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

# Enhancing Batch Chemical Manufacturing via Development of Deep Learning based Predictive Monitoring with Transfer Learning

**Hong Yee Hung[a], Zhao Jinsong[a,*]**

[a] Tsinghua University, Department of Chemical engineering, Beijing, China
* Corresponding Author: jinsongzhao@mail.tsinghua.edu.cn.

## ABSTRACT

Batch chemical processes face significant challenges due to frequent operational shifts and varying conditions, requiring models to be retrained for each new scenario. This high retraining demand limits the scalability of traditional process monitoring systems, making them unsuitable for dynamic batch operations. To address this, we propose a transfer learning-based framework that enhances adaptability by reusing learned features across different batch conditions, reducing the need for extensive retraining. Proposed method integrates Temporal Convolutional Networks (TCNs) for capturing temporal dependencies in batch data and predicting Quality-Indicative Variables (QIVs) to identify deviations. The core innovation lies in transfer learning, enabling the model to adapt to new process variations with minimal updates. This approach ensures robust, accurate monitoring even under evolving conditions. This framework is validated using the IndPenSim penicillin fermentation dataset, which simulates real-world batch process variability. Results show that the transfer learning-enhanced model effectively predicts QIVs and detects deviations across varying control strategies, demonstrating improved adaptability and reduced retraining requirements. This study highlights the potential of transfer learning to revolutionize batch process monitoring by addressing core challenges in dynamic chemical operations.

**Keywords**: Batch Process, Fermentation, Artificial Intelligence, Machine Learning, Process Monitoring, Fault Detection

## 1 INTRODUCTION

The production of high-value-added products like pharmaceuticals, biochemicals, and specialty chemicals heavily relies on batch processes due to their operational flexibility, enabling seamless transitions between various formulations and operating modes within a single facility. However, this flexibility introduces complexities such as intricate process phases and frequent mode switching, which complicate real-time monitoring and control, particularly for quality indicator variables (QIVs) [1]. Since QIVs are often challenging to measure continuously, soft sensors have become a widely adopted solution in industrial settings.

While first-principles models theoretically offer detailed insights into the underlying physicochemical mechanisms, they frequently encounter challenges in modeling the complex nonlinearities and dynamics typical of industrial processes [2]. As an alternative, data-driven methods leverage extensive historical datasets to extract latent features for QIV estimation [3]. Traditional feature extraction approaches include linear methods (e.g., PCA, PLS, ICA) [4,5] and kernel-based variants (e.g., kernel PCA, kernel PLS, SVR) [6], which have shown some efficacy in handling nonlinearity. However, these conventional techniques are often constrained by moderate computational overhead and reduced generalizability when faced with large-scale data and constantly changing operational conditions [7]. This issue becomes especially critical in batch processes, where multi-stage operations, a three-dimensional data structure (time, variables, batches), and multiple operating modes necessitate frequent model reconfiguration, thus restricting their real-time applicability.

Recent advances in deep learning offer promising avenues for addressing these challenges. Feedforward neural networks and recurrent neural networks (RNNs) exhibit relative strengths in modeling nonlinear and time-varying processes [8], while LSTM networks excel in capturing long-term dependencies [9]. Despite these advantages, LSTM-based methods may still struggle to capture short-term dynamic features between different batch phases and are susceptible to performance degradation under frequent mode transitions and significant batch-to-batch variability [10]. Consequently, transfer learning has gained traction for batch applications, as it facilitates the adaptation of previously acquired knowledge to new modes or new batches, substantially reducing the need for repeated training .

In this context, the present study proposes a soft sensing framework that integrates transfer learning with Temporal Convolutional Networks (TCNs) to achieve both feature extraction and QIV prediction. By introducing a transfer learning mechanism, the model effectively adapts to diverse operating conditions while minimizing retraining efforts. In addition, a variable selection strategy is employed to discard features with limited relevance to QIVs, thereby enhancing predictive accuracy and computational efficiency. The proposed framework addresses the following key challenges:

1. Multi-mode switching: The model can adapt to different operating modes without extensive retraining, addressing batch-to-batch variability.
2. Complex temporal dynamics: TCNs are employed to capture both short-term and long-term dependencies within batch processes.
3. Variable relevance: A variable selection method is introduced to enhance the model's accuracy by focusing on variables that have a strong influence on QIV prediction.

The proposed framework is validated using the IndPenSim penicillin fermentation dataset, which encompasses a variety of control strategies and considerable batch-to-batch variability, thus closely reflecting industrial batch production scenarios. Experimental results indicate that the proposed approach not only captures both short- and long-term temporal dependencies but also sustains robust prediction performance amid frequent mode switching. Consequently, it presents a scalable and readily adaptable solution for real-time QIV prediction and process monitoring in industrial batch operations.

# 2 PRELMINARIES

## 2.1 Benjamini-Hochberg Test

The Benjamini-Hochberg (BH) test controls the false discovery rate (FDR) in multiple hypothesis testing [11]. For soft sensor variable selection, it identifies variables significantly correlated with Quality Indicator Variables (QIVs) by ranking p-values and comparing them to an FDR-adjusted threshold. Variables below the threshold are retained, reducing dimensionality while preserving key features for accurate QIV prediction. Figure 1 illustrates the flow diagram of BH testing.

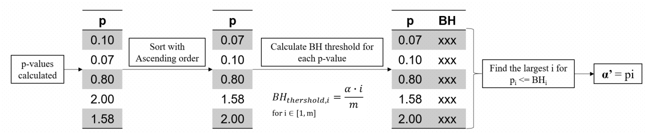


**Figure 1.** Benjamini-Hochberg (BH) program for FDR Control

## 2.2 Temporal Convolutional Network

Temporal Convolutional Networks (TCNs) are deep learning models for sequence modeling that capture both short- and long-term temporal dependencies. Unlike RNNs, which process sequences step by step, TCNs use causal convolutions to ensure predictions depend only on past data. They leverage dilated convolutions to expand their receptive field without requiring deep architectures, and residual connections to stabilize training and mitigate vanishing gradients. TCNs also produce sequence-to-sequence outputs, making them effective for time-series applications in batch processes. Figure 2 shows the TCN framework.

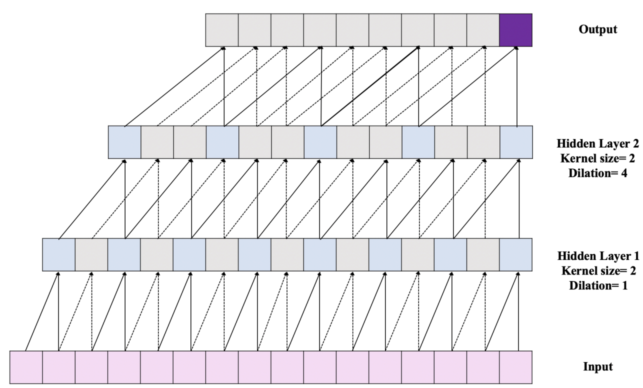


**Figure 2.** Structure of Temporal Convolutional Network

## 2.3 Transfer Learning

Transfer learning reuses knowledge from one task to improve performance on a related but different task [12]. It is particularly useful in batch processes, which often involve variability and multi-mode switching, making it impractical to gather sufficient labeled data for every new condition.

Key components include:

1. Pretrained Feature Extraction: Train a base model on one batch process or condition to capture generalizable patterns.
2. Fine-Tuning: Retrain this model on a smaller dataset from the new condition, reducing data and time

requirements while maintaining accuracy.

Domain Adaptation: Employ methods (e.g., domain-invariant feature extraction) to handle discrepancies between source and target domains. Figure 3 illustrates the transfer learning framework.

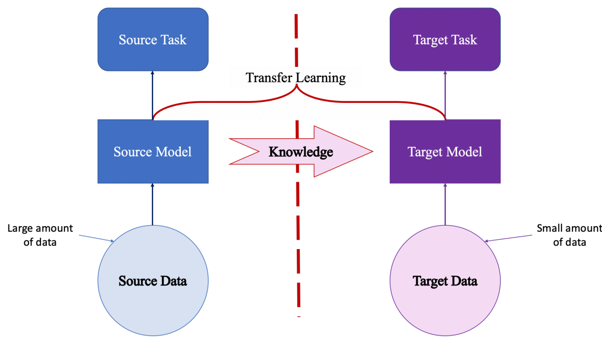


**Figure 3.** Framework of Transfer Learning

# 3 METHODOLOGY

## 3.1 Procedure of Quality Monitoring Framework

The proposed quality monitoring framework follows a two-stage approach, comprising an offline training phase and an online monitoring phase: The proposed quality monitoring framework follows a two-stage approach, comprising an offline training phase and an online monitoring phase:

1. Offline Training Phase

Historical data from past batches under normal conditions are collected, including process variables and Quality Indicator Variables (QIVs). Relevant variables are selected using a mutual information-based method combined with a Benjamini-Hochberg permutation test. A Temporal Convolutional Network (TCN) is trained on the filtered data to capture temporal dynamics, and transfer learning is incorporated for adaptability across different operating modes. Statistical thresholds for fault detection are set using the 3-sigma rule based on model prediction errors.

2. Online Monitoring Phase

Real-time process data are continuously fed into the trained TCN model, which estimates QIVs for comparison with reference profiles or setpoints. Significant deviations outside the 3-sigma thresholds trigger alarms for potential faults. When major process changes occur, transfer learning updates the model with minimal retraining to maintain monitoring accuracy.

A schematic flow chart illustrated the interplay between the offline and online stages in figure 4.

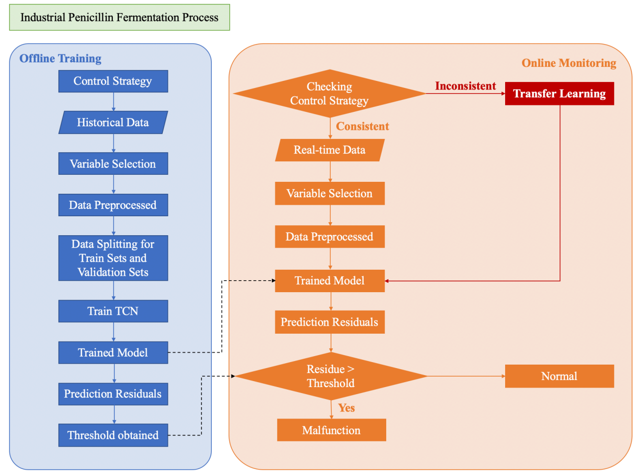


**Figure 4.** The Procedure of Monitoring Framework

## 3.2 Variable Selection by Permutation Test

### 3.2.1 Mutual Information

As an initial step, the mutual information (MI) between each process variable and the QIV is computed. MI measures the shared information between two random variables and captures both linear and nonlinear dependencies. The MI between a candidate variable X and the QIV Y is defined as:

$$I(X;Y) = \iint p(x,y) log \frac{p(x,y)}{p(x)p(y)}\, dx\, dy \tag{1}$$

where $p\ (x,\ y)$ is the joint probability density of $X$ and $Y$, and $p(x)$, $p(y)$ are their marginal distributions. Variables with high MI values indicate stronger relevance to QIVs, while variables with low MI suggest weaker connections.

### 1.2.2 BH Program-based Permutation Test

Although MI identifies candidate variables, it does not provide a definitive threshold for deciding which variables to keep or discard. To address this, a BH-based permutation test is employed.

The variable selection process begins by removing clearly irrelevant variables (e.g., batch number, condition flag) and computing the mutual information (MI) between each candidate variable $X_k$ and the QIV $Y$. Next, a permutation test is performed by randomly shuffling each $X_k$ multiple times to generate a null distribution of MI values, and a p-value is calculated between $X_k$ and $Y$ based on how often the observed MI exceeds the permuted values. The p-values are then sorted and the Benjamini-Hochberg (BH) procedure is applied to control the false discovery rate (FDR), retaining variables with p-values below the threshold. Finally, the significant variables from the BH test formed the final subset for model development.

Figure 5 illustrates the whole procedure of variable selection proposed by this work.

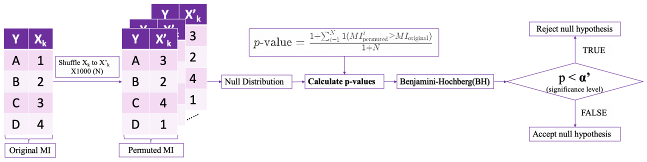


**Figure 5.** Variable Selection through MI Value and BH program -based Permutation Test

## 3.3 Proposed TCN with Transfer Learning Strategy

### 3.3.1 Batch Process Data Preprocessing

Batch processes inherently produce three-dimensional data: batch index, time, and process variables. Traditional data unfolding methods reshape these data into two dimensions, but they may disrupt the inherent structure of batch trajectories, especially when batches vary in length or exhibit phase transitions.

Instead of unfolding all batches onto a single extended axis, this work employed a sliding window approach within each batch. This strategy respects the boundary of each batch by ensuring that a window never overlaps data from different batches. Consequently, it better preserves local temporal correlations and phase-wise characteristics for. Figure 6 shows the strategy of sliding window to cope with the uneven length situation.

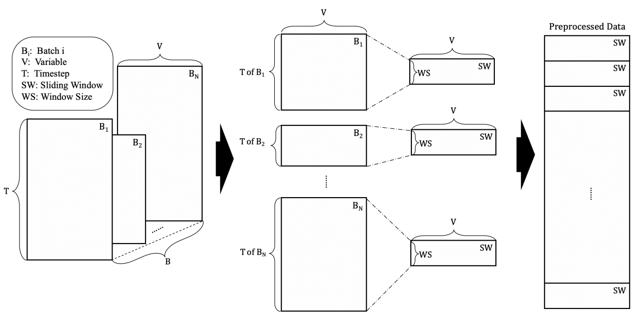


**Figure 6.** Proposed Data Preprocessing Strategy

### 3.3.2 Model Structure and Transfer Learning

The proposed framework involves the following steps after data preprocessing, which ensures synchronization of batch data and standardization of variables. First, a TCN is trained using Normal Operating Condition (NOC) data to capture temporal dependencies and predict Quality Indicator Variables (QIVs).

In this study, Normal Operating Data (NOC data) are used to train the TCN model by minimizing the mean squared error (MSE) between estimated and actual QIV values:

$$min\ MSE = \frac{1}{N}\sum_{i=1}^{N}(\widehat{Y}_i - Y_i)^2 \tag{2}$$

where $\widehat{Y}_i$ denotes the predicted QIV and $Y_i$ is the true QIV.

To address multi-mode switching and between-batch variability, transfer learning is incorporated into the TCN framework. Once a base TCN model is trained on one operating regime, it can be adapted to new or slightly changed conditions with fewer additional data. The pre-trained TCN is fine-tuned using limited data from a new regime to adapt to changes in operating conditions, such as control strategy variations or raw material differences to validate the transfer learning strategy. Real-time QIV predictions are compared against thresholds derived during training to detect deviations indicative of faults.

### 1.3.4 Process Quality Monitoring

After the TCN model is trained or adapted, it serves as the core of the quality monitoring system:

1. Real-Time QIV Prediction: During operation, the TCN predicts QIV values from incoming sensor data.
2. Threshold Determination: 3-sigma rules (Montgomery, 2012) are applied to the residuals (prediction errors) obtained during the offline training stage. Let the residuals be:

$$e_t = \left|\widehat{Y}_t - Y_t\right|^2 \tag{3}$$

Then the control limits can be set as:

$$Threshold = \mu_e \pm 3\sigma_e \tag{4}$$

where $\mu_e$ and $\sigma_e$ are the mean and standard deviation of residuals under normal conditions.

3. Fault Detection: If the residual at time $t$ exceeds these control limits, an alarm is raised to signal a potential fault or deviation from the normal operating condition.
4. Validation with Fault Data: To evaluate fault-detection performance, representative fault scenarios or abnormal batches are introduced, and the detection rate is observed based on the discrepancy of predicted QIVs from normal ranges.

This monitoring approach allows operators to detect process anomalies promptly, maintain product quality, and facilitate proactive adjustments in dynamic batch operations.

# 4 EXPERIMENT

## 4.1 Description of IndPenSim Datasets

The experiments are conducted using the IndPenSim dataset, an industrial-scale penicillin fermentation simulation developed by Goldrick et al. (2019) [13]. Unlike the traditional PenSim benchmark [14], IndPenSim models a 100,000 L bioreactor, bringing it closer to real-world industrial conditions and validated by historical industrial data. Detailed process descriptions and operational specifics can be found at the IndPenSim website and in the original publications [13]. Figure 7 illustrates the summary of IndPenSim.

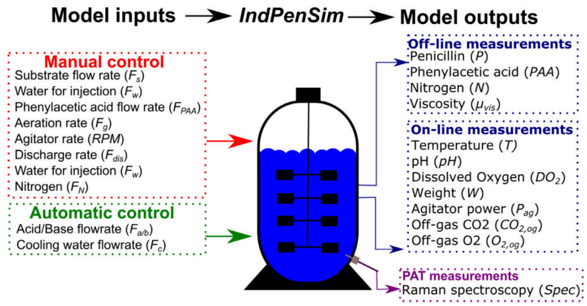


**Figure 7.** Summary of Inputs and Outputs of IndPenSim

This study uses a dataset of 100 simulated batches, with batch lengths ranging from 167 to 290 hours, sampled every 0.2 hours. The first 90 batches represent normal operating conditions and are divided into three groups of 30, each corresponding to a different control strategy. The remaining 10 batches simulate fault conditions to evaluate the fault detection capability. For model training, 20 batches from each control strategy (total of 60 batches) are used, with 30 additional normal batches reserved for evaluation. The 10 fault batches are excluded from training, as the fault conditions and control strategies are unknown. Fault batch data for transfer learning evaluation is generated using a simulator for each control strategy. Detailed batch description and corresponding control strategies are provided in Table 1 and Table 2.

**Table 1:** Batch Description from IndPenSim Datasets.

| Batch No. | Control Strategy | Batch Type |
|---|---|---|
| 1-30 | Recipe Driven Approach | Normal Batch |
| 31-60 | Operators | Normal Batch |
| 61-90 | Advanced Process Control (APC) & Raman Spectroscopy | Normal Batch |
| 91-100 | Unknown | Fault Batch |

**Table 2:** Faulty Batch generated from IndPenSim.

| Fault Batch No. | Control Strategy | Fault Type |
|---|---|---|
| 1 | Recipe Driven Approach | Quality-related |
| 2 | Operators | Quality-related |

## 4.2 Results of Variable Selection

Initially, all 32 process variables are screened based on domain knowledge, removing irrelevant ones not linked to penicillin concentration (QIV) to simplify further analysis. The BH-program-based permutation test is then applied for refined variable selection. Mutual information (MI) values between the retained variables and QIV are computed, and the permutation test evaluates their statistical significance, filtering out weak or insignificant correlations. The final set of highly relevant variables, summarized in Table 3, ensures efficient and effective model development. Figures 8 shows the MI values for the selected variables.

**Table 3:** Screened Variables by Variable Selection.

| Variable No. | Variable description | Unit |
|---|---|---|
| 1 | Aeration rate | L/h |
| 2 | Sugar feed rate | L/h |
| 3 | Acid flow rate | L/h |
| 4 | Base flow rate | L/h |
| 5 | Heating/cooling water flow rate | L/h |
| 6 | Heating water flow rate | L/h |
| 7 | Water for injection/dilution | L/h |
| 8 | Air head pressure | bar |
| 9 | Dumped broth flow | L/h |
| 10 | Substrate concentration | g/L |
| 11 | Dissolved oxygen concentration | mg/L |
| 12 | Vessel Volume | L |
| 13 | Vessel Weight | Kg |
| 14 | pH | / |
| 15 | Temperature | K |
| 16 | Generated heat | kJ |
| 17 | carbon dioxide percent in off-gas | % |
| 18 | PAA flow | L/h |
| 19 | Oil flow | L/hr |
| 20 | Oxygen Uptake Rate | g/min |
| 21 | Oxygen in percent in off-gas | % |
| 22 | Carbon evolution rate | g/h |
| 23 | Penicillin concentration | g/L |

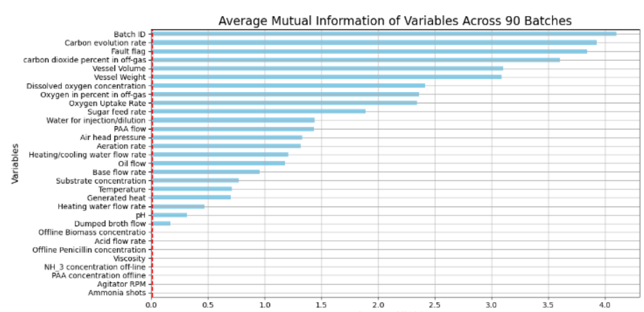


**Figure 8.** Average MI Values of Variables Across 90 batches

## 4.3 Results of Process Monitoring with Transfer Learning

The performance of the proposed TCN-based soft sensor model is evaluated through real-time quality prediction and fault detection. By using normal operating data, the TCN model is trained to predict QIV values. The accuracy of these predictions is illustrated in Figure 9, which compares predicted QIV trajectories against actual values over time. High fidelity between predictions and true values underlines the model's capability in capturing complex process dynamics.

The model's ability to detect deviations is assessed using fault batches. Figure 10 also shows the residual errors (difference between predicted and measured QIV) alongside the statistically determined control limits based on the 3-sigma rule. Instances where residuals exceed these thresholds indicate faults, demonstrating effective anomaly detection in real-time scenarios.

To validate the adaptability of the proposed framework, the model trained under Control Strategy 1 is adapted to Control Strategy 2 using transfer learning techniques. Figure 11 and 12 depicts the monitoring results after applying transfer learning, showing how the model successfully transitions between different operational strategies with minimal additional training. Table 4 illustrates the Fault Detection Rate (FDR) and False Alarm Rate (FAR) of both without transfer learning and with transfer learning strategy.

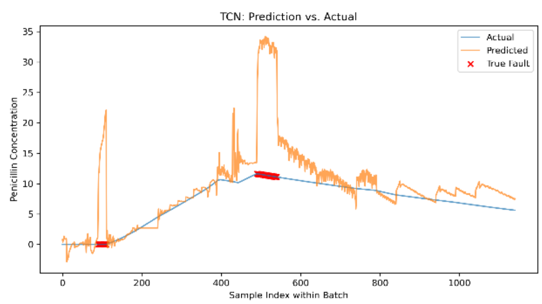


**Figure 9.** Prediction Results of Fault Batch 1

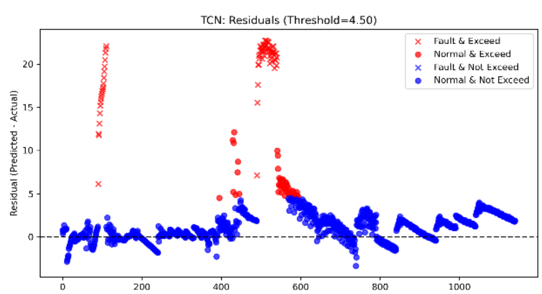


**Figure 10.** Residuals of Fault Batch 1

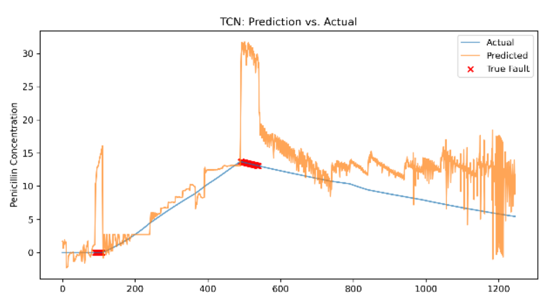


**Figure 11.** Prediction Results of Transfer Learning from Control Strategy 1 to Control Strategy 2 (Fault Batch 2)

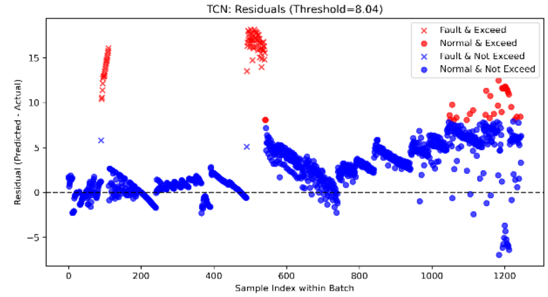


**Figure 12.** Residuals Results of Transfer Learning from Control Strategy 1 to Control Strategy 2 (Fault Batch 2)

**Table 4:** Statistical Results.

| Method. | FDR (%) | FAR (%) |
|---|---|---|
| General | 100.00 | 4.58 |
| Transfer Learning | 97.22 | 2.73 |

## 5 CONCLUSION

This study presents a real-time quality monitoring framework for industrial batch processes, integrating robust variable selection, TCN-based modeling, and transfer learning. It effectively tackles challenges including nonlinear dynamics, multi-phase operations, and batch-to-batch variability. The variable selection method uses mutual information combined with a Benjamini-Hochberg-based permutation test, while TCNs capture complex temporal dependencies and transfer learning enables swift adaptation to different control strategies. A 3-sigma rule–based threshold is employed for real-time fault detection, and experiments on the IndPenSim dataset confirm the framework's enhanced prediction, reliable monitoring, and scalability.

Looking ahead, we plan to incorporate additional sensor types such as images and vibration signals to improve adaptability in complex conditions, explore more sophisticated domain adaptation methods to handle broader operational scenarios, and apply the framework to diverse industrial data to validate its robustness and generality.

## ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China under Grant 2024YFC3013600.

## REFERENCES

1. Kadlec P, Gabrys B, Strandt S. Data-driven soft sensors in the process industry[J]. Computers & chemical engineering, 2009, 33(4): 795-814.
2. Shang H L. A survey of functional principal

component analysis[J]. AStA Advances in Statistical Analysis, 2014, 98: 121-142.
3. Sun Q, Ge Z. A survey on deep learning for data-driven soft sensors[J]. IEEE Transactions on Industrial Informatics, 2021, 17(9): 5853-5866.
4. Keithley R B, Wightman R M, Heien M L. Multivariate concentration determination using principal component regression with residual analysis[J]. TrAC Trends in Analytical Chemistry, 2009, 28(9): 1127-1136.
5. Kaneko H, Arakawa M, Funatsu K. Development of a new soft sensor method using independent component analysis and partial least squares[J]. AIChE Journal, 2009, 55(1): 87-98.
6. Drucker H, Burges C J, Kaufman L, et al. Support vector regression machines[J]. Advances in neural information processing systems, 1996, 9.
7. Liu Y, Gao Z, Li P, et al. Just-in-time kernel learning with adaptive parameter selection for soft sensor modeling of batch processes[J]. Industrial & Engineering Chemistry Research, 2012, 51(11): 4313-4327.
8. Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural networks, 2015, 61: 85-117.
9. Wang K, Gopaluni R B, Chen J, et al. Deep learning of complex batch process data and its application on quality prediction[J]. IEEE Transactions on Industrial Informatics, 2018, 16(12): 7233-7242.
10. Wang Y, Zhang J, Zhu H, et al. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 9154-9162.
11. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing[J]. Journal of the Royal statistical society: series B (Methodological), 1995, 57(1): 289-300.
12. Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2009, 22(10): 1345-1359.
13. Goldrick S, Ştefan A, Lovett D, et al. The development of an industrial-scale fed-batch fermentation simulation[J]. Journal of biotechnology, 2015, 193: 70-82.
14. Birol G, Ündey C, Cinar A. A modular simulation package for fed-batch fermentation: penicillin production[J]. Computers & chemical engineering, 2002, 26(11): 1553-1565.