Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale, Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Development of anomaly detection models independent of noise and missing values using graph Laplacian regularization

Yuna Tahashi[a], and Koichi Fujiwara[a]

[a] Department of Materials Process Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
* Corresponding Author: K. Fujiwara (fujiwara.koichi@hps.material.nagoya-u.ac.jp).

ABSTRACT

Anomaly detection is a key technique for maintaining process suitability and safety; however, the quality of process data often deteriorates due to missing or noisy values caused by sensor malfunctions. Such data imperfections may obscure real faults. If anomaly detection models are too sensitive to such abnormal data, they may cause false positives resulting in unnecessary alarms, which may obstruct detection of true process faults. Thus, deterioration of the quality of process data may affect process performance and safety. We propose a new anomaly detection method that utilizes graph Laplacian regularization as a loss function considering data-specific temporal relationships. Graph Laplacian regularization is a mathematical tool used in image processing and denoising to smooth data. We assume that successive process data temporally close to each other have similar values and maintain temporal dependencies among variables. In this study, Laplacian regularization imposes significant penalties when the outputs of neighboring samples lose smoothness, under the assumption that neighboring samples keep similar relationships. Such temporal dependencies can be expressed as a graph structure and extracted with the Nearest Correlation (NC) method. To demonstrate the usefulness of the proposed anomaly detection method, we applied it to an anomaly detection problem in a vinyl acetate monomer (VAM) process. The results show that the model with graph Laplacian regularization achieved higher performance than without graph Laplacian regularization in some fault scenarios. It was confirmed that the proposed method is effective for anomaly detection.

Keywords: Anomaly detection, Graph Laplacian regularization, Autoencoder, vinyl acetate monomer process

INTRODUCTION

Process data frequently suffer from imperfections such as missing values or measurement noise due to sensor malfunction. Such data imperfections pose a significant challenge to detection of process anomalies and can lead to false positives or missed rare fault events. Anomaly detection models with high sensitivity may over-detect these faults, leading to interference in detection of true failure events.

Various anomaly detection methods have been developed to detect such faults, such as One-Class Support Vector Machine (OCSVM) and Multivariate Statistical Process Control (MSPC). Although these methods are common and widely used, they are not always applicable for handling complicated fault events due to incompatibility with time-series data or linear models [1,2].

An autoencoder (AE) is a nonlinear expansion of principal component analysis (PCA) and is a type of neural network trained so that the output becomes close to the input. Since MSPC is based on PCA, AE can also be used for anomaly detection [3]. If an AE is trained only on normal process data, a fault event can be detected when reconstruction error (RE) between the input and the output of AE becomes large.

It is, however, difficult for AE, as well as other conventional anomaly detection methods, to appropriately cope with sudden changes in process data, which may cause false alarms and obstruct detection of true fault events in processes. Thus, deterioration of the quality of process data may affect process performance and safety.

In order to suppress such false alarms and missed fault events caused by sudden changes in data and to improve process suitability and safety, a regularization term in addition to a loss function can be introduced to a training process of an AE model [4].

A graph-based regularization term expresses the relationship between process variables and imposes a penalty when the relationship changes between input and output, which is expected to emphasize abnormal changes in the process data. Graph Laplacian regularization (GLR) has attracted attention for its ability to smooth data, particularly for denoising image data [5], as a simple and useful method that can make the input data according to a graph expressing definitions of smoothness [6-8]. For example, the distance between pixels of image data can be defined as data smoothness. GLR can be applied not only to image data but to any other type of data as long as the data can be expressed as a graph.

In this study, we utilize GLR for anomaly detection to suppress false alarms and missed fault events caused by sudden changes in process data. We propose a new anomaly detection method based on AE with GLR. In the proposed method, a graph-based regularization term is introduced when training an AE model. In addition, the graph-based regularization term can take the smoothness of data expressed as a graph structure.

In order to use GLR, an appropriate graph needs to be constructed. The proposed method adopts, as a graph construction method, the Nearest Correlation (NC) method, which is a structural learning algorithm that considers correlations between variables [9]. Using GLR based on the NC method, it is expected that missed measurements and noise contained in the process data are automatically corrected in terms of correlations between variables, and only significant changes in process characteristics such as fault events are detected.

The use of GLR with the NC method would realize a reliable anomaly detection model that can improve the efficiency and reliability of process monitoring and control in various industrial applications.

In this study, the usefulness of the proposed anomaly detection method is demonstrated through its application to simulation data of a vinyl acetate monomer (VAM) production process, generated by a process simulator provided by Omega Simulation Co., Ltd [10,11].

METHODS

Graph Laplacian regularization

The Laplacian is a mathematical operator that measures the degree of curvature or change in a function, quantifying how different the value of a function is from its surroundings.

The Laplacian operator can be extended to graphs, in which case it is called the graph Laplacian, which is used to describe relationships between the nodes of a graph, and in particular to measure the smoothness of features between neighboring nodes. The graph Laplacian $\boldsymbol{L}$ can be defined as

$$\boldsymbol{L} = \Delta - \boldsymbol{A} \tag{1}$$

where $\boldsymbol{A}$ is the adjacency matrix and $\Delta = \sum_j \boldsymbol{A}_{ij}$ is the order matrix.

GLR is a technique that uses the graph Laplacian to impose smoothness constraints on a model. Particularly, it allows the model to take the graph structure into account by adding a penalty so that the differences in output values between neighboring nodes become small. Although GLR has been mainly used in image processing, it can be used for any type of data if the relationship among variables can be expressed as a graph, such as similarity among variables, rules that variables must follow, or temporary relationships among variables in time-series data.

It is assumed that the numbers of data samples and variables are N and M, respectively, $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^N$ are the ith and the jth variables of the data matrix $\boldsymbol{X} \in \mathbb{R}^{N\times M}$, and A_{ij} is the similarity or the strength of connection between x_i and x_j. Here, A_{ij} is the (i, j) element of the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{M\times M}$ of the graph. For the predictions $\boldsymbol{y}_i, \boldsymbol{y}_j \in \mathbb{R}^N$ from $\boldsymbol{x}_i, \boldsymbol{x}_j$ and the matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times M}$ containing them, the regularization term R based on GLR is defined by the following equation:

$$\begin{aligned} R = \frac{1}{2}\sum_{i,\, j=1}^{M} A_{ij}\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2 = \sum_{i=1}^{M} \boldsymbol{y}_i\, \boldsymbol{y}_i \Delta_{i,i} - \sum_{i=1}^{M} \boldsymbol{y}_i \boldsymbol{y}_j A_{ij} \\ = \mathrm{Tr}(\boldsymbol{Y}\Delta\boldsymbol{Y}^T) - \mathrm{Tr}(\boldsymbol{YAY}^T) \\ = \mathrm{Tr}(\boldsymbol{YLY}^T) \end{aligned} \tag{2}$$

By adding this regularization term to the loss function when training the AE model, the similarity among the data can be considered to be within the model.

Nearest Correlation method (NC method)

To use GLR, a suitable graph needs to be constructed from the data. In this study, we utilize the NC method.

The NC method was originally proposed as a way to detect multiple samples that have similar correlations with the target sample. An illustrated example of the NC method is shown in Figure 1.

N samples are given and a zero matrix $\boldsymbol{S} \in \mathbb{R}^{N\times N}$ is set up. Here, a target sample is $\boldsymbol{x_1} \in \mathbb{R}^M$. First, $\boldsymbol{x_1}$ is translated to the origin of the space, that is, $\boldsymbol{x_i^{'}} = \boldsymbol{x_i} - \boldsymbol{x_1}$ and $\boldsymbol{x_1^{'}} = 0$. Then, correlation coefficients are calculated for all possible pairs of samples except $\boldsymbol{x_1^{'}}$, such as $\boldsymbol{x_2} - \boldsymbol{x_3}, \boldsymbol{x_2} - \boldsymbol{x_4}, \cdots$. A weight of 1 is assigned to the pair when its correlation coefficient exceeds a predefined threshold of γ. For example, in Fig. 1, samples $\boldsymbol{x_2}$ and $\boldsymbol{x_6}$, $\boldsymbol{x_3}$ and $\boldsymbol{x_4}$ correspond to pairs with high correlation coefficients. That is, 1 is added to $\boldsymbol{S_{2,6}}$, $\boldsymbol{S_{6,2}}$, $\boldsymbol{S_{3,4}}$ and $\boldsymbol{S_{4,3}}$. This process is repeated until all samples have become the target sample. Finally, the matrix $\boldsymbol{S}$ represents the similarity among samples from the viewpoint of correlation. In this study, the obtained matrix $\boldsymbol{S}$ has integer values greater than or equal to 0, which is then transformed into an un-weighted matrix consisting only of elements 0 and 1, with all positive values being 1. The resulting matrix is used as the adjacency matrix $\boldsymbol{A}$ of the graph for GLR.

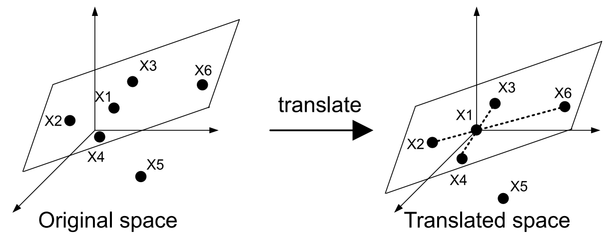
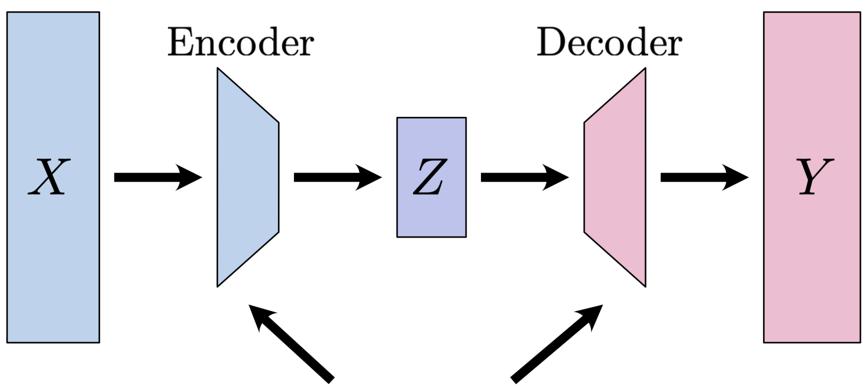


Figure 1. Image of NC method

Proposed method (NCGLR-AE)

In this study, we propose a new anomaly detection method, referred to as NCGLR-AE, which uses GLR and



$$\text{loss} = \text{mean}\left((\mathbf{X}-\mathbf{Y})^2\right) + \alpha R$$

Figure 2: Schematic diagram of the proposed anomaly detection method

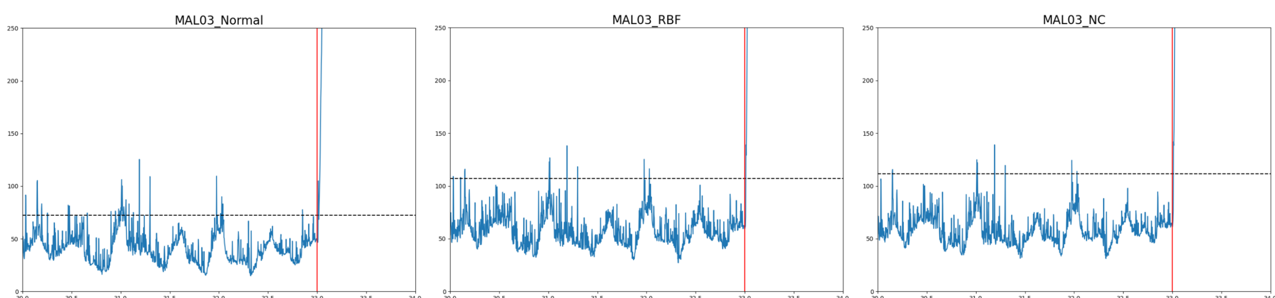


Figure 3: RE in VAM process with MAL3

(left: Normal, center: RBF, right: NC)

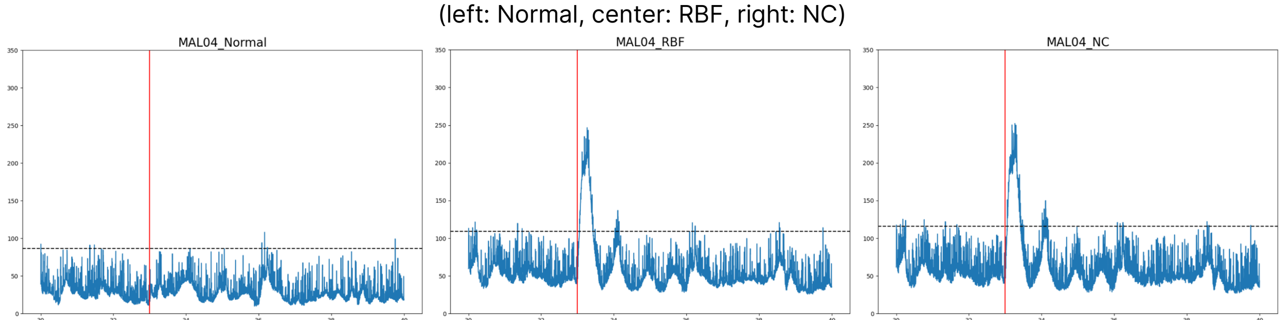


Figure 4: RE in VAM process with MAL4
(left: Normal, center: RBF, right: NC)

NC methods during AE training. A schematic diagram of the proposed method is shown in Figure 2. In AE, the input $\boldsymbol{X}$ compress the information into latent variables $\boldsymbol{Z}$ via an encoder. $\boldsymbol{Z}$ is then restored to output $\boldsymbol{Y}$ via a decoder. A graph is created by the NC method with the data variables as nodes from the data matrix $\boldsymbol{X}$. It can be transformed into the GLR R and the AE training is performed with the loss function, defined as

$$\text{loss} = \text{mean}((\boldsymbol{X} - \boldsymbol{Y})^2) + \alpha R \tag{3}$$

where α is a regularization weight.

RE between the input and the output of the trained AE is defined as the anomaly score. A control limit is set as the value at which β % is normal with respect to the abnormality score of the training data consisting of normal data only, and the abnormality score calculated from process data. The data are regarded as abnormal only when the abnormality score exceeds the control limit.

The model training procedure of the proposed method is described as follows:

1. Divide the training data $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ into training and validation data so that the order of the time-series does not change.
2. Extract data $\boldsymbol{X}_t \in \mathbb{R}^{B\times M}$ for a batch size B.
3. Calculate adjacency matrix $\boldsymbol{A}_t \in \mathbb{R}^{M\times M}$ using the NC method for the variable direction of $\boldsymbol{X}_t$.
4. Input $\boldsymbol{X}_t$ into AE and obtain output $\boldsymbol{Y}_t \in \mathbb{R}^{B\times M}$.
5. Calculate $\boldsymbol{L}_t \in \mathbb{R}^{M\times M}$ from $\boldsymbol{A}_t$ and GLR R_t from $\boldsymbol{L}_t, \boldsymbol{Y}_t$.
6. Calculate the loss function of AE from $\boldsymbol{X}_t$.
7. Extract next data for the next batch size.
8. Repeat steps 3-7 until the validation loss converges.
9. Define the value at which β % of the abnormality scores for the training data are normal as the control limit.

CASE STUDY

Dataset

Simulated process data generated from the vinyl acetate monomer (VAM) process model was used to assess the performance of the anomaly detection model trained by the proposed method. Forty hours of the VAM process operation data were generated, which included situations where sudden malfunctions occurred 33 hours after the start of measurement. The first 30 hours of data were used for training and the remaining 10 hours for testing. To validate the performance of the model for different types of faults, test data were prepared, including six different faults: faults MAL01 and MAL02, which reduced the feed composition of feedstock ethylene and acetic acid, respectively; MAL03 and MAL04 are malfunctions that reduced the feed pressure of the feedstock ethylene and oxygen, respectively; MAL09 and MAL10 are faults that changed the temperature and pressure of the cooling water, respectively. Sixty-two process variables in the VAM process, measured every 10 seconds, include pressure and flow rate values.

How to train models and determine thresholds

We employed an autoencoder (AE) as the model for anomaly detection, with a one-layer encoder and a one-layer decoder, and the number of units in the middle layer (30) and the batch size (100) were kept constant for all comparison models.

Taking advantage of the characteristics of time-series data, the graph was constructed under the assumption that data close in time have close values. To calculate the correlations between variables of temporally close data, a graph was constructed using the NC method with the variables as nodes. The GLR was calculated from the adjacency matrix of the graph and added to the loss function during batch training of AE. When validation loss did not decrease for five consecutive times, loss was considered to have converged and AE training was terminated.

In this study, we used "reconstruction error (RE)" as the anomaly score. RE is the square of the error between the input and output values.

In order to compare the performance differences between the graph construction methods, a Gaussian (Radial Basis Function; RBF) kernel, which calculates similarity based on the distance between two points, was employed in addition to the NC method. The Gaussian kernel is calculated by the following equation:

$$A_{ij} = \begin{cases} 1, & \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) > 0.99 \\ 0, & \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \leq 0.99 \end{cases} \tag{4}$$

where σ is a hyperparameter that adjusts how far away data can be and still be considered similar; if σ is large, a wide range of data points are considered similar. In this case, σ was set to 10.

Results

In this study, we used "False Alarm Rate (FAR)", "Missing Alarm Rate (MAR)" and "Detection Latency (DL)" as evaluation indicators. The definitions of FAR and MAR are given in the following equations:

$$FAR = \frac{False\ Positive}{Number\ of\ Normal\ Samples} \times 100 \tag{5}$$

$$MAR = \frac{False\ Negative}{Number\ of\ Anomalous\ Samples} \times 100 \tag{6}$$

FAR indicates the percentage of samples judged to be abnormal in the normal data before a fault event

Table 1: Anomaly detection results for each anomaly scenario. The best value for each test data is shown in bold.

Malfunction	Model	FAR [%]	MAR [%]	DL [s]
MAL01	-	**0.278**	0.198	52
	RBF	0.648	**0.159**	**42**
	NC method	0.648	**0.159**	**42**
MAL02	-	0.926	0.397	103
	RBF	0.463	**0.357**	**83**
	NC method	**0.370**	0.397	103
MAL03	-	3.70	0.198	44
	RBF	0.926	**0.159**	44
	NC method	**0.741**	**0.159**	44
MAL04	-	**0.278**	99.8	11122
	RBF	0.648	94.1	**212**
	NC method	0.648	**93.8**	**212**
MAL09	-	0.0926	0.476	123
	RBF	0.0926	0.397	102
	NC method	0.0926	**0.357**	**93**
MAL10	-	0.185	**1.67**	**412**
	RBF	0.0926	1.74	421
	NC method	**0.0**	1.79	452

occurred. The lower this indicator, the fewer false detections of abnormalities. MAR indicates the percentage of anomaly data that are incorrectly determined as normal after a fault event occurred. A lower MAR means that anomaly is correctly detected. DL is the time required from the occurrence of a fault event to its detection. DL should be small from the viewpoint of early detection of faults.

Anomaly detection results for each anomaly scenario are shown in Table 1. The best value for each set of test data is shown in bold.

The models with GLR demonstrate significant performance improvements in many anomaly patterns, particularly when graphs were constructed using the NC method. For MAL02, MAL03, and MAL10, the number of false positives was reduced by applying GLR. This indicates that unnecessary alarms were appropriately suppressed, which is expected to reduce operator burden and improve work efficiency.

Figure 3 shows an enlarged view of the MAL03 abnormal score around the normal data. There are several points where the threshold value is exceeded only when GLR is not applied. One possible reason is that the anomaly score fluctuates significantly due to noise when GLR is not applied. This suggests that GLR was effective in removing noise from process data, and that noise removal may enable stable calculation of abnormality scores.

For MAL04 and MAL09, the smallest MAR was achieved with the NC method. Figure 4 shows the MAL04 anomaly score, indicating that the fault event was incorrectly detected without GLR. On the other hand, GLR accurately detected the occurrence of the faults. It is concluded that consideration for the correlation between variables during the training process enabled the accurate detection of fault events. In addition, these results also showed that GLR is useful for time-series data processing.

MAL01 and MAL10, however, had performances that were the same regardless of GLRs, or even better when GLRs were not used. As such, there may be room for improvement in the proposed model.

CONCLUSION

In this study, an anomaly detection model incorporating graph Laplacian regularization created by the NC method is proposed. Based on the process that data points close in time have close values, it was possible to construct an anomaly detection model that utilizes the properties of time-series data.

The addition of the regularization term successfully improved the anomaly detection rate for many sets of anomaly simulation data compared to cases without the regularization term, and the correlation-aware NC method was effective in detecting anomalies with higher accuracy.

In the future, validation against real data and verification of accuracy will be carried out in order to enable more accurate anomaly detection. In addition, although this study was only validated on AE, future validation will be done on other deep-learning models.

REFERENCES

1. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural Comput 13:1443-1471 (2001). https://doi.org/10.1162/089976601750264965
2. MacGregor JF, Kourti T. Statistical process control of multivariate processes. Control Eng Pract 3:403-414 (1995). https://doi.org/10.1016/0967-0661(95)00014-L
3. Venkatasubramanian, V., and King Chan. A neural network methodology for process fault diagnosis. AIChE Journal, 35(12): 1993-2002 (1989). https://doi.org/10.1002/aic.690351210
4. Bergmann P, Löwe S, Fauser M, Sattlegger D, Steger C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In: Proc 14th Int Joint Conf Comput Vis Imaging Comput Graph Theory Appl (VISIGRAPP), Vol 5: VISAPP. SciTePress 372-380 (2019). https://doi.org/10.5220/0007364503720380
5. Zeng J, Pang J, Sun W, Cheung G. Deep graph Laplacian regularization for robust denoising of real images. arXiv preprint arXiv:1807.11637 [cs.CV] (2018). https://doi.org/10.48550/arXiv.1807.11637
6. Pang J, Cheung G. Graph Laplacian Regularization for Image Denoising: Analysis in the Continuous Domain. IEEE Transactions on Image Processing 26:1770-1785 (2017). https://doi.org/10.1109/TIP.2017.2651400
7. Hu W, Cheung G, Kazui M. Graph-based Dequantization of Block-Compressed Piecewise Smooth Images. IEEE Signal Processing Letters 23:242-246 (2016). https://doi.org/10.1109/LSP.2015.2510379
8. Liu X, Zhai D, Zhao D, Zhai G, Gao W. Progressive Image Denoising Through Hybrid Graph Laplacian Regularization: A Unified Framework. IEEE Transactions on Image Processing 23:1491-1503 (2014). https://doi.org/10.1109/TIP.2014.2303638
9. Fujiwara K, Kano M, Hasebe S. Correlation-based spectral clustering for flexible process monitoring. J Process Control 21:1438-1448 (2011). https://doi.org/10.1016/j.jprocont.2011.06.023
10. Machida Y, Ootakara S, Seki H, Hashimoto Y, Kano M, Miyake Y, Anzai N, Sawai M, Katsuno T, Omata T. Vinyl acetate monomer (VAM) plant model: A new benchmark problem for control and operation study. IFAC-PapersOnLine 49:533-538 (2016). https://doi.org/10.1016/j.ifacol.2016.07.397
11. Omega Simulation Co., Lid., https://www.omegasim.co.jp, (accessed 2024/07/31)