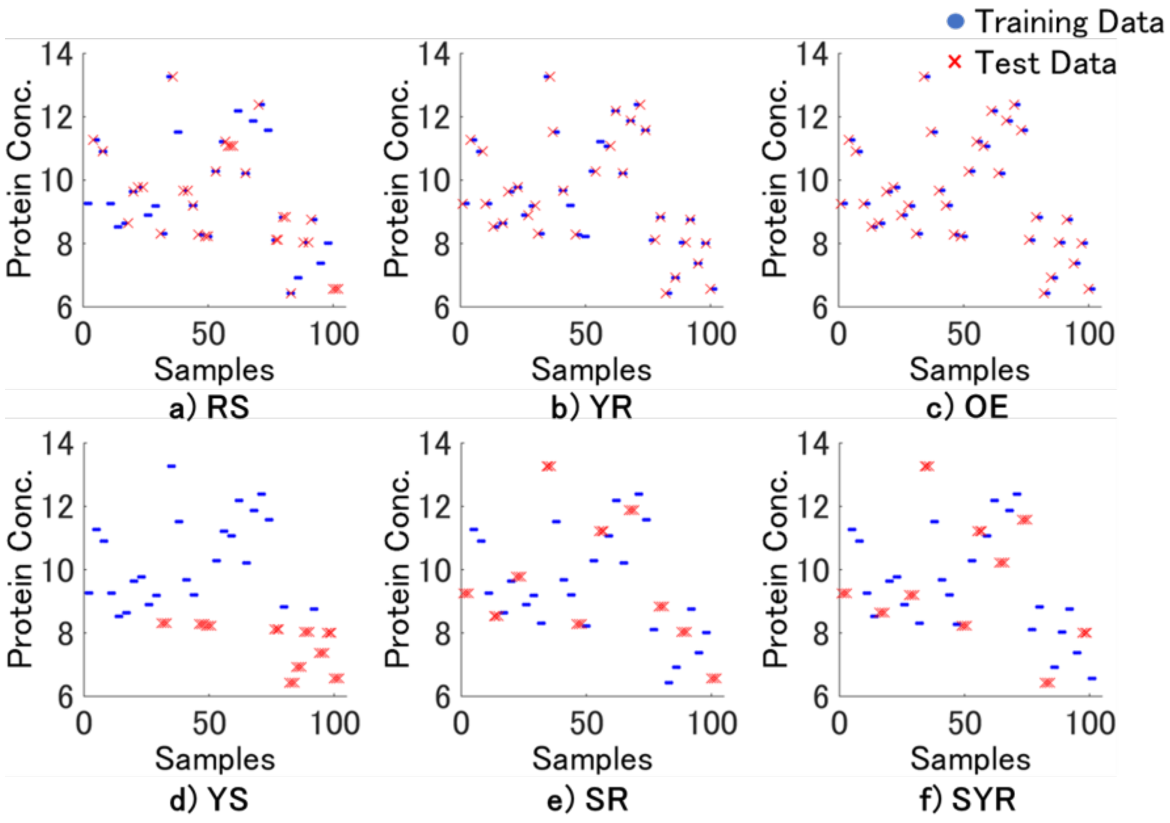

Fig. 3.3 Data splitting for flour samples.

Soft Sensor Performance

Initially, GAPLS and GAWLS models were generated for the prediction of protein content, following the simulation parameters described in Table 3.3. For both GAPLS and GAWLS, the number of variables selected is usually smaller than the original data set. In order to ensure that GAWLS, the parameters for frame size and number of intervals were limited to a combined maximum of 800 variables. Since the original variables set for NIR consists of 1150 variables, fewer variables are guaranteed to be found.

It is important to notice that GAPLS can be applied freely for both NIR and Fluorescence data, whereas GAWLS, due to its one-dimensional interval selection, can only be applied to NIR. In addition, GA methods are non-deterministic optimization algorithms, leading, most likely, to different results every time. Repeating simulations, therefore, is recommended. From a soft sensor perspective, it gives more reliability to the modeling methodology. For each fluorescence and NIR predictions, therefore, 30 runs were simulated for all six scenarios.

Table 3.4 and 3.5 present the main quantitative results for GAPLS and GAWLS. Coefficient of determination r^2, predictability index q^2 as described in Equation 3.4 and Root

Table 3.3 GAWLS and GAPLS parameters.

	GAPLS	GAWLS
Population size	300	150
Generation size	300	150
Frame size	–	160
Number of windows	–	5
Crossover probability	60%	60%
Mutation probability	5%	5%
Fitness PLS components	10	10

Mean Square Error (RMSE) are used to evaluate model performance. r^2 and r^2_{pred} expresses how accurate the model is for training and test data, respectively, where the closer the value is to one, the more accurate the prediction is. q^2 based 10 fold cross-validation expresses the predictive power for the model generated, being used to determine the optimal number of principal components (#PC) and the optimal run among the 30 simulations for each case. The closer q^2 is to 1, the more predictive the model is. RMSEpred represents the model error for test data. SVratio represents the proportion of variables selected during GAPLS and GAWLS.

There is a clear distinction between the performance of Fluorescence and NIR models, based on r^2_{pred} and $RMSE_{pred}$ values. This is most likely due to the nature of NIR data itself. NIR has more variables available for selection, which potentially allows for a greater combination of wavelengths for better models. Furthermore, NIR information just might be more meaningful for representing the flour and protein content data.

As for the differences in split, ES models have overall smaller prediction errors when compared to SS, where YR and OE present similar performance. YS and SR scenarios in particular had rather low accuracy. Being that said, however, looking at prediction errors for NIR it is evident that YS actually has lower prediction error than SR for both GAPLS and GAWLS, despite rather low r^2_{pred}. YS even outperforms SYR for the GAPLS models. Looking at r^2, q^2 and $RMSE$ alone is not enough to capture the model behavior, not explaining, therefore, the differences in performance and model quality. AD analysis acts as a complementary analysis to those indexes, using the prediction results to assess model performance and presence of anomalies.

Applicability Domain Analysis

One big aspect of modeling results is how to handle $RMSE_{pred}$ values. More than evaluating the best case scenarios, checking prediction errors on a sample basis can give more information on the predictive capabilities of the model created. Figure 3.4 shows GAPLS and

Table 3.4 GAPLS results for Fluorescence and NIR.

			Fluorescence			
	r^2	q^2	r_p^2red	$RMSE_{pred}$	$\#PC$	SV_{ratio}
RS	0.9458	0.8139	0.6613	0.9525	15	0.3891
YR	0.9507	0.7972	0.8246	0.7223	20	0.3822
OE	0.9336	0.7799	0.7644	0.8183	17	0.3553
YS	0.9552	0.7900	-14.26	2.7053	21	0.3769
SR	0.9454	0.8261	0.1941	1.7104	29	0.3829
SYR	0.9709	0.8809	0.0480	1.8847	23	0.4187
			NIR			
	r^2	q^2	r_p^2red	$RMSE_{pred}$	$\#PC$	SV_{ratio}
RS	0.9741	0.9086	0.8897	0.5547	12	0.4897
YR	0.9694	0.8905	0.9213	0.4879	12	0.4259
OE	0.9812	0.9086	0.9001	0.5382	11	0.4297
YS	0.9433	0.8790	-0.460	0.8448	10	0.4925
SR	0.9872	0.9265	0.6539	1.1201	17	0.4656
SYR	0.9907	0.9488	0.6848	1.0648	19	0.4556

Table 3.5 GAWLS results for NIR.

			NIR			
	r^2	q^2	r_p^2red	$RMSE_{pred}$	$\#PC$	SV_{ratio}
RS	0.9670	0.9394	0.8600	0.6215	10	0.2436
YR	0.9553	0.9319	0.9350	0.4440	10	0.2751
OE	0.9648	0.9366	0.9178	0.4859	10	0.2659
YS	0.9538	0.9305	-1.435	1.0737	10	0.2667
SR	0.9722	0.9430	0.6576	1.0972	10	0.2491
SYR	0.9827	0.9713	0.7005	1.0325	10	0.2340

GAWLS prediction results for NIR data, using a Y-Y plot for all data splitting scenarios. Y-Y plots show the discrepancy between the real and predicted values for all samples.

Overall, it is hard to assess whether GAPLS or GAWLS can generate better models, since both methodologies outperform the other depending on the scenario. GAPLS selects wavelengths in a more scattered distribution, whereas GAWLS looks for specific wavelength intervals with more relevant information about the system. Due to this particular distinction in the selection algorithm, GAWLS $SVratio$ tends to be smaller than GAPLS's. The important point to be discussed here, however, is related to the samples being predicted. Both methodologies suffer from this issue, but particularly for GAWLS there are a few samples that are being poorly predicted. Are those results an indication of model error, sampling error

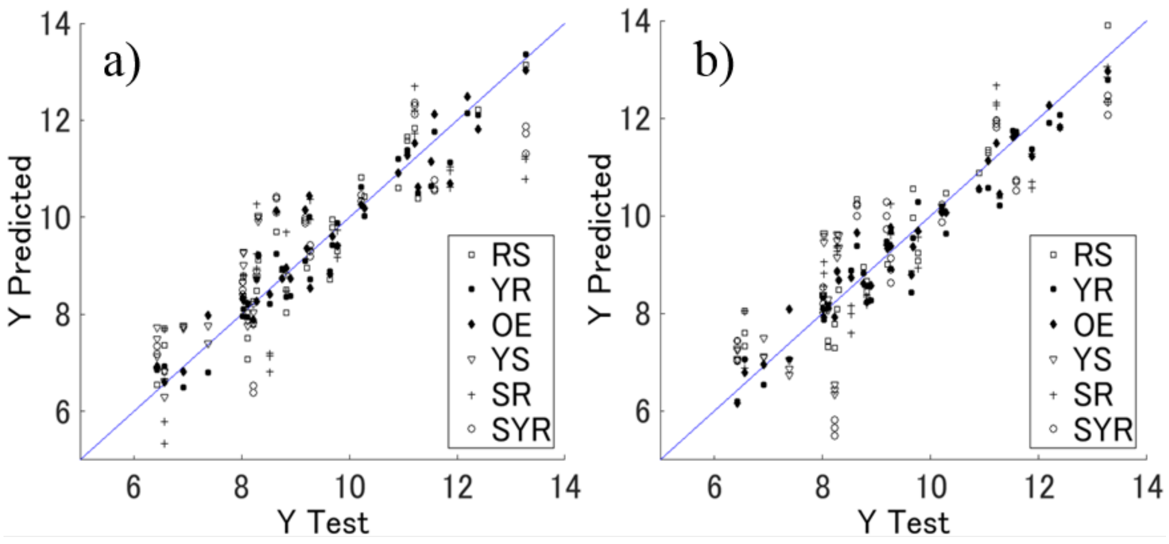

Fig. 3.4 Integrated Y-Y plot for all splitting in a) GAPLS and b) GAWLS NIR data.

or poor data splitting? Without going further in the analysis, little can be said just looking at this plot, even though it gives insight on how samples are being predicted in the big picture.

One of the techniques applied for assessing AD is described in section 3.1, where the standard deviation of prediction errors for a certain methodology can be plotted against the average prediction error for each sample. Figure 3.5 shows these plots for GAPLS Fluorescence and NIR data and GAWLS NIR data.

From a fault detection perspective, the plots shown in Figure 3.5 give valuable insight regarding which flour samples are more likely to be anomalous. Each data splitting represents a feasible scenario where data was chosen for the construction of a soft sensor model. By plotting the standard deviation of prediction errors against the average of those predictions for all samples in the test data set, one can infer which samples have an odd behavior. It is expected that samples with low standard deviation will have a lower average prediction error than those whose standard deviation are higher. When a sample presents low standard deviation and high average prediction errors, this is an indication of anomaly, particularly measurement error in the Y-variable.

For the case study discussed here and taking as reference initially GAWLS results, it is easy to visualize in Figure 3.5 that for all scenarios in ES, one experiment stands out from the other ones, and that for SS, SYR has at least one triplicate with very discrepant values around $Y = 1.5$. For all cases, experiments from the same flour sample were detected to be anomalous. In the ES scenarios, only one experiment was detected because the other two anomalous experiments are included in the training data for the model generation. For SYR, the entire sample stood out as an anomalous scenario, since data was split based on

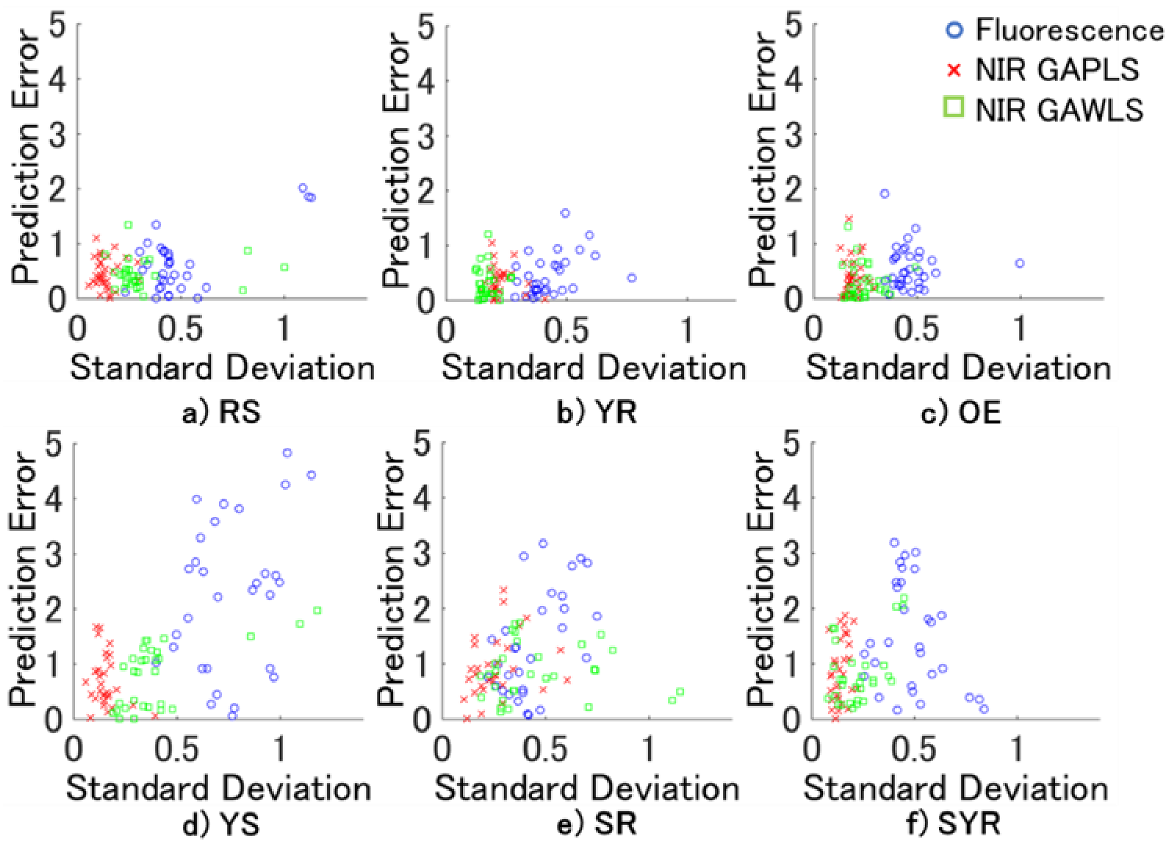

Fig. 3.5 Standard deviation and prediction errors plot for Fluorescence GAPLS models and NIR GAPLS and GAWLS models.

the samples, and not the experiments. The evaluation of the AD of those samples allowed the detection of anomalous experiments or triplicates, even when the training database was contaminated with anomalies.

Another interesting aspect can be noticed for the remaining SS scenarios. Taking a closer look on the SR plot, no sample appears to be anomalous, which is odd considering that other four different data splitting scenarios detected one sample. By checking its training data set, however, the presence of the anomalous triplicate was confirmed. In the test data set presented, therefore, there are no anomalies to be detected, which are reflected on a seemingly normal plot. A similar behavior can be seen in the YS plot, which suffers from the same issue as SR. What is, however, the true impact of this assessment?

Here, two different aspects of data are affecting the soft sensor performance: data splitting and outlier contamination. Depending on the scenario, one effect outweighs the other and vice-versa. For all ES scenarios, for example, the training data set is contaminated with two anomalous experiments. Due to the proximity of both training and test data set, however, the prediction is not really affected by it. Looking exclusively at prediction accuracy, thus, would be deceptive, since the anomaly would never be detected. Moreover, if both training and test data have essentially the same information, regardless of being anomalous or not, models are generated and predict with great accuracy that particular test data. This shows how misleading the construction of a model can be if one is not really concerned about which information is being added on both training and test data sets. One could falsely conclude, by looking only at the ES scenarios, that the soft sensor methodology performed rather well, which is not a reliable conclusion whatsoever.

For the SS scenarios, separating entire flour samples decreased the overall accuracy of the model, but allowed a better visualization of the true nature of the system. Furthermore, for SYR it showed that one entire triplicate was faulty. This affects SYR prediction error, since the model is trying to predict samples that do not belong to the AD of the model created. For YS and SR there is a pronounced combined effect of poor data splitting and training data contamination that affects the performance of the models generated.

GAPLS results are similar to GAWLS ones, if only NIR results are considered. Fluorescence data can hardly detect the anomalies on ES, since its samples are far too spread to identify clearly which experiments are anomalous or not. As for the NIR results, GAPLS results can be said to be somewhat different from the GAWLS ones. All detected experiments match the ones found by GAWLS in the ES section, but in the SS section, the judgment is more difficult, since there is no clear detachment of one triplicate compared to the rest of the data.

Besides the standard deviation of prediction errors, though, the potential of T^2 and Q indexes for AD analysis can also be discussed. This approach takes into account the distance from the PLS model generated, where T^2 is related to changes in input variation for that model and Q assess the distance from the model in the form of prediction residuals. The main assumption here is that for low prediction errors, test data T^2 and Q would fall within the training sample range for the same indexes. Based on this principle, higher prediction errors would have T^2 and Q values beyond the training data thresholds. Any samples whose behavior does not correspond to what was previously mentioned are potentially anomalous. The threshold is defined as 99% of the maximum value within the training data range for both indexes. Figure 3.6 to 3.9 shows T^2 and Q plots for GAPLS and GAWLS NIR data.

As expected, there was a bigger overlap for both indexes in the ES scenarios than the SS ones, since training and test data are more similar in the former. With that being said, though, there is no consistency on which samples are exceeding the threshold and there is no clear correlation between prediction error and any index, regardless of the methodology used for model generation.

In the standard deviation analysis, both ES and SS splits had a general convergence on which samples were most likely to be anomalous. For T^2 and Q analysis, however, different scenarios led to different faulty samples. Since data splitting has to be carefully considered, so to avoid false conclusions, having a reliable index behind it whose applicability depends little on the correct data splitting is desirable.

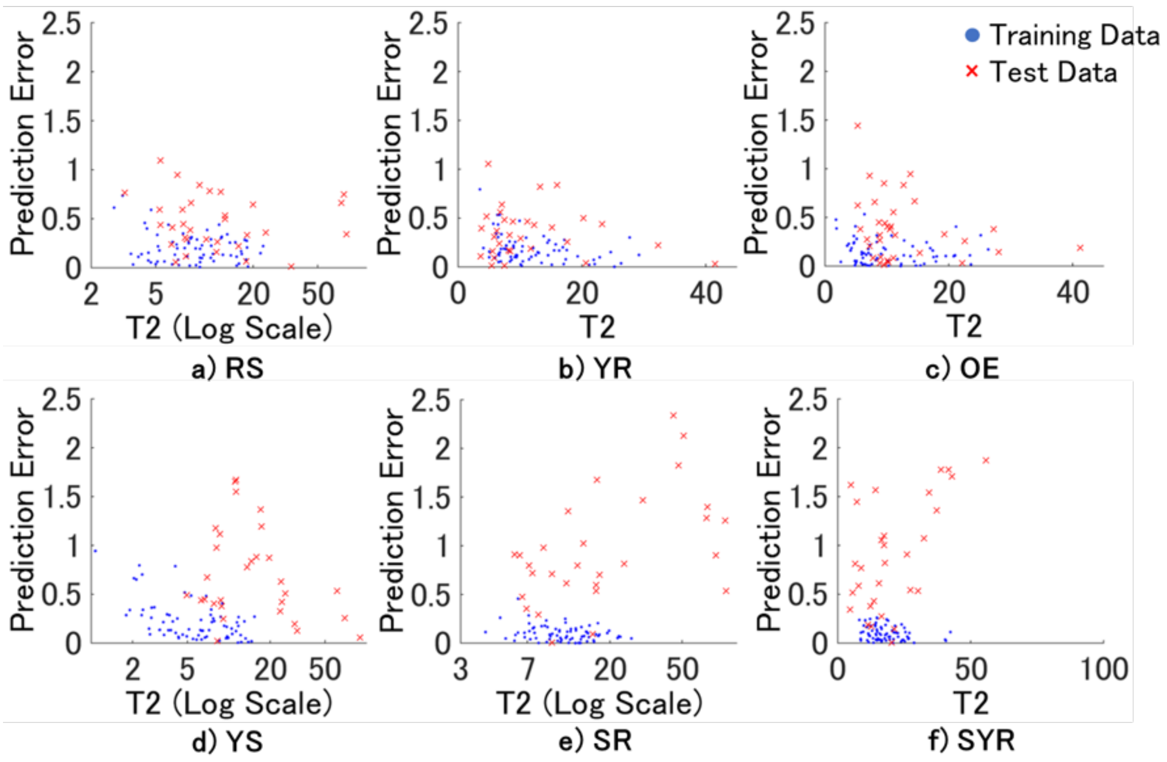

Fig. 3.6 T^2 and prediction errors plot for NIR GAPLS.

The characteristics and interpretation of each index reflect on their performance. T^2 is associated to input variation within the model generated. Considering the samples with highest prediction errors, test T^2 values only exceeded the training data range for a significant amount of samples for the models with the worst accuracy, YS and SR. For the other models generated, test data shows little correlation between prediction errors and input variation.

Q is associated with prediction residuals and, thus, with the distance from the model. For GAPLS, distance from the model and prediction error show some correlation, especially for the SS scenarios, even though there is little discrimination between low and high prediction errors. This reflects on the consistency of this analysis, where most ES samples are exceeding the threshold, even for low prediction errors, and for SS all of them exceed the threshold, also regardless of the prediction error. Q results for GAWLS, on the other hand, could be considered the best among all T^2 and Q results presented. This is not to say that the Q

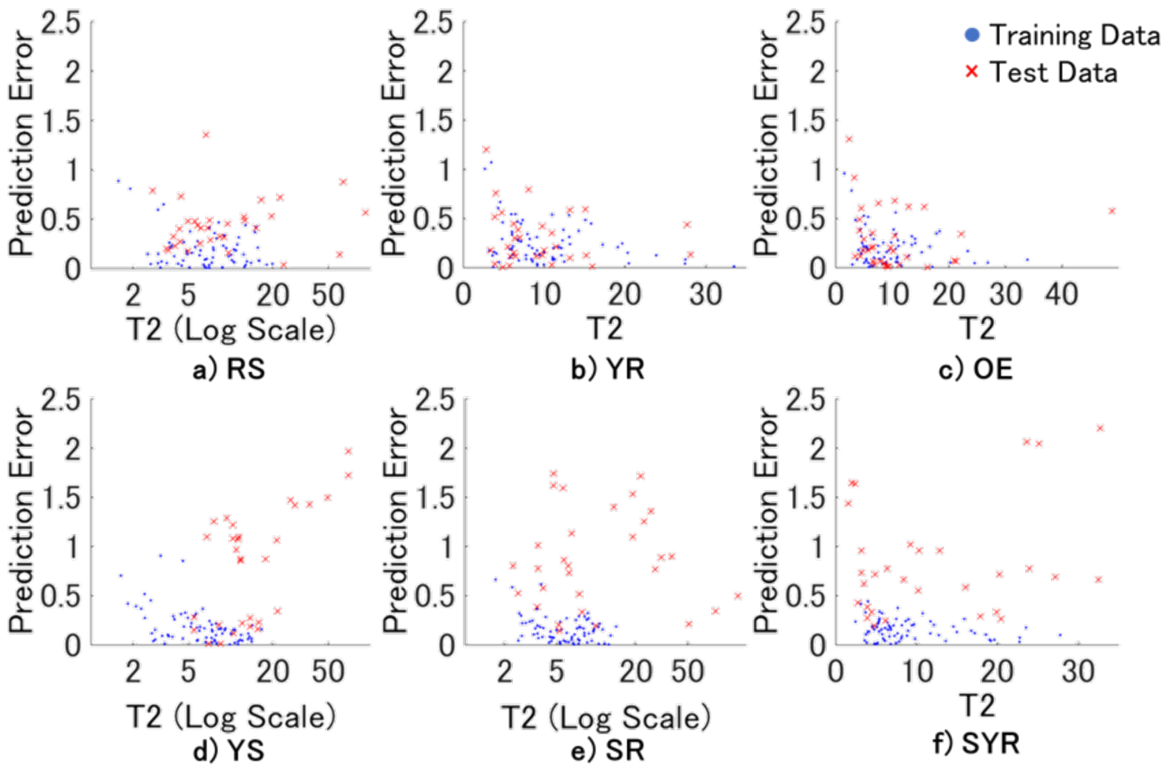

Fig. 3.7 T^2 and prediction errors plot for NIR GAWLS.

is successful on detecting anomalies, but rather that it showed the least inconsistency in detection. On Figure 3.9, the anomaly detected from the standard deviation approach is detected in the SYR plot where a high prediction error shows a value of Q within training data range. The same can be seen for the individual experiments, coming from the same flour sample, detected on each ES scenarios. Along with those detections, however, several other distinct samples are being detected as well. In the end, differently from the standard deviation analysis, T^2 and Q indexes are not consistent enough for them to be reliable as indexes for AD evaluation and, consequently, anomaly detection.

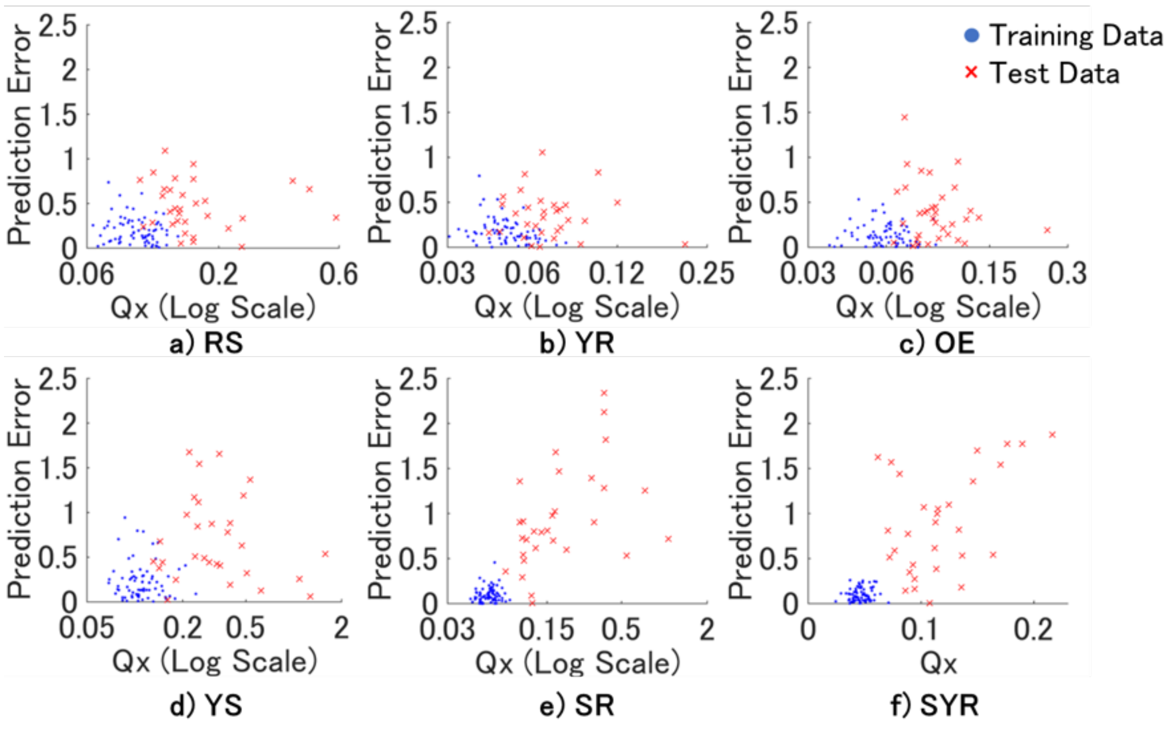

Fig. 3.8 Q and prediction errors plot for NIR GAPLS.

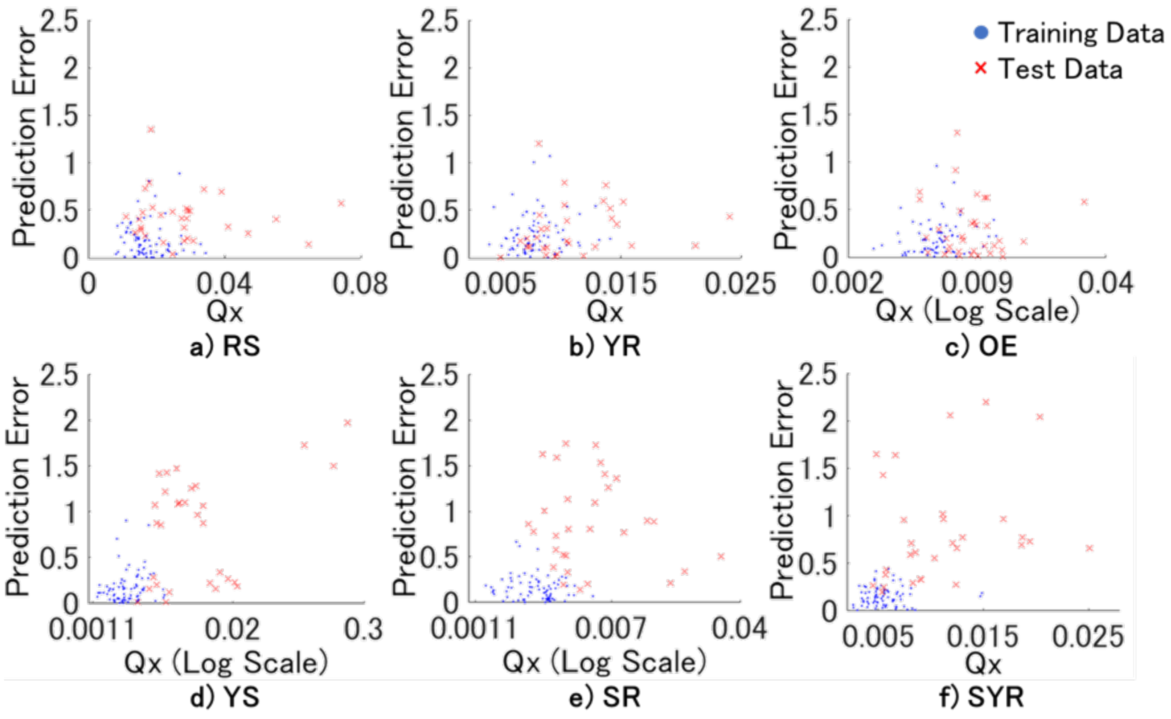

Fig. 3.9 Q and prediction errors plot for NIR GAWLS.

Chapter 4

Conclusion and Future Plans

Regardless of the application, relying on the data being used for assessment is desirable. To what extent can one trust on the already available information is challenged here by showing different aspects of modeling. From applicability domain and data splitting applied to evaluate soft sensor capabilities to unsupervised approaches used for process monitoring, the crucial element here is how to not deceive yourself into ignoring information that could lead to misunderstandings and false conclusions.

Initially, one should be careful on how training and test data sets are created. This is a basic premise, on which the whole work is based upon. When it comes to developing a soft sensor of any sort, the way data is selected has a great impact in the final accuracy and predictive capability of the model developed. Overlooking this matter is recurrent in research; methodology itself becomes so important that data sets used for test come in second place. It should be the other way around. How to pursue better modeling is important, but if the premise of the work relies on biased data sets from the start, very few conclusions can be taken from any model generated.

Biased data sets not only prevent models from high performance, but also constrains their application to other aspects of statistical analysis. Predictive modeling and anomaly detection are two feasible applications that might come from the same modeling source. The flour data set presented in this work dealt not only with regression models, but also on how to use those predictions in favor of anomaly detection. GAPLS and GAWLS were used for modeling, where their non-deterministic structure led to several models generating different predictions for each sample. Those results were essential for the AD analysis explored in this case study. By evaluating the average prediction error for each sample against the variation of such errors, the AD of all scenarios could be assessed and anomalies on Y-values could be detected. AD is the key element here, acting as a bridge between prediction and fault detection and allowing intricate reliability assessment.

The distinction between ES and SS was an example of this, where the former had high accuracy models, but with rather narrow AD, and SS had lower prediction accuracy, but with higher AD. The different perspective on the data led to more substantial info about the nature of samples. The detection of possible anomalous experiments was possible for both ES and SS, however a deepened sense of understanding regarding those anomalies was presented by SS, due to the bigger dissimilarity between data sets.

Complementary to data splitting and AD issues, the use of unsupervised data was another aspect worth of discussion. It shed some light on the nature of process monitoring and its relation with supervised data and supervised techniques in practical applications. This part of the work aimed to challenge the usual notion the supervised approaches are fundamentally better than unsupervised ones, and by doing so propose a change on how processes can be analyzed in the future.

The notion of supervision is deeply connected to certainty, and more than, reliability. Accepting labels without further inquiry is common, due to the inherent sense of ease and trust associated to them. Such mindset, however, might be harmful if by any reason those labels do not match the reality of the chemical plant. The notion of permanency is, at most temporary in a process, so one should be careful with the data being chosen over an extended period. A false sense of security can be easily manufactured by routine itself or by epistemic overconfidence, rooted in biases we all carry.

Unsupervised methodologies can approach the process from a fresh perspective, free of bias. Again, this is not to say supervised information should be ignored, but rather that an unsupervised layer could support existing methodologies, aiding in process monitoring. This is only valid, however, if unsupervised methodologies have the potential to perform close to supervised ones. The simulation data set and TEP case studies aimed to justify the use of an unsupervised approach, by proposing a combined GTM and Graph Theory approach that would perform as well, or even outperform, supervised methodologies.

GTM extracts relevant information for similarity assessment of distinct samples, while minimizing the impact of unnecessary features in such assessment. Graph Theory takes this information and brings it to another level, by encapsulating similar data through the core relationship between samples. It is by joining the merits of both methodologies that data discrimination and visualization could be devised. The results presented here for both case studies reveal an unexplored potential arisen from the exploration of connections between data, which has space for further developments. The use of Graph Theory in chemical engineering, from the samples perspective, is little to no existent.

Considering the developments made so far, however, it is clear that there is a myriad of developments that can be pursued in the future, so to improve and diversify the range of applications of the methodologies described in this work. In the realm of Graph Theory, for example, network modeling can be explored, trying to fit data in the system by assuming distinct graph growth models. GTM extensions can be more thoroughly analyzed, so to truly incorporate system dynamics, allowing a more reliable assessment of the process' dynamics.

From a practical point of view, the evaluation of real chemical processes brings new challenges. When it comes to on-line monitoring, for example, GTM+GT performed well and could handle most faults presented. Analyzing new data and assessing their relationship depends greatly on NCs obtained. On one hand, Restraining NC's similarity domain can lead to future normal samples being mislabeled. This is related to applicability domain issues that should be considered more thoroughly for future endeavors. On the other hand, anomalies whose range and dynamics was similar to the normal instances are being mislabeled as well. One could argue that, because of such similarity, mislabeling has little impact in the existent data pool. Regardless, properly defining what truly characterizes an anomaly should also be subjected to more studies.

Moreover, one of the main assumptions regarding the proposed methodology is how only one normal state exists, facing multiple anomalous scenarios. The presence of multiple normal states, however, is also possible and very interesting from a research point of view. The current methodology developed cannot cope with these systems from a purely unsupervised approach. From this premise, Semi-supervised techniques might be an interesting follow-up for this research. Bits of information can be gradually added to the data set, aiming to identify different normal regions in the operational space. In the end, despite the results presented and the analysis portrayed here, one has to admit that there are so many possible advances available and paths to pursue in the future.

References

[1] L. Chiang, R. Braatz, and E. Russell, *Fault Detection and Diagnosis in Industrial Systems*. Springer London, 2001.

[2] C. Aldrich and L. Auret, *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*. Springer, 2013.

[3] H. Kaneko, M. Arakawa, and K. Funatsu, "Applicability domains and accuracy of prediction of soft sensor models," *AIChE Journal*, vol. 57, no. 6, pp. 1506–1513, 2011.

[4] C. Kuehnert, *Data-driven Methods for Fault Localization in Process Technology*. KIT Scientific Publishing, 2013.

[5] H. Noura, D. Theilliol, J. Ponsart, and A. Chamseddine, *Fault-tolerant Control Systems: Design and Practical Applications*. Springer, 2009.

[6] J. Smith, H. Van Ness, and M. Abbott, *Introduction to Chemical Engineering Thermodynamics*. McGraw-Hill, 2005.

[7] R. Bird, W. Stewart, and E. Lightfoot, *Transport Phenomena*. Wiley, 2007.

[8] H. Fogler, *Essentials of Chemical Reaction Engineering*. Pearson Education, 2010.

[9] W. McCabe, J. Smith, and P. Harriott, *Unit Operations of Chemical Engineering*. McGraw-Hill Education, 2005.

[10] N. Olivier-Maget, G. Hétreux, J. M. Le Lann, and M. V. Le Lann, "Model-based fault diagnosis for hybrid systems: Application on chemical processes," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1617–1630, 2009.

[11] M. Canova, S. Midlam-Mohler, P. Pisu, and A. Soliman, "Model-based fault detection and isolation for a diesel lean trap aftertreatment system," *Control Engineering Practice*, vol. 18, no. 11, pp. 1307–1317, 2010.

[12] D. Freedman, *Statistical Models: Theory and Practice*. Cambridge University Press, 2005.

[13] T. Kourti, "Application of latent variable methods to process control and multivariate statistical process control in industry," *International Journal of Adaptive Control and Signal Processing*, vol. 19, no. 4, pp. 213–246, 2005.

[14] S. Bersimis, S. Psarakis, and J. Panaretos, "Multivariate statistical process control charts: an overview," *Quality and Reliability Engineering International*, vol. 23, no. 5, pp. 517–543, 2007.

[15] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[16] J. Yu, "A particle filter driven dynamic gaussian mixture model approach for complex process monitoring and fault diagnosis," *Journal of Process Control*, vol. 22, no. 4, pp. 778–788, 2012.

[17] D. McRaney, *You Are Not So Smart: Why You Have Too Many Friends on Facebook, Why Your Memory Is Mostly Fiction, and 46 Other Ways You're Deluding Yourself.* Penguin Publishing Group, 2011.

[18] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[19] E. L. Russell, L. H. Chiang, and R. D. Braatz, "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 51, no. 1, pp. 81–93, 2000.

[20] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive pca for adaptive process monitoring," *Journal of Process Control*, vol. 10, no. 5, pp. 471–486, 2000.

[21] Z. Ge and Z. Song, "Distributed pca model for plant-wide process monitoring," *Industrial & Engineering Chemistry Research*, vol. 52, no. 5, pp. 1947–1957, 2013.

[22] S. W. Choi, E. B. Martin, A. J. Morris, and I.-B. Lee, "Fault detection based on a maximum-likelihood principal component analysis (pca) mixture," *Industrial & Engineering Chemistry Research*, vol. 44, no. 7, pp. 2316–2327, 2005.

[23] J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, no. 1, pp. 223–234, 2004.

[24] S. Kittiwachana, D. L. S. Ferreira, G. R. Lloyd, L. A. Fido, D. R. Thompson, R. E. A. Escott, and R. G. Brereton, "One class classifiers for process monitoring illustrated by the application to online hplc of a continuous process," *Journal of Chemometrics*, vol. 24, no. 3-4, pp. 96–110, 2010.

[25] J. Yu and S. J. Qin, "Multimode process monitoring with bayesian inference-based finite gaussian mixture models," *AIChE Journal*, vol. 54, no. 7, pp. 1811–1829, 2008.

[26] J. Yu, "A nonlinear probabilistic method and contribution analysis for machine condition monitoring," *Mechanical Systems and Signal Processing*, vol. 37, no. 1–2, pp. 293–314, 2013.

[27] Y. Masuda, H. Kaneko, and K. Funatsu, "Multivariate statistical process control method including soft sensors for both early and accurate fault detection," *Industrial & Engineering Chemistry Research*, vol. 53, no. 20, pp. 8553–8564, 2014.

[28] Y. Cong, B.-k. Li, X.-g. Yang, Y. Xue, Y.-z. Chen, and Y. Zeng, "Quantitative structure–activity relationship study of influenza virus neuraminidase a/pr/8/34 (h1n1) inhibitors by genetic algorithm feature selection and support vector regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 127, no. 0, pp. 35–42, 2013.

[29] J. Yu, "A bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses," *Computers & Chemical Engineering*, vol. 41, no. 0, pp. 134–144, 2012.

[30] M. S. Escobar, H. Kaneko, and K. Funatsu, "Flour concentration prediction using gapls and gawls focused on data sampling issues and applicability domain," *Chemometrics and Intelligent Laboratory Systems*, vol. 137, no. 0, pp. 33–46, 2014.

[31] J. L. Godoy, R. J. Minari, J. R. Vega, and J. L. Marchetti, "Multivariate statistical monitoring of an industrial sbr process. soft-sensor for production and rubber quality," *Chemometrics and Intelligent Laboratory Systems*, vol. 107, no. 2, pp. 258–268, 2011.

[32] H. Kaneko and K. Funatsu, "A soft sensor method based on values predicted from multiple intervals of time difference for improvement and estimation of prediction accuracy," *Chemometrics and Intelligent Laboratory Systems*, vol. 109, no. 2, pp. 197–206, 2011.

[33] Q. Chen, U. Kruger, M. Meronk, and A. Y. T. Leung, "Synthesis of t2 and q statistics for process monitoring," *Control Engineering Practice*, vol. 12, no. 6, pp. 745–755, 2004.

[34] K. F. K. Hasegawa, Y. Miyashita, “Ga strategy for variable selection in qsar studies: Application of ga-based region selection to a 3d-qsar study of acetylcholinesterase inhibitors,” *Journal of Chemical Information and Modeling*, vol. 37, no. 2, p. 5, 1997.

[35] K. F. M. Arakawa, Y. Yamashita, “Genetic algorithm-based wavelength selection method for spectral calibration,” *Journal of Chemometrics*, vol. 25, no. 1, p. 10, 2010. - John Wiley & Sons, Ltd.

[36] G. Seni and J. Elder, *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan & Claypool Publishers, 2010.

[37] P. Kadlec, B. Gabrys, and S. Strandt, “Data-driven soft sensors in the process industry,” *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.

[38] S. Wold, M. Sjöström, and L. Eriksson, “Pls-regression: a basic tool of chemometrics,” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, 2001.

[39] B. Lin, B. Recke, J. K. H. Knudsen, and S. B. Jørgensen, “A systematic approach for soft sensor development,” *Computers & Chemical Engineering*, vol. 31, no. 5–6, pp. 419–425, 2007.

[40] G. Baffi, E. B. Martin, and A. J. Morris, “Non-linear projection to latent structures revisited (the neural network pls algorithm),” *Computers & Chemical Engineering*, vol. 23, no. 9, pp. 1293–1307, 1999.

[41] H. Kaneko and K. Funatsu, “Adaptive soft sensor model using online support vector regression with time variable and discussion of appropriate parameter settings,” *Procedia Computer Science*, vol. 22, no. 0, pp. 580–589, 2013.

[42] K. Sun, J. Liu, J.-L. Kang, S.-S. Jang, D. S.-H. Wong, and D.-S. Chen, “Development of a variable selection method for soft sensor using artificial neural network and nonnegative garrote,” *Journal of Process Control*, vol. 24, no. 7, pp. 1068–1075, 2014.

[43] S. Xu, S. Yin, R. Srinivasan, and M. Helander, *Proactive Alarms Monitoring using Predictive Technologies*, vol. Volume 31, pp. 1537–1541. Elsevier, 2012.

[44] A. Adhitya, S. F. Cheng, Z. Lee, and R. Srinivasan, *Evaluating the Effectiveness of Anticipatory Alarms for Proactive Process Monitoring*, vol. Volume 32, pp. 565–570. Elsevier, 2013.

[45] F. Harary, *Graph Theory*. Perseus Books, 1994.

[46] H. Kaneko and K. Funatsu, “Nonlinear regression method with variable region selection and application to soft sensors,” *Chemometrics and Intelligent Laboratory Systems*, vol. 121, no. 0, pp. 26–32, 2013.

[47] J. J. Downs and E. F. Vogel, “A plant-wide industrial process control problem,” *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[48] B. Murphy and R. Morrison, *Introduction to Environmental Forensics*. Elsevier Science, 2014.

[49] J. Alonso-Gutierrez, E.-M. Kim, T. S. Batth, N. Cho, Q. Hu, L. J. G. Chan, C. J. Petzold, N. J. Hillson, P. D. Adams, J. D. Keasling, H. Garcia Martin, and T. S. Lee, “Principal component analysis of proteomics (pcap) as a tool to direct metabolic engineering,” *Metabolic Engineering*, vol. 28, no. 0, pp. 123–133, 2015.

[50] D. Raine, P. Langley, E. Shepherd, S. Lord, S. Murray, A. Murray, and J. P. Bourke, “Principal component analysis of atrial fibrillation: Inclusion of posterior ecg leads does not improve correlation with left atrial activity,” *Medical Engineering & Physics*, vol. 37, no. 2, pp. 251–255, 2015.

[51] X. Liu, X. Chen, W. Wu, and Y. Zhang, “Process control based on principal component analysis for maize drying,” *Food Control*, vol. 17, no. 11, pp. 894–899, 2006.

[52] C. Zhao and F. Gao, “Fault-relevant principal component analysis (fpca) method for multivariate statistical modeling and process monitoring,” *Chemometrics and Intelligent Laboratory Systems*, vol. 133, no. 0, pp. 1–16, 2014.

[53] J. Camacho and J. Picó, “Online monitoring of batch processes using multi-phase principal component analysis,” *Journal of Process Control*, vol. 16, no. 10, pp. 1021–1035, 2006.

[54] T. Voegtlin, “Recursive principal components analysis,” *Neural Networks*, vol. 18, no. 8, pp. 1051–1063, 2005.

[55] S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, “Fault detection and identification of nonlinear processes based on kernel pca,” *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 1, pp. 55–67, 2005.

[56] W. Lin, Y. Qian, and X. Li, “Nonlinear dynamic principal component analysis for on-line process monitoring and diagnosis,” *Computers & Chemical Engineering*, vol. 24, no. 2–7, pp. 423–429, 2000.

[57] W. Ku, R. H. Storer, and C. Georgakis, “Disturbance detection and isolation by dynamic principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 30, no. 1, pp. 179–196, 1995.

[58] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[59] C. M. Bishop, M. Svensén, and C. K. I. Williams, “Gtm: The generative topographic mapping,” vol. 10, no. 1, pp. 215–234, 1998.

[60] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[61] E. Hartman, J. D. Keeler, and J. M. Kowalski, “Layered neural networks with gaussian hidden units as universal approximations,” *Neural Comput.*, vol. 2, no. 2, pp. 210–215, 1990.

[62] J. Park and I. W. Sandberg, “Approximation and radial-basis-function networks,” *Neural Comput.*, vol. 5, no. 2, pp. 305–316, 1993.

[63] C. M. Bishop, G. E. Hinton, and I. G. D. Strachan, “Gtm through time,” 1997.

[64] I. Olier, J. Amengual, and A. Vellido, “A variational bayesian approach for the robust analysis of the cortical silent period from emg recordings of brain stroke patients,” *Neurocomputing*, vol. 74, no. 9, pp. 1301–1314, 2011.

[65] A. Vellido, “Missing data imputation through gtm as a mixture of -distributions,” *Neural Networks*, vol. 19, no. 10, pp. 1624–1635, 2006.

[66] A. Vellido, E. Martí, J. Comas, I. Rodríguez-Roda, and F. Sabater, “Exploring the ecological status of human altered streams through generative topographic mapping,” *Environmental Modelling & Software*, vol. 22, no. 7, pp. 1053–1065, 2007.

[67] I. Olier and A. Vellido, “Variational bayesian generative topographic mapping,” *Journal of Mathematical Modelling and Algorithms*, vol. 7, no. 4, pp. 371–387, 2008.

[68] M. Arakawa, T. Miyao, and K. Funatsu, “Development of drug-likeness model and its visualization,” *Journal of Computer Aided Chemistry*, vol. 9, pp. 70–80, 2008.

[69] C. A. Nicolaou and N. Brown, “Multi-objective optimization methods in drug design,” *Drug Discovery Today: Technologies*, vol. 10, no. 3, pp. e427–e435, 2013.

[70] F. Prado-Prado, X. García-Mera, P. Abeijón, N. Alonso, O. Caamaño, M. Yáñez, T. Gárate, M. Mezo, M. González-Warleta, L. Muiño, F. M. Ubeira, and H. González-Díaz, “Using entropy of drug and protein graphs to predict fda drug-target network: Theoretic-experimental study of mao inhibitors and hemoglobin peptides from fasciola hepatica,” *European Journal of Medicinal Chemistry*, vol. 46, no. 4, pp. 1074–1094, 2011.

[71] F. Couenne, C. Jallut, B. Maschke, M. Tayakout, and P. Breedveld, “Structured modeling for processes: A thermodynamical network theory,” *Computers & Chemical Engineering*, vol. 32, no. 6, pp. 1120–1134, 2008.

[72] Y.-C. Lin, L. T. Fan, S. Shafie, B. Bertók, and F. Friedler, “Graph-theoretic approach to the catalytic-pathway identification of methanol decomposition,” *Computers & Chemical Engineering*, vol. 34, no. 5, pp. 821–824, 2010.

[73] R. S. H. Mah, “Application of graph theory to process design and analysis,” *Computers & Chemical Engineering*, vol. 7, no. 4, pp. 239–257, 1983.

[74] E. Cai, D. liu, L. Liang, and G. Xu, “Monitoring of chemical industrial processes using integrated complex network theory with pca,” *Chemometrics and Intelligent Laboratory Systems*, vol. 140, no. 0, pp. 22–35, 2015.

[75] M. S. Escobar, H. Kaneko, and K. Funatsu, “Combined generative topographic mapping and graph theory unsupervised approach for nonlinear fault identification,” *AIChE Journal*, pp. n/a–n/a, 2015.

[76] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[77] S. A. Einwiller and S. Steilen, “Handling complaints on social network sites – an analysis of complaints and complaint responses on facebook and twitter pages of large us companies,” *Public Relations Review*, vol. 41, no. 2, pp. 195–204, 2015.

[78] S. M. Firestone, M. P. Ward, R. M. Christley, and N. K. Dhand, “The importance of location in contact networks: Describing early epidemic spread using spatial social network analysis,” *Preventive Veterinary Medicine*, vol. 102, no. 3, pp. 185–195, 2011.

[79] B. Rienties and E.-M. Nolan, “Understanding friendship and learning networks of international and host students using longitudinal social network analysis,” *International Journal of Intercultural Relations*, vol. 41, no. 0, pp. 165–180, 2014.

[80] P. Gajer, M. T. Goodrich, and S. G. Kobourov, “A multi-dimensional approach to force-directed layouts of large graphs,” *Computational Geometry*, vol. 29, no. 1, pp. 3–18, 2004.

[81] T. M. J. Fruchterman and E. M. Reingold, “Graph drawing by force-directed placement,” *Softw. Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.

[82] Y. F. Hu, “Efficient and high quality force-directed graph drawing,” *The Mathematica Journal*, vol. 10, pp. 37–71, 2005.

[83] M. Bastian, S. Heymann, and M. Jacomy, *Gephi: An Open Source Software for Exploring and Manipulating Networks*. 2009.

[84] R. Diestel, *Graph Theory*. Springer, 2006.

[85] D. Vukičević and G. Caporossi, “Network descriptors based on betweenness centrality and transmission and their extremal values,” *Discrete Applied Mathematics*, vol. 161, no. 16–17, pp. 2678–2686, 2013.

[86] M. Wieling and J. Nerbonne, “Bipartite spectral graph partitioning for clustering dialect varieties and detecting their linguistic features,” *Computer Speech & Language*, vol. 25, no. 3, pp. 700–715, 2011.

[87] M. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E*, vol. 74, no. 3, p. 36104, 2006.

[88] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[89] D. A. Spielman and S.-H. Teng, “Spectral partitioning works: Planar graphs and finite element meshes,” *Linear Algebra and its Applications*, vol. 421, no. 2–3, pp. 284–305, 2007.

[90] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical Review E*, vol. 69, no. 2, p. 026113, 2003. TO BE READ.

[91] H. Shen, *Community Structure of Complex Networks*. Springer Berlin Heidelberg, 2013.

[92] A. Lancichinetti and S. Fortunato, “Limits of modularity maximization in community detection,” *Physical Review E*, vol. 84, no. 6, p. 066122, 2011. PRE.

[93] J. Yu, “Local and global principal component analysis for process monitoring,” *Journal of Process Control*, vol. 22, no. 7, pp. 1358–1373, 2012.

[94] M. Jia, F. Chu, F. Wang, and W. Wang, “On-line batch process monitoring using batch dynamic kernel principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 101, no. 2, pp. 110–122, 2010.

[95] Q. Xu, C. Ding, J. Liu, and B. Luo, “Pca-guided search for k-means,” *Pattern Recognition Letters*, vol. 54, no. 0, pp. 50–55, 2015.

[96] C. M. Bishop, M. Svensén, and C. K. I. Williams, “Developments of the generative topographic mapping,” *Neurocomputing*, vol. 21, no. 1–3, pp. 203–224, 1998.

[97] J. Silvestre-Blanes, “Structural similarity image quality reliability: Determining parameters and window size,” *Signal Processing*, vol. 91, no. 4, pp. 1012–1020, 2011.

[98] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[99] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” *Physical Review E*, vol. 74, no. 1, p. 016110, 2006. PRE.

[100] M. Hamada, A. Wilson, C. Reese, and H. Martz, *Bayesian Reliability*. Springer New York, 2008.

[101] I. Nabney, *NETLAB: Algorithms for Pattern Recognition*. Springer, 2002.

[102] S. Weaver and M. P. Gleeson, “The importance of the domain of applicability in qsar modeling,” *Journal of Molecular Graphics and Modelling*, vol. 26, no. 8, pp. 1315–1326, 2008.

[103] I. Tetko, I. Sushko, A. Pandey, H. Zhu, A. Tropsha, E. Papa, T. Öberg, R. Todeschini, D. Fourches, and A. Varnek, “Critical assessment of qsar models of environmental toxicity against tetrahymena pyriformis: Focusing on applicability domain and overfitting by variable selection,” *J. Chem. Inf. Model.*, vol. 48, no. 9, pp. 1733–1746, 2008.

[104] A. Talevi, M. Goodarzi, E. V. Ortiz, P. R. Duchowicz, C. L. Bellera, G. Pesce, E. A. Castro, and L. E. Bruno-Blanch, “Prediction of drug intestinal absorption by new linear and non-linear qspr,” *European Journal of Medicinal Chemistry*, vol. 46, no. 1, pp. 218–228, 2011.

[105] I. I. Baskin, N. Kireeva, and A. Varnek, “The one-class classification approach to data description and to models applicability domain,” *Molecular Informatics*, vol. 29, no. 8-9, pp. 581–587, 2010.

[106] H. Dragos, M. Gilles, and V. Alexandre, “Predicting the predictability: A unified approach to the applicability domain problem of qsar models,” *Journal of Chemical Information and Modeling*, vol. 49, no. 7, pp. 1762–1776, 2009.

[107] J. A. Fernández Pierna, F. Wahl, O. E. de Noord, and D. L. Massart, “Methods for outlier detection in prediction,” *Chemometrics and Intelligent Laboratory Systems*, vol. 63, no. 1, pp. 27–39, 2002.

[108] R. L. Kodell, B. A. Pearce, S. Baek, H. Moon, H. Ahn, J. F. Young, and J. J. Chen, “A model-free ensemble method for class prediction with application to biomedical decision making,” *Artificial Intelligence in Medicine*, vol. 46, no. 3, pp. 267–276, 2009.

[109] F. Preparata and M. Shamos, *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[110] X. Wang, X. L. Wang, Y. Ma, and D. M. Wilkes, “A fast mst-inspired knn-based outlier detection method,” *Information Systems*, vol. 48, no. 0, pp. 89–112, 2015.

[111] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, “Enhancing density-based clustering: Parameter reduction and outlier detection,” *Information Systems*, vol. 38, no. 3, pp. 317–330, 2013.

[112] H. Kaneko and K. Funatsu, “Estimation of predictive accuracy of soft sensor models based on data density,” *Chemometrics and Intelligent Laboratory Systems*, vol. 128, no. 0, pp. 111–117, 2013.

[113] H. Kaneko and K. Funatsu, “Applicability domain based on ensemble learning in classification and regression analyses,” *Journal of Chemical Information and Modeling*, vol. 54, no. 9, pp. 2469–2482, 2014.

[114] R. Dawkins, *The Greatest Show on Earth: The Evidence for Evolution*. Free Press, 2009.

[115] J. R. Lakowicz, *Principles of Fluorescence Spectroscopy*. Springer London, Limited, 2009.

[116] T. Scheper, B. Hitzmann, E. Stärk, R. Ulber, R. Faurie, P. Sosnitza, and K. F. Reardon, “Bioanalytics: detailed insight into bioprocesses,” *Analytica Chimica Acta*, vol. 400, no. 1–3, pp. 121–134, 1999.

[117] H. Siesler, Y. Ozaki, S. Kawata, and H. Heise, *Near-Infrared Spectroscopy: Principles, Instruments, Applications*. Wiley, 2008.

Appendix A

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 $\Delta Q_{MAT}$
            FOR $j$ = edgeList($i$)
                $\Delta Q_{ij}$ = modularityDelta($i, j$) #Modularity calculation from Equation
2.22
                $\Delta Q_{MAT}(j)$ = $\Delta Q_{ij}$
            END
            IF max $(\Delta Q_{MAT}(j)) > 0$
                ASSIGN i to respective max($\Delta Q_{MAT}(j)$) cluster $j$
                converged = false
            END
        END
        ERASE K empty clusters
        $C = C_{OLD} - K$
    END
    IF $C_{OLD} \neq C$
        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
```

Appendix B

Fluorescence Spectroscopy

Fluorescence is a phenomenon that happens in excited single states, i.e., when there are two paired electrons by opposite spin, one in the excited orbital and the other in the ground-state orbital. Once excited, the electron can rapidly come back to the original state, in emissions that last around 10 ns[115].

All measurements are obtained using a spectrofluorometer, which measures how intensively the medium responses to a range of wavelengths provided by a xenon lamp. Each excitation wavelength affects elements in the medium in different way, resulting in light emitted in different wavelengths, whose intensity is captured by the device. Once all this data is gathered, it is treated and presented as a 2D excitation emission spectrum[116], as shown in Figure B.1. Each pair's intensity corresponds to one input variable. In addition, each flour sample is measured three times, leading to an input matrix that summarizes all information. The rows represent the samples and columns, each emission-excitation pair's intensity.

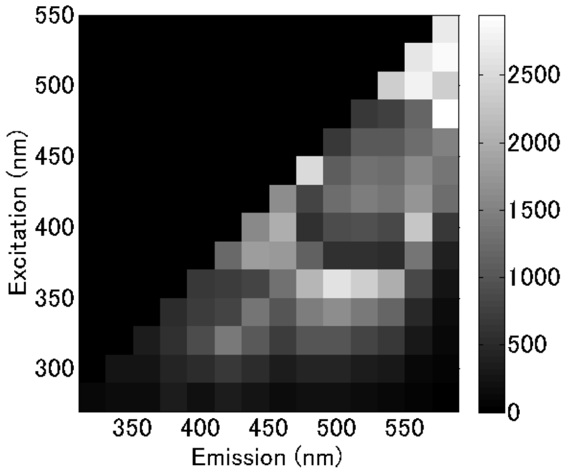

Fig. B.1 Flour sample 2D Fluorescence spectrum.

Appendix C

NIR Spectroscopy

Near-infrared Spectroscopy is a spectroscopy method that evaluates the medium response to near-infrared radiation (800 - 2500 nm). It is not particularly sensitive, but it can easily penetrate on objects, meaning that samples do not need much preparation to be measured[117]. An NIR spectrometer focuses light on a sample, measuring two complementary variables, transmittance and reflectance. The former measures the fraction of incident light that passes through a sample and the latter measures the fraction that reflects light from it. Each NIR spectrum gives clues to the identification of patterns, which are connected to its chemical composition. A typical NIR spectrum can be seen in Figure C.1. Each wavelength's reflectance represents one variable. Similarly to Fluorescence spectroscopy, each flour sample provides NIR spectra that is arranged in an input data matrix.

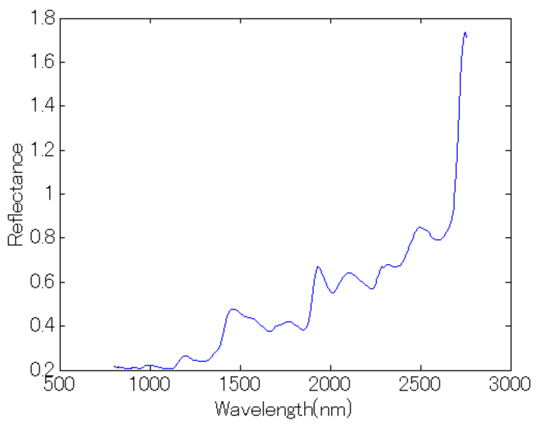

Fig. C.1 NIR spectrum for one flour sample.