# Early fault diagnosis in chemical processes through multistep multivariable prediction

**Djogap F. Chrysler J.** * **Moncef Chioua** *

* *Department of Chemical Engineering, Polytechnique Montréal, Montréal, QC, Canada (e-mail: chrysler-jacobson.djogap-feujo@polymtl.ca).*

**Abstract:** Effective fault detection and diagnosis (FDD) in chemical process systems is critical for maintaining safe and reliable operations. While deep learning methods have improved fault classification performance, they often require long sequences of data after a fault occurs, delaying timely interventions. In this work, we propose an early fault diagnosis method that enables rapid fault diagnosis using multistep multivariable prediction. Our approach employs a transformer-based prediction model to predict future values of key process variables, enriching the current data with these predictions. An LSTM-based model then classifies the enriched data into specific fault categories, leveraging both current and predicted information for improved precision. We evaluate the performance of the proposed approach on the Tennessee Eastman Process benchmark, demonstrating its effectiveness in early fault diagnosis.

*Keywords:* Abnormal situation management, Early fault diagnosis, Chemical process, Transformer, LSTM.

## 1. INTRODUCTION

Abnormal situation management is crucial in the process industry to ensure safety and continuity, as faults can lead to risks, environmental impacts, and high costs (Chiang, 2000b). While distributed control systems have enabled large-scale data collection for data-driven monitoring (Md Nor et al., 2020), increased interactions in industrial systems have made processes more complex and raised the risk of faults (Bi et al., 2022). Although Advanced Process Control (APC) solutions have improved performance during normal operations (Lee et al., 2018), robust abnormal situation management and intelligent monitoring methods remain essential for managing this complexity (Qin and Chiang, 2019).

Proactive management of abnormal situations requires tools that enable timely fault detection. Developing an early fault diagnosis algorithm is essential for industrial efficiency and safety. Rapid and precise fault diagnosis from real-time data helps operators take proactive action, reducing unplanned downtime, minimizing production losses, and enhancing safety.

Traditionally, process control systems use a monitoring loop consisting of fault detection, fault isolation, fault identification, and process correction (Chiang, 2000a). Recently, researchers have combined the first three steps into fault classification, treating fault diagnosis as a multiclass classification problem. Common algorithms include k-nearest neighbors (KNN) (Peterson, 2009), Fisher discriminant analysis (FDA) (Russell et al., 2000), and support vector machines (SVM) (Meyer and Wien, 2001). However, these methods often struggle with the complex, non-linear nature of industrial processes. Deep neural networks, especially recurrent neural networks (RNNs), have become prominent due to their ability to process temporal sequences and learn long-term dependencies (Zhao et al., 2018).

RNN-based methods focus on sequential data, integrating the output of each step as part of the input for the next (Zhang et al., 2019). These techniques are relevant for dynamic chemical processes that constantly vary. Methods like Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and their bidirectional versions (BiLSTM and BiGRU) have been used to capture these dynamic characteristics (Zhao et al., 2018).

In (Zhao et al., 2018), an LSTM-based fault diagnosis method applied to the Tennessee Eastman Process benchmark shows better performance than traditional methods. In (Han et al., 2020), an optimized LSTM network outperforms classical neural networks and multilayer perceptrons in diagnosis precision.

Although these architectures show satisfactory classification precision, they are tested on data spanning hours, allowing faults to fully develop and become easier to classify (Wei et al., 2022). This can be impractical, as classifying a fault hours after its appearance is not helpful for productivity and safety. Accurate fault classification in the initial moments after a fault's appearance is more relevant. However, current algorithms show poor precision during these critical moments (Wei et al., 2022).

Therefore, we propose an early fault diagnosis method that enables timely and accurate identification of faults using real-time process data. This method first applies a multistep prediction model to predict upcoming values for each process variable, enriching the initial data with these predicted values. The enriched data is then classified into one of several fault categories. Bai and Zhao (2023) suggests

an approach involving multi-step predictions for different process variables values, monitoring which one exceeds a set limit to detect a fault. However, this method overlooks the potential of combining variables that, even without reaching a limit, could indicate a fault earlier. In contrast, our classification-based approach simultaneously considers all process variables enriched with their predicted values. Furthermore, our goal goes beyond fault detection to achieve precise fault diagnosis through classification based on process data enriched by predictions.

The contributions of this paper are as follows:

- A workflow to determine the extent to which faults can be diagnosed as early as possible, taking fault dynamics into account to help the operator respond quickly to mitigate the effect of the process fault.
- A combination of two models: a first model for predicting process variables values, and a second model for fault diagnosis based on the classification of process variables values enriched with their predictions.
- Results obtained from the Tennessee Eastman Process benchmark that demonstrate how the proposed approach can allow a similar level of classification precision to be reached earlier, thereby providing operators more time to intervene.

The remainder of the paper is organized as follows: Section 2 presents the proposed prediction and classification-based early fault diagnostic method. Section 3 evaluates the effectiveness of the proposed model on the Tennessee Eastman Process benchmark, comparing performance with and without the use of process variables values prediction. Section 4 concludes the study and discusses future work.

## 2. METHODOLOGY

Our approach for early fault diagnosis combines multistep prediction of process variables values with an LSTM-based classification model. We denote by $T$ the total number of timesteps in the time series, and by $n$ the number of process variables considered. The method (Figure 1) proceeds as follows:

- A transformer-based prediction model learns to predict future values of $n$ process variables over a horizon $H$, using a lookback window of size $w$.
- The predicted variables are concatenated with the current measurements, forming an enriched feature set capturing short-term anticipated dynamics.
- An LSTM classifier then classifies each enriched sequence into one of the possible faults or nominal operation.

We detail below each component of the approach.

### *2.1 Multi-step prediction of time series of process variables values*

Multi-step prediction of process variables values in this work uses a transformer model (Vaswani, 2017), which is well-suited for capturing temporal dependencies through its self-attention mechanism (Bai and Zhao, 2023). Specifically, given a multivariate time series

$$X = \{x_1, x_2, \ldots, x_T\}, \quad x_i \in \mathbb{R}^n,$$

The goal is to predict each variable over a horizon $H$, given a lookback window of size $w$. For each position $i$ from 1 to $T - w - H + 1$, we take

$$X_i = \{x_i, \ldots, x_{i+w-1}\}$$

as input and learn to predict

$$\hat{Y}_i = P(X_i), \quad \hat{Y}_i \in \mathbb{R}^{H \times n},$$

where $P$ denotes the trained transformer model. The corresponding target (ground truth) is

$$Y_i = \{x_{i+w}, \ldots, x_{i+w+H-1}\}.$$

Algorithm 1 describes the training procedure.

---

**Algorithm 1: Multistep Prediction using Transformer Inputs:**

- $X = \{x_1, x_2, \ldots, x_T\}$ (time series data with $n$ variables)
- $w$ (lookback window), $H$ (prediction horizon)

**Outputs:**

- Predicted sequences $\hat{Y}_i$ for $i \in [1, T - w - H + 1]$

```
Initialize transformer model parameters
for each epoch do
    for i = 1 to T − w − H + 1 do
        X_i = {x_i, ..., x_{i+w−1}}
        Y_i = {x_{i+w}, ..., x_{i+w+H−1}}
        Ŷ_i = P(X_i)
        Update parameters by minimizing loss between Ŷ_i and Y_i
    end for
end for
```

---

### *2.2 Fault classification based on time series of process variables values*

We use a Long Short-Term Memory (LSTM) network for fault classification from time series data. LSTMs, a type of recurrent neural network (RNN), are well-suited for learning long-term dependencies due to their architecture with memory cells and gating mechanisms (Zhao et al., 2018).

Our approach uses time series data $X = \{x_1, x_2, \ldots, x_T\}$ and fault labels $F = \{F_0, F_1, \ldots, F_{m-1}\}$, where $F_0$ represents the nominal (fault-free) state and $m - 1$ distinct fault types. We segment the time series into sequences with a lookback window of size $w$. For each position $i$ from 1 to $T - w + 1$, we extract a sequence $X_i = \{x_i, x_{i+1}, \ldots, x_{i+w-1}\}$ as an input sample. Algorithm 2 details the training steps of the LSTM model $C$, which learns to map these input sequences to their fault labels in $F$.

During training, the LSTM processes each sequence, capturing temporal patterns and dependencies. The model's weights are adjusted through backpropagation and gradient descent to minimize classification error, associating each input sequence $X_i$ with the correct fault type $C(X_i) = F_j$.

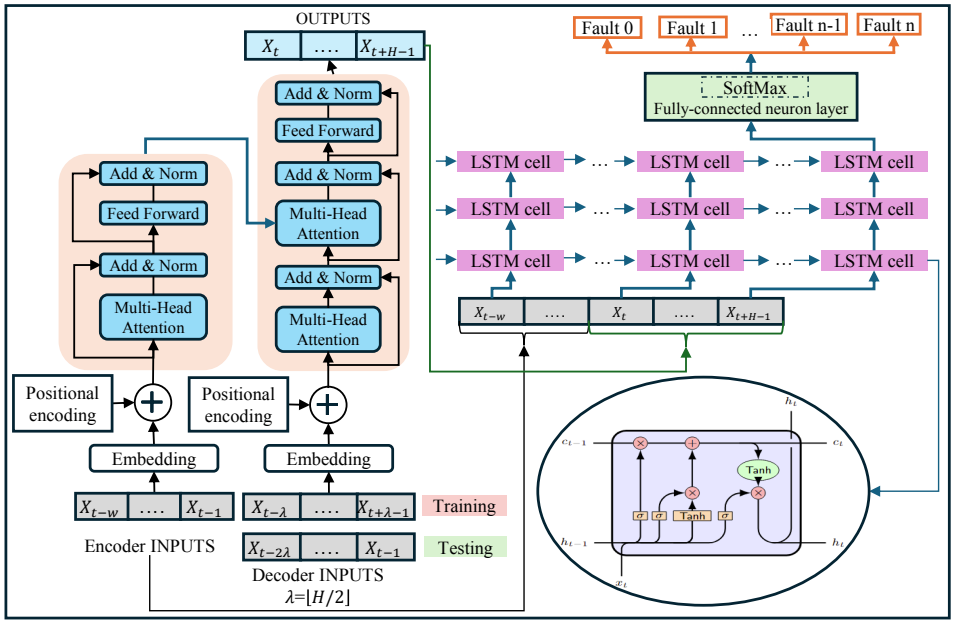



Fig. 1. Steps in the proposed early fault classification based on time series prediction of process variables values. The transformer prediction model architecture shown on the left in Figure 1 is adapted from (Sitapure and Kwon, 2023).

**Algorithm 2: Fault classification based on time series of process variables values Inputs:**

- $X = \{x_1, \ldots, x_T\}$ (time series data)
- $w$ (lookback window)
- Fault labels $F = \{F_0, \ldots, F_{m-1}\}$

**Output:** Fault type for new data

1: Initialize LSTM parameters
2: **for** $i = 1$ to $T - w + 1$ **do**
3: $X_i = \{x_i, \ldots, x_{i+w-1}\}$
4: Label $L_i \in \{F_0, \ldots, F_{m-1}\}$
5: **end for**
6: Train LSTM classifier $C$ on $(X_i, L_i)$ pairs
7: Given new sequence $X_{\text{new}}$, output $C(X_{\text{new}})$

Once trained, the LSTM classifier can predict the fault type for any new data sequence $X_{\text{new}}$ of length $w$ by evaluating $C(X_{\text{new}})$, effectively identifying the presence and type of fault based on learned temporal patterns.

### *2.3 Fault classification based on process variables values prediction*

The fault classification method using process variable prediction is outlined in Algorithm 3. This method enhances fault diagnosis by incorporating predicted time series data to improve classification precision. It employs a transformer prediction model $P$ (architecture shown on the left in Figure 1) and an LSTM classification model $C$ (architecture shown on the right in Figure 1). For a new time series sequence $X_{\text{new}}$, the transformer model first predicts a sequence $\hat{Y}_{\text{new}} = P(X_{\text{new}})$. This predicted sequence is combined with the original observed sequence to form an enriched feature vector

$$V_{\text{new}} = [X_{\text{new}}, \hat{Y}_{\text{new}}],$$

providing the LSTM classifier with both current and predicted data, as illustrated in Figure 1.

**Algorithm 3: Fault classification based on process variables values prediction Inputs:**

- Transformer model $P$
- LSTM classifier $C$
- New sequence $X_{\text{new}}$ of length $w$

**Output:** Fault type

1: $\hat{Y}_{\text{new}} \leftarrow P(X_{\text{new}})$
2: $V_{\text{new}} \leftarrow [X_{\text{new}}, \hat{Y}_{\text{new}}]$
3: **return** $C(V_{\text{new}})$

## 3. CASE STUDY: TENNESSEE EASTMAN PROCESS (TEP)

We evaluate our method using the Tennessee Eastman Process (TEP) benchmark (Downs and Vogel, 1993), which is widely used for fault detection and diagnosis testing. This section describes the TEP, the performance metrics employed, data preprocessing, training procedures, and the classification results with and without prediction.

### *3.1 Process description and dataset*

The TEP generates two products, G and H, from four reactants (A, C, D, and E), along with an inert component (B) and a byproduct (F). The process consists of five key units: reactor, condenser, separator, stripper, and compressor. A total of 52 variables can be monitored from this process, including 22 process sensors, 18 composition measurements, and 12 manipulated variables.

The original dataset comprises 500 simulations, each lasting 25 hours and generated with unique random seeds, covering 20 fault conditions and a normal state. Process variables values are sampled every three minutes. In this study, only the first 100 samples of each run were used for training and evaluation, focusing the model on learning the initial dynamics of faults. This approach reflects real-world scenarios, where faults are typically addressed promptly to prevent prolonged process disruption. The evaluation specifically targeted selected faults with the longest detection delays (Lomov et al., 2021), as shown in Table 1, and the 12 most affected variables, shown in Table 2, identified using a Random Forest classifier (Lovatti et al., 2019).

Table 1. Selected faults

| Fault ID | Description | Type |
|---|---|---|
| 0 | Nominal operation | - |
| 8 | A, B, C feed composition (Stream 4) | Random variation |
| 10 | C feed temperature (Stream 4) | Random variation |
| 13 | Reaction kinetics | Slow drift |
| 17,18,20 | Unknown | Unknown |

Table 2. Selected variables

| No. | Variable Name | Units |
|---|---|---|
| 1 | Reactor cooling water outlet temperature | °C |
| 2 | Stripper temperature | °C |
| 3 | Compressor Recycle Valve | % |
| 4 | Separator cooling water outlet temperature | °C |
| 5 | Stripper pressure | kPa |
| 6 | Reactor pressure | kPa |
| 7 | Stripper steam flow | kg $h^{-1}$ |
| 8 | Product separator pressure | kPa |
| 9 | Compressor Work | kW |
| 10 | Purge %A | mol% |
| 11 | Product separator temperature | °C |
| 12 | Reactor cooling water flow | $m^3h^{-1}$ |

### *3.2 Performance metrics*

The evaluation of the performance of the proposed method was conducted using precision for classification and mean absolute error (MAE) for prediction.

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{1}$$

where $TP$ represents the true positives and $FP$ represents the false positives.

For prediction, the mean absolute error (MAE) is used to measure the average magnitude of errors between the predicted and actual values. It is calculated as follows:

$$\text{MAE} = \frac{1}{nH} \sum_{i=1}^{n} \sum_{j=1}^{H} \left| y_{ij} - \hat{y}_{ij} \right|, \tag{2}$$

where $n$ is the number of variables, $H$ is the prediction horizon, $y_{ij}$ is the actual value of the $i$-th variable at the $j$-th timestep, and $\hat{y}_{ij}$ is the predicted value of the $i$-th variable at the $j$-th timestep.

### *3.3 Data Pre-processing and Training*

We used the first 100 timesteps (300 minutes) of each simulation for both fault classification and variable prediction. Table 3 summarizes the data distribution and preprocessing steps. All process variables values (nominal and faulty runs) were z-score normalized based on the sample mean and standard deviation of the normal state (fault-free) training data, using

$$x'_i = \left| \frac{x_i - \bar{x}_i}{s_i} \right|, \tag{3}$$

where $\bar{x}_i$ and $s_i$ denote the mean and standard deviation (respectively) of the nominal training data for variable $i$. The normalized data were reshaped into sliding windows with stride 1 for classification (window size $w$) and prediction ($w$ plus horizon $H$).

Table 3. Data distribution and preprocessing details

| Item | Description |
|---|---|
| **Number of variables** | 12 selected key variables |
| **Number of faults** | 6 fault types + 1 nominal |
| **Samples per simulation** | 100 timesteps (300 minutes) |
| **Training / Validation / Testing** | *For classification (LSTM)*<br>*50 / 20 / 30 runs per fault* |
| **Training / Validation / Testing** | *For prediction (Transformer)*<br>*250 / 20 / 30 runs total* |
| **Normalization** | Z-score using Eq. (3) |
| **Lookback window ($w$)** | Varies in {5, 10, ..., 85} |
| **Prediction horizon ($H$)** | Varies in {0, 5, 10, ..., 85} |

### *3.4 Hyperparameter Tuning*

We use three LSTM layers with decreasing hidden units (128, 100, and 50), each followed by a 20% dropout layer to prevent overfitting, and a final dense layer with seven output units (one nominal plus six faults) and softmax activation. The transformer for prediction has four encoder and four decoder blocks, each with eight attention heads, a model dimension $d_{\text{model}} = 128$, and feed-forward dimension $\text{dim}_{\text{FFN}} = 256$. Early stopping with a patience of 20 epochs is adopted for both models (maximum 100 epochs), and the Adam optimizer is used with categorical cross-entropy (LSTM) or mean squared error (transformer) losses.

To determine the smallest lookback window $w$ and the largest prediction horizon $H$ that yield the highest classification performance for a sequence of length $w + H$, we begin by measuring classification precision without making any predictions, while varying the sequence length. As shown in the figure 2, when the sequence length is 90, the classification precision reaches 100%. We therefore select 90 as the maximum value for $w + H$. Next, to find the best compromise between a minimal lookback window, a maximal horizon, and acceptable classification precision, we train $17 \times 17$ models corresponding to all $(w, H)$ pairs such that $w$ and $H$ vary in steps of 5 from 5 to 85, under the constraint $w + H \leq 90$. This approach enables us to compare every possible $(w, H)$ configuration within the specified range and identify the option that offers the best trade-off between the size of the historical window, the prediction horizon, and the resulting classification precision.

### *3.5 Classification results*

Figure 2 presents the classification precision obtained for faults 8, 10, 13, 17, 18, and 20 over time intervals ranging from 5 to 90 timesteps. The time intervals are indicated in timesteps, where one timestep equals 3 minutes. This

sequence length represents the ideal scenario for fault diagnosis without using prediction. However, our objective is to test different combinations of $w$ and $H$ such that $w + H$ equals the sequence length, in order to find the optimal combination.

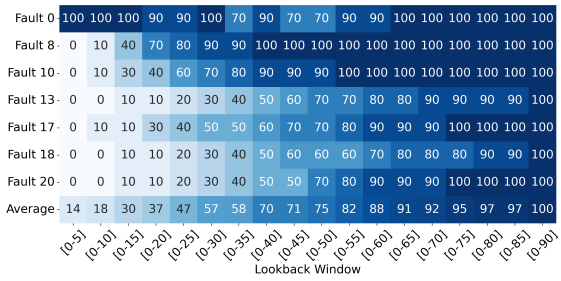



Fig. 2. Classification precision (%) across fault scenarios for different Lookback Windows.

In the first five timesteps after fault introduction, the LSTM-based classification model categorizes all faults and the nominal state as fault-free, since the faults' effects are not yet observable on the process variables values. From timesteps 6 to 15, classification precision improves from 0%, achieving high value for various faults while perfectly classifying the normal operating state. Faults 8 and 10 reach 100% precision during the [0–40] and [0–55] intervals, respectively. During this period, precision for the fault-free state fluctuates due to emerging fault signatures causing occasional misclassifications with normal operations. Beyond the 60th timestep, precision for the fault-free state and faults 8 and 10 reaches 100%, with other faults progressively achieving 100% by the 90th timestep as sufficient data become available for distinct classification, though some faults may still be misclassified as normal due to subtler signatures. The last row in Figure 2 shows a steady increase in average classification precision across faulty and non-faulty scenarios, rising from approximately 14% at timesteps [0–5] to 100% at [0–90].

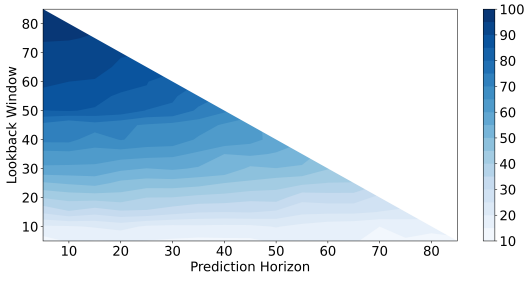



Fig. 3. Classification precisions across varying lookback lengths and prediction horizons.

Figure 3 shows how the classification precision changes for different combinations of lookback $w$ and prediction horizon $H$, focusing on a 90-timestep total length ($w+H = 90$). We see that the highest classification precisions appear at relatively large $w$ and short $H$. By comparing the classification precisions across different lookback lengths ($w$) and prediction horizons ($H$) in Figure 3 with the precision patterns shown over time in Figure 2, we derive the results presented in Figure 4. These results suggest that $w = 50$ and $H = 10$ deliver the best compromise between average classification precision and early diagnosis.

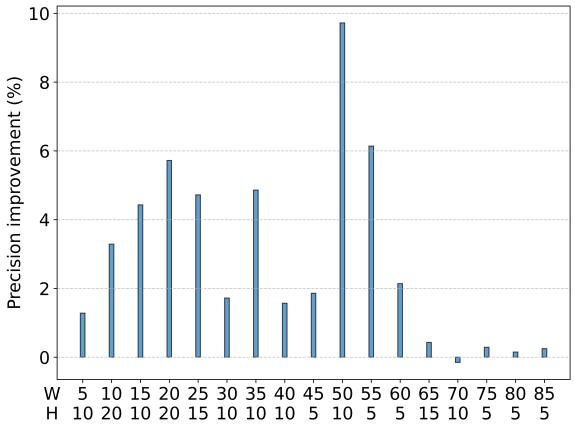
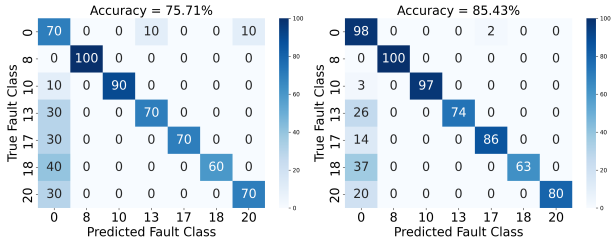



Fig. 4. Precision improvement for different sequence lengths $w$ with prediction. The selected horizon $H$ corresponds to the prediction horizon that yields the best precision for the corresponding $w$.

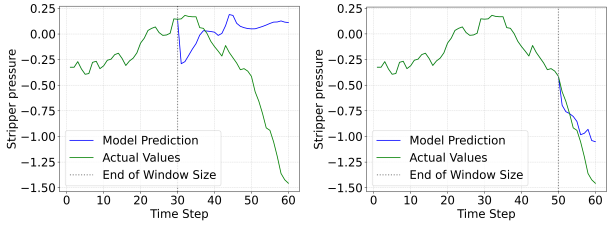
Figure 5 compares the obtained confusion matrices without (a) and with (b) prediction enrichment for a lookback of 50 timesteps. With a lookback of 50 timesteps and a prediction horizon of 10 timesteps, the proposed approach achieves an overall improvement in classification precision of approximately 10%.




(a) (b)

Fig. 5. Confusion matrix with $w = 50$. (a): without prediction ($H = 0$), (b): with prediction ($H = 10$).

This improvement is observed across the seven tested fault scenarios, particularly for nominal operation and Fault 17, where the classification precision is significantly enhanced.




(a) (b)

Fig. 6. Example of predicted (blue) vs. actual (green) stripper pressure for Fault 17. The vertical dotted line indicates the transition from observed data to predictions. (a): $w = 30$, $H = 30$; (b): $w = 50$, $H = 10$.

Figure 6 shows predicted (blue) vs. actual (green) stripper pressure for Fault 17, comparing two different lookback windows. The vertical dotted line marks where the historical (observed) data ends and the model's predictions begin.

In Figure 6(a), with a 30-time-step lookback and a 30-time-step prediction horizon, the model captures the overall downward trend but gradually deviates from the actual measurements, indicating difficulties with short-term fluctuations. In contrast, Figure 6(b), which uses a 50-time-step lookback and a 10-time-step prediction horizon, produces predictions that closely match the measured values, suggesting that a longer lookback window improves the model's ability to track underlying patterns accurately.

### *3.6 Discussions*

Our results show that combining a short-horizon prediction with a sufficiently long lookback window can improve early fault classification in the Tennessee Eastman Process. In particular, setting $w = 50$ timesteps (150 minutes) and $H = 10$ timesteps (30 minutes) allows an approximately 85% macro-average F1 score at the 60th timestep, compared to about 75% when using all 60 timesteps purely for historical data. Thus, the proposed approach can diagnose faults about 30 minutes earlier at a similar precision level.

For faults that evolve slowly (e.g., Fault 13), we observed a smaller improvement because their progressive nature requires more historical data to distinguish them from normal conditions, making short-horizon predictions less effective. Nonetheless, the combined approach still achieves a better balance between early diagnosis and classification performance compared to using classification alone. We also considered evaluating other baseline methods from the TEP fault diagnosis literature. While a direct comparison is beyond the scope of this paper, our approach focuses on improving the timing of fault diagnosis by leveraging short-term predictions rather than solely maximizing final precision.

## 4. CONCLUSION

In this work, we proposed an early fault diagnosis method for chemical processes by combining multistep transformer-based prediction of process variables values with LSTM-based classification. Tests on the benchmark Tennessee Eastman Process demonstrated that our approach improves classification precision by 10% using data from the first 150 minutes after the introduction of faults 8, 10, 13, 17, 18, and 20. This provides the operator with an additional 30 minutes to intervene, compared to the time normally required to achieve this level of precision with a conventional LSTM-based classification model.

Future work will focus on refining the prediction models by optimizing parameters, such as the number of decoder and encoder layers, and exploring the method's applicability to a wider range of fault types and industrial processes.

## REFERENCES


Bai, Y. and Zhao, J. (2023). A novel transformer-based multi-variable multi-step prediction method for chemical process fault prognosis. *Process Safety and Environmental Protection*, 169, 937–947.

Bi, X., Qin, R., Wu, D., Zheng, S., and Zhao, J. (2022). One step forward for smart chemical process fault detection and diagnosis. *Computers & Chemical Engineering*, 164, 107884.

Chiang, L.H. (2000a). Data-driven methods for fault detection and diagnosis in chemical processes.

Chiang, L. (2000b). *Fault Detection and Diagnosis in Industrial Systems*. Springer Science & Business Media.

Downs, J.J. and Vogel, E.F. (1993). A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3), 245–255.

Han, Y., Ding, N., Geng, Z., Wang, Z., and Chu, C. (2020). An optimized long short-term memory network based fault diagnosis model for chemical processes. *Journal of Process Control*, 92, 161–168.

Lee, J.H., Shin, J., and Realff, M.J. (2018). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, 114, 111–121.

Lomov, I., Lyubimov, M., Makarov, I., and Zhukov, L.E. (2021). Fault detection in tennessee eastman process with temporal deep learning models. *Journal of Industrial Information Integration*, 23, 100216.

Lovatti, B.P., Nascimento, M.H., Neto, Á.C., Castro, E.V., and Filgueiras, P.R. (2019). Use of random forest in the identification of important variables. *Microchemical Journal*, 145, 1129–1134.

Md Nor, N., Che Hassan, C.R., and Hussain, M.A. (2020). A review of data-driven fault detection and diagnosis methods: Applications in chemical process systems. *Reviews in Chemical Engineering*, 36(4), 513–553.

Meyer, D. and Wien, F. (2001). Support vector machines. *R News*, 1(3), 23–26.

Peterson, L.E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2), 1883.

Qin, S.J. and Chiang, L.H. (2019). Advances and opportunities in machine learning for process data analytics. *Computers & Chemical Engineering*, 126, 465–473.

Russell, E.L., Chiang, L.H., Braatz, R.D., Russell, E.L., Chiang, L.H., and Braatz, R.D. (2000). Fisher discriminant analysis. *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*, 53–65.

Sitapure, N. and Kwon, J.S.I. (2023). Exploring the potential of time-series transformers for process modeling and control in chemical systems: an inevitable paradigm shift? *Chemical Engineering Research and Design*, 194, 461–477.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wei, Z., Ji, X., Zhou, L., Dang, Y., and Dai, Y. (2022). A novel deep learning model based on target transformer for fault diagnosis of chemical process. *Process safety and environmental protection*, 167, 480–492.

Zhang, S., Bi, K., and Qiu, T. (2019). Bidirectional recurrent neural network-based chemical process fault diagnosis. *Industrial & Engineering Chemistry Research*, 59(2), 824–834.

Zhao, H., Sun, S., and Jin, B. (2018). Sequential fault diagnosis based on lstm neural network. *IEEE Access*, 6, 12929–12939. doi:10.1109/ACCESS.2018.2794765.