Process and alarm data integration under a two-stage Bayesian framework for fault diagnostics

Anna Stief*, James R. Ottewill*, Ruomu Tan**, Yi Cao***

*ABB Corporate Research Center, ul. Starowishna 13a, 31-038 Krakow, Poland
Emails: anna.stief@pl.abb.com; james.ottewill@pl.abb.com
** School of Water, Energy and Environment, Cranfield University, Cranfield, UK MK43 0AL
Email: R.Tan@cranfield.ac.uk
*** College of Chemical and Biological Engineering, Zhejiang University, Hangzhou, 310027, China
Email: caoyi2018@zju.edu.cn

Abstract: Process and alarm data are usually available from industrial processes. It is common practice to use process data for process monitoring and diagnostics. In contrast, alarm data is typically used to determine the instantaneous health state of the process, often as part of a protection system. Alarm data is also used in alarm management systems and for alarm flood detection. Despite the fact that both data types perform similar, although not identical, functions in process monitoring, they are rarely used in combination. One of the main reasons for this is that the fusion between alarm and process data is not trivial: process data is sampled continually and is numerical, while alarm data is binary and appears at discrete times. The two data sources can contain complimentary information regarding the health state of the process, therefore their fusion is a promising direction for fault diagnostics algorithm development.

A two-stage Bayesian framework is proposed to fuse alarm and process data on the decision level for fault diagnostics targeting industrial processes. Instead of the raw process data, the principal components of the process variables are used as the inputs of a naïve Bayes classifier. This step reduces the correlation between the process variables and reduces the dimension of the data. The alarm history is transformed into binary alarm features and input to a second, separate naïve Bayes classifier. The second stage of the method fuses the local classification results of the alarm and process data and provides the final classification result. The results show that the overall performance of the method fusing alarm and process data is superior when compared to the results of a similar single stage method using either alarm or process data.

Keywords: Process monitoring, Data fusion, Diagnostics, Bayes methods, Fault detection, Alarms

1. INTRODUCTION

Due to the ever growing need for safer, more reliable and more predictable industrial processes, the role of process monitoring is becoming increasingly important. The increasing complexity of industrial facilities and the growing requirements for improved efficiency have led to such facilities being equipped with a great number of sensors and monitoring systems. Large quantities of data are available, in the form of process and alarm data, amongst others. As the processes are getting more intricate, traditional model-based approaches are losing popularity compared to data driven methods, which can cope with the added complexity in a more flexible manner (Yin et al., 2015). Process data is commonly used for process monitoring, especially for fault detection and diagnostics. Data-driven process monitoring methods have been developed for early incipient fault detection, for fault classification and for fault prediction (Yang et al., 2013).

Alarms also play an important role in maintaining safe, reliable and efficient operations in industrial plants. They are used to determine the instantaneous health state of the process, often as part of a protection system. Alarm data is also used for process monitoring in alarm management systems and for pattern mining in multiple alarm flood sequences (Lai and Chen, 2017). The results of pattern mining algorithms can provide valuable insights about the health state of the process, can support an operator pinpoint the root cause of a problem and can help in the identification of bad alarm configurations (Lai et al., 2017). Other applications use alarm data to extract fault templates from a set of alarms to identify faults (Charbonnier et al., 2016).

Despite the fact that both process signals and alarms may be used to perform similar, although not identical, functions in process monitoring, they are rarely used in combination. One of the main reasons for this is that process data is numerical and sampled continually, while alarm data is either in the form of a historical log or is in binary format and appears at discrete times. The fusion of such mixed data is not a trivial problem. There are a limited number of works reported in the literature which have attempted to deal with high-dimensional mixed type variables especially in statistical process control (SPC). A density-based statistical process control (SPC) approach is proposed in (Ning and Tsung, 2015).
2012) to use high-dimensional mixed type data for process monitoring based on local outlier factor scheme. Another work (Tuerhong and Kim, 2014) proposed a Gower distance-based multivariate control chart for a mixture of continuous and categorical variables. Whilst both of these works considered data which contained both categorical and continuous numerical process variables, alarms were not considered.

Data fusion from different sources has proven to increase the performance of diagnostics algorithms (Stief et al., 2017). Alarm and process data can contain complimentary information regarding the health state of the process, therefore their fusion can potentially improve the results of fault diagnostics. Algorithms can perform fusion on different abstraction levels: on the data-level, on the feature level or on the decision level (Hall and Llinas, 1997). As alarms are binary and process measurements are numerical their fusion on the data or feature level is problematic, however decision level fusion is a promising direction.

Bayesian methods have been successfully applied to fault detection and diagnosis in process monitoring (Wang and Zhao, 2017). In this paper, a two-stage Bayesian framework is proposed to fuse alarm and process data on the decision level for fault diagnostics targeting industrial processes. The Bayesian framework was selected because it provides modularity for the data fusion and transparency facilitating flexible maintenance decisions (Jaramillo et al., 2017). The method is a reformulation of a previously described two-stage Bayesian framework (Stief et al., 2017), where different signal types were fused to improve the performance of diagnostics for a condition monitoring scenario. The contribution of this paper is to show that the method can also be extended to process monitoring problems, resulting in a framework for fusing heterogeneous alarm and process data. Instead of the raw process data, the principal components of the process variables are used as the inputs of a naïve Bayes classifier, which outputs fault probabilities at the first stage of the method. This step reduces the correlation between the process variables and reduces the dimension of the data. The alarm history is transformed into binary alarm features, which are inputs to another naïve Bayes classifier which also outputs fault probabilities at the first stage of the method. The second, decision-level stage of the method fuses the local probabilities of the alarm and process data and provides the final classification result in the form of probabilities.

The rest of the paper is organized as follows: in the next section the proposed two-stage Bayesian framework for alarm and process data integration is detailed. Then the datasets which were used to validate the method are described alongside details of the implementation of the training, validation and testing of the algorithm. Next, the results are presented with a discussion. Finally, conclusions are given, highlighting the potentials of alarm and process data fusion.

2. METHODS

In this section, an overview of the applied method is given. Firstly, a description of how the Naïve Bayes classifier can be applied to fault classification is given. It is also illustrated how this method can be applied to alarm and process data. Finally, the structure of the two-stage Bayesian framework is given, showing how alarm and process data may be analysed at the local level and how the outcomes of this analysis may be fused to obtain a classification result at a global, decision level.

2.1 Naïve Bayes classifier

For a dataset containing $n$ fault categories $F = \{F_1, F_2, ..., F_n\}$ and $m$ features $Y = \{y_1, ..., y_m\}$, a naïve Bayes classifier can be applied for fault classification, as described in (Jaramillo et al., 2017).

The probability that an observation is classified as fault $F_i$, given the value of a feature $y_k$, can be described in the following Bayesian form

$$P(F_i | y_k) = \frac{P(y_k | F_i)}{P(y_k)} P(F_i),$$

(1)

where $P(y_k | F_i)$ is the likelihood function and $P(F_i)$ is the prior probability. The prediction of the classifier is the index of the Maximum a posteriori (MAP) fault class.

2.2 Alarm data fusion

Alarm data is usually available in the form of historical alarm logs. Schleburg et al. (2013) have previously given an example of an alarm log, including descriptions of the information that may be contained within such a log. In this paper, we define the following terminology to differentiate between the varieties of information contained within the alarm logs. Alarm type is defined as a possible alarm connected to a sensor measurement. The status of an alarm type is considered to indicate whether or not a specific alarm type is active at a given time. An alarm event represents the instance when the status of an alarm type transitions from inactive, or no alarm, to active.

In order to process such alarm data with the Naïve Bayes classifier using equation (1), the alarm logs must be aligned with the associated process data and converted to a binary form. Each alarm type is considered as a separate feature. If the status of an alarm type at a given time is active, then the value of the associated feature is 1, otherwise it is equal to 0. If for a given time stamp there is no record of any alarm events and there are no alarm types with active status, then it is assumed that all of the alarm types have a zero value for that observation. For the exact interpretation of alarms, the alarm management settings have to be taken into consideration.

If the alarm data is in binary form, where $A = \{a_1, a_2, ..., a_d\}$ are the possible alarm features, the local likelihood functions $P(a_i | F_i)$ can be calculated for each fault case $F_i$. The sum of the alarm values are divided by $z$, which is the total number of observations for a fault case $F_i$. 


Unless there is prior knowledge available about the distribution of faults, $P(F)$ is assumed to have a uniform distribution. The probability that an observation is classified as fault $F_a$ given that the alarm features have values $A$ may be written as

$$P(F_a | F) = \sum_{j=1}^{z} P(a_j | F) .$$  \hspace{1cm} (2)

The process data contains measurements from different process sensors. These sensor measurements can be treated as features. However, in order to reduce both the correlation between sensor measurements and the dimensionality of the dataset, it is recommended that Principal Component Analysis (PCA) is applied to the raw process data. PCA can be calculated, for example, using Singular Value Decomposition or Eigenvalue Decomposition (Jolliffe, 2010).

The Naïve Bayes classifier takes the principal components as process features. The proposed implementation of the method for process data fusion does not assume any specific distribution present in the data. Instead, it uses Kernel Density Estimation (KDE), which is a well-established non-parametric method for estimating the probability density function of univariate random processes (Bowman and Azzalini, 1997). KDE determines the cumulative density functions, based on which, fault indicative thresholds are set for the PCs. If a particular PC exceeds its associated threshold, it is considered to be indicating a potential fault. The likelihood functions for each fault case are constructed on the basis of the probability that each PC would cross its respective threshold given that a particular fault category is present. The thresholds are determined by applying KDE to the data recorded under typical operating conditions, where no known fault is present. Thresholds are set symmetrically on the lower and upper end at 2.5% and 97.5% of the cumulative density functions. The probability that a fault is $F_i$, given that a PC $y_k$ has crossed its threshold is calculated using (1), while the probability that an observation is classified as fault $F_i$ is calculated in the same manner as in the case of alarm data, using Eq. (3).

$$P(A | F) = \frac{\sum_{i=1}^{m} a_i}{z}.$$  \hspace{1cm} (3)

The performance of alarm data and process data. If the alarm data predicted $F_a$ and process data predicted $F_p$, then the probability that the final fault prediction is $F_i$ is calculated as

$$P(F_i | F_a, F_p) = \frac{P(F_a | F_i) \cdot P(F_p | F_i) \cdot P(F_i)}{P(F_a) \cdot P(F_p)} .$$  \hspace{1cm} (4)

The final prediction of the algorithm is the index of the Maximum a posteriori (MAP) fault class.

2.4 A two-stage Bayesian framework

The structure of the proposed two-stage Bayesian framework is shown in Figure 1. The aim of the algorithm is to improve the performance of successful fault diagnostics by combining the results of two naïve Bayes classifiers, one for process data and one for alarm data.

![Fig. 1. Flow diagram of the two-stage Bayesian framework](image)

At the first stage there are two Naïve Bayes classifiers, one for the binary alarm features, which are fused after alignment with the process data, and one for the principal components of the process data. The second, global, stage takes the results of the two classifiers and calculates the final fault class prediction using the confusion matrixes in the form of likelihoods.

3. IMPLEMENTATION

In this section an overview is given about a case study dataset and the implementation of the two-stage Bayesian framework on the acquired dataset.
3.1 Dataset

The dataset, which is used to showcase the performance of the algorithm, is from a case study conducted using a multiphase flow facility at the Process System Engineering laboratory of Cranfield University (Stief et al., 2018). The process schematic is shown in Figure 2. Previously, process data obtained from the facility has been used in the development of process monitoring and fault detection techniques (Ruiz-CárceI et al., 2015). In addition to process data, the dataset considered in this paper also contains alarm data. The combined process and alarm data were collected under several flow conditions, with the facility being used to first mix, and then separate, air and water. Data was recorded from multiple sensors each of which were connected to a Supervisory Control and Data Acquisition (SCADA) system. As well as being used to control the process, the SCADA system was also configured to output alarms according to certain criteria being met.

Fig. 2. Schematic of the multiphase flow facility

Datasets were recorded for four different health conditions of the process. Normal process operation was tested under several flow conditions, referred to as $F_0$ in the following sections. Air blockage ($F_1$), air leakage ($F_2$) and diverted flow ($F_3$) were the seeded incipient faults, which were respectively induced by closing V11, opening V10 and opening U39 valves manually. The valves were gradually operated to simulate incipient faults. These faults were tested under two operating conditions “A” and “B” (Table 1). Normal process operation data was also recorded at these operating conditions.

Table 1. Operating conditions

| Operating conditions | “A” | “B” |
|----------------------|-----|-----|
| Air flow rate (Sm$^3$/h) | 120 | 150 |
| Water flow rate (kg/s) | 0.1 | 0.5 |

The process data had 17 different process variables including measurements from flow meters, pressure sensors, temperature sensors and valve positions (Table 2). The process data was sampled at 1 Hz by the SCADA system, the number of observations collected per fault class per operating conditions are shown in Table 3.

Alarm data was logged during the case study, in the form of alarm logs. Each log contained a timestamp of the alarm event, the tag of the sensor, which indicated the alarm and some additional information about the alarm type. Only 5 alarm types appeared in the dataset, these 5 alarms were treated as features and transformed into a binary form. They were also aligned with timestamps of the process data by rounding the time of the alarm event down to the nearest second. As the process is quite slow, this simplification step does not influence the results of the analysis.

Table 2. Process variables

| Sensor tag | Measured process variable | Unit |
|------------|---------------------------|------|
| FT305/302  | Inlet air flow rate       | Sm$^3$/h |
| FT305      | Inlet air temperature     | °C   |
| PT312      | Air delivery pressure     | barg |
| FT102/104  | Inlet water flow rate     | kg/s |
| FT102      | Inlet water temperature   | °C   |
| FT102      | Inlet water density       | kg/m$^3$ |
| PT417      | Pressure in the mixing zone | barg |
| PT408      | Pressure at the riser top | barg |
| PT403      | Pressure in the top separator | barg |
| FT404      | Top separator output air flow rate | m$^3$/h |
| FT406      | Top separator output water flow rate | kg/s |
| PT501      | Pressure in the 3-phase separator | barg |
| PIC501     | Air outlet valve 3-phase separator | % |
| LS502      | Water-oil 3-level separator | % |
| LS503      | Water coalescer level     | % |
| LVC502-SR  | Water coalescer outlet valve | % |
| LL101      | Water tank Level          | m   |

Table 3. Number of observations per fault class

| Fault class         | Operating conditions | “A” | “B” |
|---------------------|----------------------|-----|-----|
| $F_0$ (Normal)      |                      | 2206| 2352|
| $F_1$ (Blockage)    |                      | 3544| 4183|
| $F_2$ (Leakage)     |                      | 2881| 5389|
| $F_3$ (Diverted flow) |                    | 7320| 4199|

Except for the observations listed in Table 3, additional operating conditions were tested to collect more normal data, which resulted in an extra 3937 observation for $F_0$.

3.2 Training, validation and testing

The dataset which is used for the training of the first stage of the algorithm is called the training set. The dataset used for
the training of the second stage of the algorithm is called the validation set and the dataset used to test the performance of the algorithm is called the test set. These three datasets must be separate. A crucial point when implementing the method is to select these datasets in the most appropriate way. In a practical industrial system, the amount of data under normal process operation is usually much greater than the amount of data which is available during periods of faulty operation. To obtain more precise threshold values, it is suggested to incorporate healthy data under different flow conditions. The thresholds were trained using the dataset, which was collected under several different operating conditions under the normal process operation. The datasets for each fault case were split randomly, 60% of the data was selected to the training set, and 20-20% of the data were put into the validation set and test set. The division of the data was applied separately for operating condition A and B.

3.3 Optimization of performance

In addition to the proper selection of the training set, validation set and test set, the performance of the algorithm is also dependent on the prior distribution of faults, the thresholds and on how many principal components are selected as features for the process variables. All of these can highly influence the correct prediction rate of the algorithm. In this case study there was no prior information available about the facility: the prior distribution of the fault cases were set to uniform, so the number of observations from the different fault classes does not influence the results of the Naive Bayes classifiers. The thresholds were set according to a standard 95% confidence interval, however for other datasets the optimal threshold value can vary. An optimization step can solve this issue evaluating the performance of the validation set at different threshold values and choosing the threshold which gives the best performance. A way to measure the performance of the algorithm is to summarise the correct prediction rate of the confusion matrix of the process data. A similar approach can be applied to find the optimal number of principal components. In this case study the number of principal components were determined by taking the first \( N=9 \), such that 99.9% of the total variance of the dataset is concentrated in the first \( N \) principal components.

4. RESULTS AND DISCUSSION

As previously noted, the dataset was randomly split to training set, validation set and test set. This random split was applied 100 times with the results from all 100 data divisions being averaged to generate a final result. Averaged results are shown below for the two operating conditions “A” and “B”. The process and alarm results of the classifiers are compared with the fused results. The advantages and limitations of each data type and the two-stage Bayesian classifier are also discussed.

4.1 Operating condition “A”

Tables 4 and 5 show the results of the applying the first data fusion stage for alarm data and process data respectively. Data from Operating Condition “A” is considered. Results are presented in the form of confusion matrixes, where the columns represent the conditions diagnosed by the algorithm and the rows represent the actual conditions. The diagonal contains the correct classification values. For clarity the fault cases are listed again: \( F_0 \): Normal process operation, \( F_1 \): Air blockage, \( F_2 \): Air leakage, \( F_3 \): Diverted flow.

Table 4. Alarm data, operating condition “A”

| Diagnosed Condition | \( F_0 \) | \( F_1 \) | \( F_2 \) | \( F_3 \) |
|---------------------|-----------|-----------|-----------|-----------|
| \( F_0 \)           | 0,41      | 0,59      | 0,00      | 0,00      |
| \( F_1 \)           | 0,00      | 0,95      | 0,03      | 0,02      |
| \( F_2 \)           | 0,06      | 0,21      | 0,73      | 0,00      |
| \( F_3 \)           | 0,00      | 0,72      | 0,25      | 0,03      |

Table 4 shows the results obtained after applying the first data fusion stage for alarm data for operating condition. An average rate of correct classification may be calculated as the average of the values in the diagonal of the confusion matrix. Note that this simple metric assumes that all faults incur an equal cost, which in practice typically is not the case (one fault mode may lead to more severe penalties in terms of cost or safety). The metric is included in order to easily and quantifiably compare the performance of the algorithm for the different test cases; for greater insight into the performance of the algorithm for specific fault modes, readers should refer to the confusion matrices.

For operating condition “A” the average rate of correct classification was 53%. This observation shows that in general, for operating condition “A”, the alarm data did not contain enough information to adequately classify the faults. This is primarily due to the limited number of alarms fired during the case study. However the algorithm correctly classified the air blockage fault \( F_1 \) in 95% of the cases. The false alarm rate was only 2%, but the 59% missed alarm rate was significantly higher. The misclassification between faults was also high, especially in case of \( F_3 \) (diverted flow), where the misclassification rate was 97%.

Table 5. Process data, operating condition “A”

| Diagnosed Condition | \( F_0 \) | \( F_1 \) | \( F_2 \) | \( F_3 \) |
|---------------------|-----------|-----------|-----------|-----------|
| \( F_0 \)           | 0,30      | 0,16      | 0,25      | 0,29      |
| \( F_1 \)           | 0,03      | 0,97      | 0,00      | 0,00      |
| \( F_2 \)           | 0,11      | 0,00      | 0,89      | 0,00      |
| \( F_3 \)           | 0,03      | 0,00      | 0,00      | 0,97      |

...
Table 5 shows the results obtained after applying the first data fusion stage for process data for operating condition “A”. The average rate of correct classification was 78% for the results of the process data. The algorithm was correct in 97% of the cases for F₁ and F₂. The results also indicate a relatively high classification rate for F₃ with 89%. The false alarm rate was around 6%, but the missed alarm rate was even higher than for the alarms with a 70% rate.

Table 6. Fused data, operating condition “A”

| Diagnosed Condition | F₀ | F₁ | F₂ | F₃ |
|---------------------|----|----|----|----|
| Actual condition    |    |    |    |    |
| F₀                  | 0.54| 0.16| 0.00| 0.29|
| F₁                  | 0.03| 0.97| 0.01| 0.00|
| F₂                  | 0.06| 0.00| 0.94| 0.00|
| F₃                  | 0.02| 0.00| 0.01| 0.97|

Table 6 shows the results obtained after performing the second data fusion stage, combining the results of both alarm and process data. The average rate of correct classification was 85% for the fused data for operating condition “A”. This is a 7% performance improvement compared to the averaged results of the process data and a 32% improvement compared to the results of the alarm data. The false alarm rate was around 4%, while the missed alarm rate was 36%. The misclassification rate between faults was insignificant. The correct classification rates were at least as good as the individual results obtained from the alarm and process data individually, and in many cases were improved.

4.2 Operating condition “B”

Table 7 shows the results obtained after applying the first data fusion stage for alarm data for operating condition “B”. The average rate of correct classification was 48.5% for the results of the alarm data for operating condition “B”. These results are similar to the alarm data for operating condition “A”, again indicating that the alarm data did not contain enough information to adequately classify the faults. The false alarm rate was around 2% and the missed alarm rate is 80%. The results indicate a 100% correct classification rate for F₁ (diverted flow), while there is a high misclassification rate of fault for F₁ and F₂.

Table 7. Alarms data, operating condition “B”

| Diagnosed Condition | F₀ | F₁ | F₂ | F₃ |
|---------------------|----|----|----|----|
| Actual condition    |    |    |    |    |
| F₀                  | 0.20| 0.02| 0.18| 0.59|
| F₁                  | 0.00| 0.53| 0.07| 0.40|
| F₂                  | 0.07| 0.18| 0.23| 0.54|
| F₃                  | 0.00| 0.00| 0.00| 1.00|

Table 8 shows the results obtained after applying the first data fusion stage for process data for operating condition “B”. The average rate of correct classification was 82% for the results of the process data. The algorithm was correct in 96% of the cases for F₁ and had a relatively high classification rate of 88% for F₂. The false alarm rate was around 5% and the missed alarm rate was 25%. The process data performed better compared to operating condition “A” with regards to lower false and missed alarm rates, however the misclassification between faults increased especially for F₂ with 23%.

Table 8. Process data, operating condition “B”

| Diagnosed Condition | F₀ | F₁ | F₂ | F₃ |
|---------------------|----|----|----|----|
| Actual condition    |    |    |    |    |
| F₀                  | 0.75| 0.12| 0.96| 0.07|
| F₁                  | 0.04| 0.96| 0.00| 0.00|
| F₂                  | 0.07| 0.02| 0.70| 0.21|
| F₃                  | 0.05| 0.00| 0.07| 0.88|

The average rate of correct classification was 89% for the fused data for operating condition “B” shown in Table 9. This is a 7% performance improvement compared to the averaged results of the process data and a 40.5% improvement compared to the results of the alarm data. The false alarm rate was around 4%, while the missed alarm rate was 15%. The misclassification rate between faults was also decreased relative to the results of alarm and process data. The correct classification rates were better than or as good as the results of the alarm and process data, except for the case of F₂ in the results of the alarm data. Although the alarm data was 100% correct in classifying F₁, the algorithm also classified F₀, F₁ and F₂ in almost half of the cases as F₃, which reduced the confidence in the alarm data for F₂ at the global fusion stage.

Table 9. Fused data, operating condition “B”

| Diagnosed Condition | F₀ | F₁ | F₂ | F₃ |
|---------------------|----|----|----|----|
| Actual condition    |    |    |    |    |
| F₀                  | 0.85| 0.02| 0.08| 0.05|
| F₁                  | 0.03| 0.97| 0.00| 0.00|
| F₂                  | 0.07| 0.02| 0.88| 0.03|
| F₃                  | 0.05| 0.00| 0.07| 0.88|

The fused results improved the performance of the algorithm under both operating conditions with an average improvement of 7%. The alarm data was not reliable for fault classification for both operating conditions, while the process data gave a better insight. The fused results show that even in the case of insufficient alarm data the method is able to improve the performance of the fault classification relying only on the process data.
In this paper a two-stage Bayesian framework for fusing alarm and process data for fault detection and classification for process monitoring purposes was proposed. In a first, local stage, alarm and process data are fused independently using a Naïve Bayes classifier. The second stage of the method fuses the local classification results of the alarm and process data, providing a final classification result. Fusing both process and alarm data together in this second, global stage, was shown to improve the performance of the algorithm by correctly classifying faults relative to considering each data type independently. Under two different operating conditions, it was shown that fusing both process and alarm data together was able to improve the classification performance of the algorithm by an average value of 7%. The results showed that even in the case of only a few alarm events occurring, the proposed method for process and alarm fusion is able to improve the performance of the fault classification relative to the case of only considering process data.

ACKNOWLEDGEMENT

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675215, for which the authors are grateful.

REFERENCES

Bowman, A.W., Azzalini, A., 1997. Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations. Clarendon Press.

Charbonnier, S., Bouchair, N., Gayet, P., 2016. Fault template extraction to assist operators during industrial alarm floods. Eng. Appl. Artif. Intell. 50, 32–44. doi:10.1016/j.engappai.2015.12.007

Hall, D.L., Llinas, J., 1997. An Introduction to Multisensor Data Fusion. Proc. IEEE 85, 6–23.

Jaramillo, V.H., Ottewill, J.R., Dudek, R., Lepiareczyk, D., Pawlik, P., 2017. Condition monitoring of distributed systems using two-stage Bayesian inference data fusion. Mech. Syst. Signal Process. 87, 91–110. doi:10.1016/j.ymssp.2016.10.004

Jolliffe, I.T., 2010. Principal component analysis. Springer.

Lai, S., Chen, T., 2017. A method for pattern mining in multiple alarm flood sequences. Chem. Eng. Res. Des. 117, 831–839. doi:10.1016/j.cherd.2015.06.019

Lai, S., Yang, F., Chen, T., 2017. Online pattern matching and prediction of incoming alarm floods. J. Process Control 56, 69–78. doi:10.1016/j.jprocont.2017.01.003

Ning, X., Tsung, F., 2012. A density-based statistical process control scheme for high-dimensional and mixed-type observations. IIE Trans. (Institute Ind. Eng. 44, 301–311. doi:10.1080/0740817X.2011.587863

Ruiz-Cárceles, C., Cao, Y., Mba, D., Lao, L., Samuel, R.T., 2015. Statistical process monitoring of a multiphase flow facility. Control Eng. Pract. 42, 74–88. doi:10.1016/j.conengprac.2015.04.012

Schleburg, M., Christiansen, L., Thornhill, N.F., Fay, A., 2013. A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system. J. Process Control 23, 839–851. doi:10.1016/j.jprocont.2013.03.010

Stief, A., Ottewill, J.R., Orkisz, M., Baranowski, J., 2017. Two stage data fusion of acoustic, electric and vibration signals for diagnosing faults in induction motors. Elektron. ir Elektrotechnika 23, 19–24.

Stief, A., Tan, R., Cao, Y., Ottewill, J.R., 2018. Analytics of Heterogeneous Process Data : Multiphase Flow Facility Case Study, in: 10th IFAC Symposium on Advanced Control of Chemical Processes.

Tuerhong, G., Kim, S.B., 2014. Gower distance-based multivariate control charts for a mixture of continuous and categorical variables. Expert Syst. Appl. 41, 1701–1707. doi:10.1016/j.eswa.2013.08.068

Wang, Y., Zhao, C., 2017. Probabilistic fault diagnosis method based on the combination of nest-loop fisher discriminant analysis and analysis of relative changes. Control Eng. Pract. 68, 32–45. doi:10.1016/j.conengprac.2017.07.009

Yang, Z., Wang, J., Chen, T., 2013. Detection of correlated alarms based on similarity coefficients of binary data. IEEE Trans. Autom. Sci. Eng. 10, 1014–1025. doi:10.1109/TASE.2013.2248000

Yin, S., Li, X., Gao, H., Kaynak, O., 2015. Data-based techniques focused on modern industry: An overview. IEEE Trans. Ind. Electron. 62, 657–667. doi:10.1109/TIE.2014.2308133