GCN-based Soft Sensor Utilizing Process Flow Diagram

Hiroki Horiuchi, Yoshiyuki Yamashita\*

Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei, Tokyo 184-8588, Japan

yama\_pse@cc.tuat.ac.jp

Abstract

Ensuring the secure and steady operation of chemical plants, online and real-time measurement of process variables is inevitable. However, certain important variables remain unmeasured due to prohibitive costs or inherent complexities. Soft sensors offer a pivotal solution to estimate unmeasured variables from the available data. Prevailing methods employ either physical models or data-driven models, such as PLS, LSTM, and CNN, to construct the estimation model. This study proposes the integration of piping and instrumentation diagram (P&ID) knowledge into data-driven soft sensor modeling, deploying a graph convolutional network (GCN) to capture spatial relationships. The proposed methodology has been validated through its application to concentration estimation in the Tennessee Eastman Benchmark Simulation, demonstrating its effectiveness and robustness to the accurate estimation.

**Keywords**: Graph Convolutional Networks, P&ID, soft sensor

* 1. Introduction

For safe and stable operation of chemical plants, it is essential to monitor plant conditions and properly control. However, there are many process variables where frequent measurement is difficult or costly. It is difficult to control such process variables in real time, resulting in control lag. Therefore, soft sensors are used as a method to estimate the values of process variables that are difficult to measure in real time. In recent years, data-driven methods such as Deep Neural Network (DNN) and Convolutional Neural Network (CNN) have been studied and their prediction accuracy has been improved. However, these methods are susceptible to the influence of training data and are weak in the extrapolation domain, making it difficult to cope with faults and changes in operation (D. Wu et al., 2021). When used for quality control and operation monitoring, soft sensors must have the versatility to make stable predictions under a variety of process conditions.

In this study, we develop a machine learning method for soft sensors that utilizes process knowledge. Piping & Instrumentation Diagrams (P&IDs), which are the blueprints of chemical plants, contain a variety of information about the plant. By utilizing this information for soft sensors, we aim to develop a method that has high prediction accuracy and is not easily affected by operational changes or faults in the plant.

* 1. Method
		1. Graph Convolutional Network (GCN)

The concept of CNN, originally proposed by Y. LeCun et al. (1998), has been extensively used for Euclidean data such as images and texts. However, there kinds of direct embedding methods lack generalization capabilities for dynamic graphs (H. Cai et al., 2018). GCN aggregate information from graph structure and embedding (M. Xu et al., 2020). They introduce spectral convolutions, where a signal vector is multiplied with a filter parameterized in the Fourier domain. However, this approach has high computational complexity due to the multiplication with the Fourier basis. To address this problem, a graph convolutional network was proposed by T. N. Kipf and M. Welling (2017).

A multilayer GCN model is generalized as follows:

|  |  |
| --- | --- |
| $$H^{\left(l+1\right)}= σ(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{\left(l\right)}W^{\left(l\right)})$$ | (1) |

where $H^{\left(l+1\right)}$ and $H^{\left(l\right)}$ are the output and input of layer $l,θ^{\left(l\right)}$ is the parameter of layer $l$ and $σ(∙)$ is the activation function used for nonlinear modelling. Generally, a two-layer GCN model is generalized as follows:

|  |  |
| --- | --- |
| $$f\left(X,A\right)= σ(\hat{A} ReLU\left(\hat{A} XW^{\left(0\right)}\right)W^{\left(1\right)})$$ | (2) |

where $W^{\left(0\right)} \in R^{C×H}$ is the weight matrix from the input layer to the hidden unit layer, $C$ is the length of time, and $H$ is the number of hidden units. $W^{\left(1\right)} \in R^{H×F}$ is the weight matrix from the hidden layer to the output layer, $f\left(X,A\right)\in R^{N×F}$ denotes the output with a forecasting length of $F$, and $ReLU(∙)$ is a common nonlinear activation function.

* + 1. Directed graph
			1. Definition of graph

A graph is represented as $G=\left(V,E\right)$ where $V$ is the set of nodes and $E$ is the set of edges. Let $v\_{i} \in V$ to denote a node and $ e\_{ij}=\left(v\_{i}, v\_{j}\right)\in E$ to denote an edge pointing from $v\_{j}$ to $v\_{i}$. The neighborhood of a node $v$ is defined as $N\left(v\right)=\left\{u\in V \right| \left(v,u\right)\in E\}$. The adjacency matrix $A$ is a $n×n$ matrix with $A\_{ij}=1$ if $e\_{ij} \in E$ and $A\_{ij}=0$ if $e\_{ij}\notin E$. A graph may have node attributes $X$, where $X \in R^{n×d}$ is a node feature matrix with $x\_{v} \in R^{d}$ representing the feature vector of a node $v$. Meanwhile, a graph may have edge attributes $X^{e}$, where $X^{e}= R^{m×c}$ is an edge feature matrix with $x\_{(v,u)}\in R^{c}$ representing the feature vector of an edge $(v,u)$.

A directed graph is a graph with all edges directed from one node to another. An undirected graph is considered as a special case of directed graphs where there is a pair of edges with inverse directions if two nodes are connected. A graph is undirected if and only if the adjacency matrix is symmetric.

* + - 1. k-NN graph

The *k*-NN graph is a representation of the relationships between data points in each dataset. It operates on the principle that similar data points in a feature space tend to exhibit similar characteristics. This graph is constructed by connecting each data point to its *k*-nearest neighbors, forming a network of interconnected nodes. To build the *k*-NN graph, one must first define a distance metric to quantify the similarity between data points. Common distance metrics include Euclidean distance, Manhattan distance, or cosine similarity. For each data point, the algorithm identifies its *k*-nearest neighbors based on the chosen metric and establishes edges between them in the graph (Y. Zhang, Jianbo Yu, 2022).

* + - 1. Proposed method

In chemical processes, unit operations and streams are physically connected with pipes. P&ID contains a large amount of knowledge about the relationships between different variables. To utilize the information of P&ID, it should be transformed into a graph. The graph works as loose constraints for the relationships between variables. Wu et al. showed a method to create a graph and to process the node data (D. Wu et al., 2021). Detailed steps to construct graphs from chemical processes are listed below.

Step 1: Create a stream node and a unit operation node based on the P&ID. Set edges between the stream node and the unit operation node from upstream to downstream.

Step 2: Correspondence between the node created in step 1 and the measurement node.

Step 3: Set edges from the stream and unit operation node created in step 1 toward the measurement node based on the P&ID.

Step 4: Add a manipulated node to the graph in step3 based on the P&ID. Set edges from the manipulated node to the control node and from the control node to the measurement node.

Step 5: Remove unnecessary nodes and directly connect existing nodes.

Step 6: Create self-loop edges.

In this study, lagged explanatory variables were also used to predict the objective variable. The total number of nodes never changed before or after adding the lag, because each lagged explanatory variable was stacked in a row of one measurement node. After P&ID was transformed into a graph, graph convolution is introduced by GCN, which is utilized to extract information from a graph.

* 1. Case study
		1. Setup and dataset

The proposed method is evaluated on Tennessee Eastman Process (TEP) which is one of most popular benchmarks in chemical process. This process contains five main unit operations: a reactor, a condenser, a product stripper, a vapor-liquid separator, and a recycle compressor. This process flow is shown in Figure 1. The process produces two products from four reactants, including an inert and a byproduct. The process has 41 measurements and 12 manipulated variables in (Downs and Vogel, 1993).

In our study, we used 10,000 data measured every five minutes (Rieth et al., 2017). However, concentrations were measured every 10 or 15 minutes. Therefore, the most recent data was used for missing data. As a pre-processing step, data standardization was performed. In addition, data were forecasted using data up to three previous time points. The objective variable was set to the concentration of product component H.

Furthermore, 20 different faults were generated in the TEP simulator to check the prediction accuracy in the extrapolation region. We obtained 960 pieces of data for each fault and used 8,000 pieces of normal data for training and all data after the occurrence of the fault as test data for validation.



Figure 1. Process flow of the TEP

|  |
| --- |
| Table 1. RMSE and MAE for each method |
|  | PLS | MLP | 2D-CNN | *k*-NN+GCN | Proposed method |
| RMSE | 0.3578 | 0.4184 | 0.3492 | 0.3286 | **0.3198** |
| MAE | 0.2527 | 0.3143 | 0.2512 | 0.2285 | **0.2194** |

* + 1. Result and Discussion
			1. Prediction for normal data

The proposed method was applied to predict the concentration of product component H. The prediction model was based on the training dataset for normal run (IDV00). The prediction accuracy was evaluated on the test dataset for normal run. The GCN model consists of an input layer, two hidden layers, and an output layer. The input layer has 136 units, while the hidden layers have 64 and 16 units, respectively. The output layer has 1 unit. ReLU activation functions were employed in the hidden layer, and MSE was used for the loss function. In addition, Adam was used to optimize the model. For comparison, PLS, MLP, 2D-CNN, *k*-NN+GCN were also used to predict the concentration H. Here, the MLP model consists of an input layer, two hidden layers, and an output layer. The hidden layers have 64 and 16 units, respectively. ReLU activation functions were employed in the hidden layer, and MSE was used for the loss function. The *k*-NN +GCN is a GCN model with *k*-NN graph. This method is proposed by Y. Zhang and J. Yu (2022). The architecture of each of these four methods is optimized by grid search. First, we tested whether incorporating P&ID into the model would improve the accuracy of the model. Prediction errors are summarized in Table 1 by MSE, RMSE and MAE.

Table 1 shows that the GCN model using the adjacency matrix created based on the P&ID has the highest accuracy. This shows the effectiveness of utilizing domain knowledge in a graphical form. In addition, it can be said that incorporating P&ID into GCN as graphical data can leverage domain knowledge and improve accuracy. An example of component concentration prediction is shown in Fig. 2.



Figure 2. Prediction results for normal data

|  |
| --- |
| Table 2. RMSE for each fault type |
| Fault number | 3 | 9 | 11 | 15 | 20 |
| PLS | 0.3027 | 0.3023 | 0.3039 | 0.3024 | 0.3142 |
| MLP | 0.3284 | 0.3285 | 0.3305 | 0.3281 | 0.3418 |
| 2D-CNN | 0.3025 | 0.3021 | 0.3018 | 0.3011 | 0.3291 |
| *k*-NN+GCN | 0.3233 | 0.3240 | 0.3251 | 0.3227 | 0.3423 |
| Proposed method | **0.2932** | **0.2930** | **0.2949** | **0.2951** | **0.3031** |

* + - 1. Prediction for fault data

The proposed method was applied to the prediction of the concentration of product component H. The prediction model was based on the training dataset for normal run. The prediction accuracy was evaluated on the test dataset for faults run. PLS, MLP, 2D-CNN, *k*-NN+GCN were also used to predict the concentration of component H. The results of applying the model trained on normal data to all fault data showed that the proposed method improved RMSE for most faults. Some excerpts of the prediction errors are summarised in Table 2 by RMSE.

Table 2 confirms that the proposed method has the best results for five faults data. Since fault data was not used in the training data, the prediction of fault data is an example of prediction in the extrapolation domain. The high prediction accuracy of the proposed method suggests that the prediction in the extrapolation region was enhanced by utilizing the P&ID information. It is thought that the use of P&ID enabled more appropriate setting of learning parameters, resulting in higher prediction accuracy.

* 1. Conclusions

In summary, our study aimed to enhance the accuracy of soft sensors in chemical plants by integrating P&ID into the GCN method. The results demonstrated the superior performance of the P&ID-based GCN soft sensor compared to a *k*-NN graph based on GCN and other traditional methods such as PLS, MLP and CNN.

The use of P&ID as domain knowledge significantly improved prediction accuracy, showcasing the potential of graphical representation in leveraging domain-specific insights. The proposed method not only surpassed alternative soft sensor, but also showed a new way to introduce domain knowledge when developing soft sensors.

This research highlights the promising synergy between graph-based neural networks and domain knowledge. In addition, it was confirmed that the proposed method suppresses the decrease in prediction accuracy in the extrapolation domain, which has been a drawback of conventional machine learning methods.

Soft sensors are expected to be used for quality control and operation monitoring. When used for quality control purposes, they are required to have high prediction accuracy and versatility to make stable predictions under various process conditions. The proposed method was found to improve the results of these two elements.

References

H. Cai, W. Vincent, W. Zheng, K. C. Chang, 2018, A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications, IEEE Transactions on Knowledge & Data Engineering, vol. 30, no. 9, pp. 1616-1637

J. J. Downs, E. F. Vogel, 1993, A plant-wide industorial process control problem, Comput. Chem. Eng., vol. 17, pp. 245-255

L. Feng, C. Zhao, Y. Li, M. Zhou, H. Qiao, C. Fu, 2021, Multichannel Diffusion Graph Convolutional Network for the Prediction of Endpoint Composition in the Converter Steelmaking Process, IEEE Trans. Instrument. and Meas., vol. 70, 3000413

M. Jia, J. Hu, Y. Liu, Z. Gao, Y. Yao, 2023, Topology-Guided Graph Learning for Process Fault Diagnosis, Industrial & Engineering Chemistry Research, vol. 62, pp. 3238-3248

T. N. Kipf, M. Welling, 2017, Semi-Supervised Classification with Graph Convolutional networks, International Conference on Learning Representation (ICLR), Toulon, France

G. D. Molina, D. A. R. Zumoffen, M. S. Basualdo, 2011, Plant-wide control strategy applied to the Tennessee Eastman process at two operating points, Comput. Chem. Eng., vol. 35,
pp. 2081-2097

C. Reinartz, M. Kulahci, O Ravn, 2021, An extended Tennessee Eastman simulation dataset for fault-detection and decision support systems, Comput. Chem. Eng., vol. 149, 107281

Rieth, A. Cory, Amsel, D. Ben, Tran, Randy, Cook, B. Maia, 2017, Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation, Harvard Dataverse, V1

F. A. A. Souza et al., 2016, Review of soft sensor methods for regression applications, Chemometrics and Intelligent Laboratory Systems, vol. 152, pp. 69–79

D. Wu, J. Zhao, 2021, Process topology convolutional network model for chemical process fault diagnosis, Process Safety and Environmental Protection, vol. 150, pp. 93-109

T. Wu, I. Carreno, A. Scaglione, D. Arnold, 2023, Spartio -Temporal Graph Convolutional Neural Networks for Physics-Aware Grid Learning Algorithms, IEEE Trans. Smart Grid,
vol. 14, pp. 4086-4099

M. Xu, 2020, Understanding graph embedding methods and their applications, SIAM, vol. 63,
no. 4, pp. 825-853

Y. Zhang, Jianbo Yu, 2022, Pruning graph convolutional network-based future learning for fault diagnosis of industrial processes, Journal of Process Control, vol. 113, pp. 101-113