A novel multi-step prediction model for process monitoring

Yi Shan Lee [a,1], Sai Kit Ooi [b,2], Junghui Chen [a,3,*]

[a] Department of Chemical Engineering, Chung Yuan Christian University, Chung-Li, Taoyuan, Taiwan, R.O.C
[b] United Microelectronics Corporation (Singapore Branch), Singapore
[1] bmtys1995@gmail.com; [2] kittyooi_123@hotmail.com; [3] jason@wavenet.cycu.edu.tw
* corresponding author

ARTICLE INFO

Article history
Received July 15, 2023
Revised September 14, 2023
Accepted March 9, 2024
Available online May 31, 2024

Selected paper from The 2023 6th International Symposium on Advanced Intelligent Informatics (SAIN'23), Yogyakarta (Virtually), September 21, 2023, http://sain.ijain.org/2023/. Peer-reviewed by SAIN'23 Scientific Committee and Editorial Team of IJAIN journal.

Keywords
Multi-step prediction
Early warning
Dynamic model
Probabilistic model
State-space model

ABSTRACT

In the competitive market, process monitoring can ensure the quality of products, but strong nonlinearities, slow dynamics, and uncertainties characterize the complexities of the large-scale chemical plant. When the fault occurs, it will not influence the process instantaneously but will react after a few time points. After all the products affected by the faults are inspected, it is too late to fix the process. Conventional approaches neither do nor care about early detection before any disturbance significantly affects the process. To estimate disturbances propagated through the process, a multi-step prediction model is essential. The purpose of early process monitoring is to detect any problem with the currently running process as early as possible. In this paper, a multi-step prediction system is proposed. The system is a dynamic model that can capture the dynamic relationship of past process input variables and future process output variables. It provides a lower dimension and a lower noise-contaminated space for data analysis. Particularly, the past input and output process data can be mapped from the observation space into the latent space to acquire their intrinsic properties. The latent variables preserve the dynamic information for future multi-step prediction so that early warning can be achieved. An industrial example of the PVC dying process is presented to show the multistep predictive ability of the proposed method.

1. Introduction

Statistical process monitoring methods are necessary for effective monitoring of the production status of a process plant to keep quality consistency and operation safety. However, the nonlinearity, process dynamics, and process uncertainties in chemical processes make online monitoring challenging. Due to the complexity in large processes, the process model based on first-principle knowledge is very difficult to be obtained. In contrast, the data-driven models which is able to perform well in process modeling without any precise physical knowledge of the process attracts a lot of attention from the industry. Dynamic partial least squares (DPLS) [1] and dynamic inner PLS [2] were developed to describe the dynamic properties of processes, but both of them cannot handle the nonlinear behavior of the systems. To handle nonlinear dynamic processes, the kernel trick is used in DPLS. Its shallow structure with a single kernel function confines the kernel-based dynamic PLS in the highly nonlinear process. Also, the large historical data would cause a heavy computational burden. Recently, recurrent neural networks (RNNs) [3], [4] and long-short term memory (LSTM) [5]–[7] are widely used as soft-sensor models because of their ability to represent the nonlinear dynamic feature. To keep a lower dimension and a

lower noise-contaminated space while constructing dynamic models, integrate LSTM or RNN with latent variable models. The Gaussian-Bernoulli restricted Boltzmann machine [8] and an ensemble model that integrates the stacked autoencoder and bi-directional LSTM were developed [9]. Although the aforementioned methods can learn the dynamic properties of processes, they are only used to estimate and monitor the data at the current time point only, not at the future multi-step time points, so the monitoring models neither do nor care about early detection before any disturbance significantly affects the process. To our best knowledge, the performance of their applications on early warning monitoring has rarely been researched.

To realize a model which detects the influence of faults at the early time, a multistep prediction model which can describe the temporal properties of the process as well as predict the future trend properties of the process is required. At present, approaches to multistep prediction are mainly developed under some fundamental multistep prediction strategies [10]. In the iterative and direct strategy, nonlinear autoregressive with exogenous input (NARX) was proposed [11]. With a one-step-ahead NARX developed to predict the output in the next time point, the forecasted values at the future time point are used subsequently and recursively as inputs for predictions at future time points until the entire prediction horizon is forecasted. The drawback of NARX with the iterative strategy may yield the accumulation of errors and the dimension of data would increase with the expansion of the window size. This certainly increases the computation loading and makes the performance of the model affected by the redundancy of variables and the noise of data to a certain extent. To eliminate the impact of correlated variables and noise, the model in the latent feature space instead of the observation space is highly attractive. The optimally pruned extreme learning model (OP-ELM) with the direct strategy [12]–[14] was proposed to perform the multi-step prediction. It is a multi-model approach. The number of one-step-ahead prediction models depends on the number of prediction horizons. And the prediction of each model in each horizon is independent of the others. The direct strategy can prevent the problem of error accumulation, but it would induce conditional independence of the prediction. Like ELM, OP-ELM has a fast computation speed, but it is considered a shallow model, so it may not be able to describe the complicated industrial process.

To describe the complex dynamic behavior of processes, RNN and LSTM are widely used to develop multi-step prediction tasks in different fields such as multi-step prediction for cutterhead torque [15], travel time prediction [16] etc. Also, transductive LSTM for time series prediction is applied to weather forecasting [17]. The LSTM autoencoder neural network (LSTMED) [18], [19], which is a task-oriented method for uneven dynamic process monitoring model, was developed in an unsupervised manner to solve the problem of insufficient faulty data [20]. Particularly, the approach which can expertise the important input information is conducted into the modeling of multistep methods. The denoising spatial-temporal encoder-decoder (DSTED) [21] which integrated denoising gated recurrent unit (DGRU) [22], [23] and sequence-to-sequence spatial-temporal encoder-decoder network was proposed to predict the burn-through point of ore sintering process. The LSTM-temporal convolutional network, which is a temporal emphasized model, is proposed in [24] to predict the photovoltaic power multistep ahead to control the intermittency of solar energy. Hybrid methods, such as WT-ED-LSTM, which consists of wavelet transformation, encoding-decoding based on LSTM and prediction LSTM with attention mechanism, and VMD-LSTM, which integrated variation mode decomposition (VMD) [25], [26], attention-LSTM [27], and support vector regression (SVR) [28], [29], were proposed for coal price forecast [30]. Although the aforementioned method looks promising in multistep prediction, but there were developed as deterministic models; thus, they cannot demonstrate the uncertain nature in reality.

Most predictive models developed in the past were focused on predicting the data at the current time. When any disturbances occur, the disturbance would propagate through the entire process before the qualities are significantly affected. After all the products are inspected, it is too late to fix the process. If the abnormal occurrence, which does not significantly influence the process at the beginning, can still be detected as early as possible, the operators can compensate for the disturbance before the quality

significantly deviates from the desired specification. In this paper, a multi-step prediction system is proposed. The multi-step prediction system is a dynamic model and it is called the Multi-NSSM, which is short for Multi-Step Nonlinear State Space Model. Multi-NSSM captures the dynamic relationship of manipulated, process input variables and process output variables. Multi-NSSM integrates the encoder-decoder and the probability state-space model to efficiently extract the latent features for future process output probability estimation at multi-ahead time points. Specifically, a multi-step prediction system provides a lower dimension and a lower noise-contaminated space for data analysis. The past input and the past output process data can be mapped from the observation space into the latent space to acquire the intrinsic properties. They preserve the dynamic information for future multi-step predictions so that early warning can be achieved. In contrast to one-step-ahead prediction learning scheme, the multi-step ahead prediction learning scheme makes sure that the prediction result within the prediction horizon is good on average. Thus, multi-step ahead prediction can reject the effect of the noisy data and stabilize the probability estimation of the future data.

2. Method

In a typical operation, there are 4 main types of variables: manipulated variables, process input variables, process output variables, and quality variables. Manipulated variables are the ones that maintain some of the process output variables in the desired condition, like the control valve openings while process input variables are any variables which are not adjustable, like the upstream flows, heating steams, and the other source inputs to the process, but they cause the changes of process output variables. The set of time-series data of both manipulated and process input variables are symbolized by $\mathbf{U}=\left\{\mathbf{u}_t \in \square^{M}, t=1,\cdots,T\right\}$ here. To make simple, compact explanations, only process input variables representing both manipulated and process input variables are discussed as follows. Process output variables refer to the variables that reflect the process operating status such as temperatures, pressures, etc. of the process; it is symbolized by $\mathbf{X}=\left\{\mathbf{x}_t \in \square^{N}, t=1,\cdots,T\right\}$. In the large-scale system, process output variables cannot be completely and immediately affected by the changes of the manipulated or process input variables due to the slow dynamic behaviors of the operating system. Thus, there is a dynamic relationship between $\mathbf{U}$ (the process input variables) and $\mathbf{X}$ (the process output variables). The dynamic behaviors of process input variables ($\mathbf{u}_t$) and process output variables ($\mathbf{x}_t$) can be represented by a stochastic nonlinear state-space model. They are written as.

$$\mathbf{z}_t = f_u(\mathbf{u}_t, \mathbf{d}_t) + \mathbf{m}_t$$

$$\mathbf{z}_{t+1} = f_z(\mathbf{z}_t) + \boldsymbol{\varepsilon}_{t+1}$$

$$\mathbf{x}_{t+1} = g_x(\mathbf{z}_{t+1}) + \mathbf{n}_{t+1} \quad (1)$$

where $\mathbf{z}_t$ represents the state-space variables extracted from process input variables and unmeasured disturbances ($\mathbf{d}_t$). Meanwhile, $\mathbf{z}_t$ also represents the latent variable (LV) of the process model. f_u is the vector mapping function that describes the observation variables $(\mathbf{u}_t)$ from the observation space into the latent space to acquire the inherent properties of the process. As the nonlinear dynamic process is depicted in the latent space, f_z, which is a step-ahead prediction vector function, predicts the dynamic nonlinear behavior of the process in the next time step. With the corresponding step-ahead predicted latent variables $(\mathbf{z}_{t+1})$, g_x is the unknown vector mapping functions that reconstruct the observations of the originally predicted process variable. $\boldsymbol{\varepsilon}_t$ and $\mathbf{m}_t$ are the process noise associated with dynamic changes in latent variables while $\mathbf{n}_t$ represents the observation noise of the process data. All noises assume to follow zero-mean Gaussian distributions, $p(\mathbf{m}_t) \sim N(0, \mathbf{I})$, $p(\boldsymbol{\varepsilon}_t) \sim N(0, \mathbf{I})$ and $p(\mathbf{n}_t) \sim N(0, \boldsymbol{\Gamma})$, where the covariance matrices of $\boldsymbol{\varepsilon}_t$ and $\mathbf{m}_t$ are assumed to be identity matrices.

To predict the future trend of the process output variables, the dynamic transition in (1) for one-time point to the future time points should be rewritten as :

$$\mathbf{z}_{P_t} = f_U(\mathbf{U}_{Pt}, \mathbf{D}_{Pt})$$

$$\mathbf{z}_{F_t} = f_z(\mathbf{Z}_{Pt})$$

$$\mathbf{z}_{F_t} = g_x(\mathbf{Z}_{Ft}) \quad (2)$$

where $\mathbf{Z}_{P_t} = \left[\mathbf{z}_t, \mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau_p}\right]$ is the state-space variables from the current and previous τ_p time points while $\mathbf{Z}_{F_t} = \left[\mathbf{z}_{t+1}, \mathbf{z}_{t+2}, \cdots, \mathbf{z}_{t+\tau_f}\right]$ denotes the state variables of the future τ_f time points. The definitions of $\mathbf{U}_{P_t}$, $\mathbf{D}_{P_t}$, and $\mathbf{X}_{F_t}$ are similar to those of $\mathbf{Z}_{P_t}$ and . $\mathbf{Z}_{F_t}$., but they are not detailed here. As the disturbances are unmeasurable, their information would be inherited by the past measured process outputs, so $f_{\mathrm{U}}\left(\mathbf{U}_{p_t}, \mathbf{D}_{P_t}\right)$ can be replaced by $f_{\mathrm{U}}\left(\mathbf{U}_{p_t}, \mathbf{X}_{P_t}\right)$. Thus, $f_{\mathrm{U}}\left(\mathbf{U}_{p_t}, \mathbf{X}_{P_t}\right)$ is a dynamic vector mapping function that maps process input variables and process output variables from the observation space into the latent space to extract the dynamic characteristics of the data. $f_{\mathbf{Z}}\left(\mathbf{Z}_{p_t}\right)$ is the multi-step prediction vector function that predicts the long-term future trend of the state in the latent space based on $\mathbf{Z}_{P_t}$. With the predicted latent variables in the future time points, the corresponding long-term future process output variables $\left(\mathbf{X}_{F_t}\right)$ can be reconstructed by the vector mapping functions, $g_{\mathrm{X}}\left(\mathbf{Z}_{F_t}\right)$. With noise probability distributions, the probability distributions of the state-space model in (2) can be further rewritten in a conditional probability form.

$$p(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t}, \mathbf{X}_{P_t}) \sim N\left(f_{\mathrm{U}}\left(\mathbf{U}_{P_t}\right), \mathbf{I}\right)$$

$$p(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t}) \sim N\left(f_{\mathrm{z}}\left(\mathbf{Z}_{P_t}\right), \mathbf{I}\right)$$

$$p(\mathbf{X}_{F_t}|\mathbf{Z}_{F_t}) \sim N\left(g_{\mathrm{x}}\left(\mathbf{Z}_{F_t}\right), \mathbf{\Gamma}\right) \quad \textbf{(3)}$$

$p\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ in (3) describes the state estimation transition of the latent variables from the current and previous time points to the future time points. $p\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ is referred to as estimation transition distribution and $p\left(\mathbf{X}_{F_t} \mid \mathbf{Z}_{F_t}\right)$ is known as emission distribution for reconstructing the future process output variables. To acquire this multi-step prediction state-space model, the functions and the parameters $\left\{f_{\mathrm{U}}(\square), f_{\mathbf{Z}}(\square), \mathrm{g}_{\mathbf{X}}(\square), \mathbf{\Gamma}\right\}$ in (3) should be estimated by maximizing the conditional log-likelihood function of the future process output variables given process input variables. The modeling scheme will be detailed below.

To have the model predict the future process outputs by the past process inputs and outputs, the goal of the modeling scheme is set to maximize the probability likelihood of the future process output estimates conditioned on the past process inputs and outputs within the window sequential data. The maximum probability is defined as

$$\max \mathrm{In} \prod_{t=1}^{T-t_f-t_p} p_\theta(\mathbf{X}_{F_t} | \mathbf{U}_{P_t}, \mathbf{X}_{P_t}) \quad (4)$$

The probability likelihood of the whole sequence with T samples can be represented by multiplying the probability likelihood of the window data $T-\tau_f-\tau_p$. $\mathbf{X}_{P_t} = \left[\mathbf{x}_{t-\tau_p}, \mathbf{x}_{t-\tau_p+1}, \cdots, \mathbf{x}_t\right]$,

$\mathbf{U}_{P_t} = \left[\mathbf{u}_{t-\tau_p}, \mathbf{u}_{t-\tau_p+1}, \cdots, \mathbf{u}_t\right]$, and $\mathbf{X}_{F_t} = \left[\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \cdots, \mathbf{x}_{t+\tau_f}\right]$, where τ_f refers to the multi-step prediction horizon while τ_p represents the length of the window data with the past measurements. Henceforth, for easy and compact expression, only time t window sequential data as (4) is tentatively considered in the following derivations. Time series process show as Fig. 1.

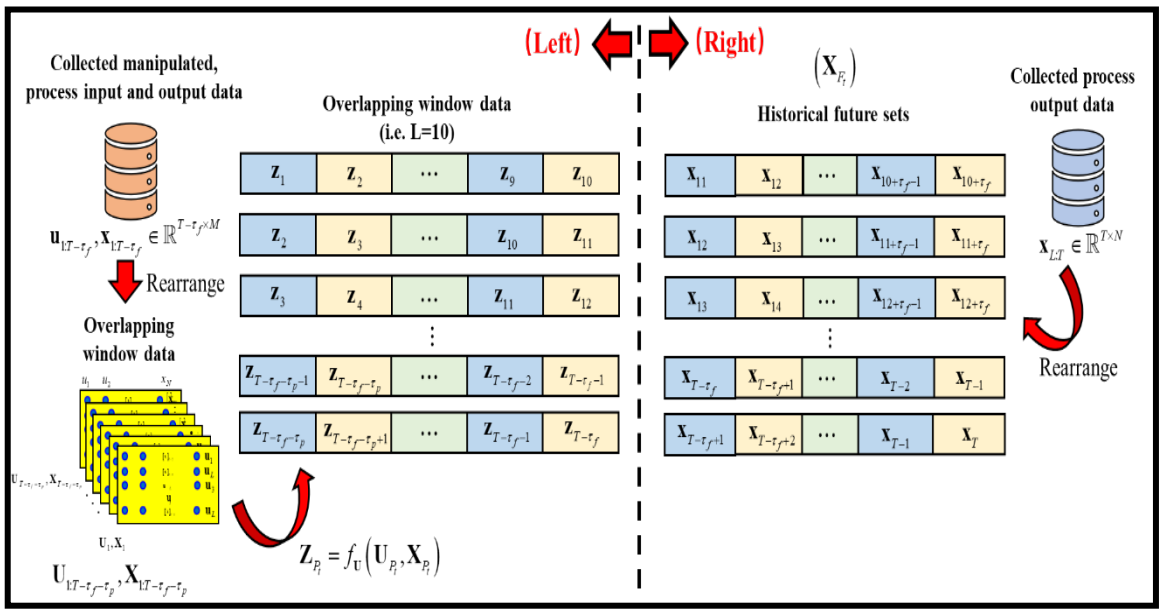


Fig. 1. Time-series process input and output data are mapped from the observation space into the latent space. Then rearrange the time-series latent variables (the left side) and the future time-series process outputs (the right side) into 2-dimensional window data

In large-scale processes, climate and environmental changes and the various types of sensor errors cannot be avoided, so there are uncertain measurement data. To have a robust model, the proposed method is used to establish a multistep predictive model that can predict the long-term trend of the process outputs at the future time points in the latent space. The latent features $(\mathbf{Z})$ can be obtained through mapping $T-\tau_f-\tau_p$ overlapping windows with the process input and output data $(\mathbf{U},\mathbf{X})$ into the latent variables $\mathbf{Z}=\{\mathbf{Z}_t\};t=1,\cdots,T-\tau_f-\tau_p$ (left side of Fig. 1). The historical future process outputs are rearranged into 2-dimensional window data, (right side of Fig. 1). The window data are assigned as the target to train the model, which has a future prediction ability.

To obtain the inner behaviors of the processes, measurement data can be mapped from the observation space into the latent space. Assume process output variables are the function of latent variables. The conditional probability distribution can be expressed as

$$p_\theta(\mathbf{X}_{F_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}) = \iint p(\mathbf{X}_{F_t}|\mathbf{Z}_{F_t}) \underbrace{p(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t})p(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t})}_{p_\theta(\mathbf{Z}_{F_t},\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t})} d\mathbf{Z}_{F_t} d\mathbf{Z}_{P_t} \tag{5}$$

With Bayes and Jensen equality, $\ln p_\theta\left(\mathbf{X}_{F_t} \mid \mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$ can be further rearranged into the variational lower bound, written as

$$L(\mathbf{X}_{F_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}) = E_{q_\emptyset(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t})}[\ln p_\theta(\mathbf{X}_{F_t}|\mathbf{Z}_{F_t}) \tag{6}$$

$$-E_{q_\emptyset(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t})}[KL\left(q_\emptyset(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t})\middle|\middle|p_\theta(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t})\right)]$$

$$-KL(q_\emptyset(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t})|\left|p_\theta(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t})\right)$$

The whole Multi-NSSM model structure is presented in Fig. 2 and it contains three parts of the models, including the posterior distribution model ($q_\phi\left(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$ and $p_\theta\left(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$), the sequence-to-sequence encoder-decoder network ($q_\phi\left(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t}\right)$ and $p_\theta\left(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t}\right)$), and the reconstructed decoder network ($p_\theta\left(\mathbf{X}_{F_t}|\mathbf{Z}_{F_t}\right)$). The description of each part in the variational lower bound is presented in the subsections below.

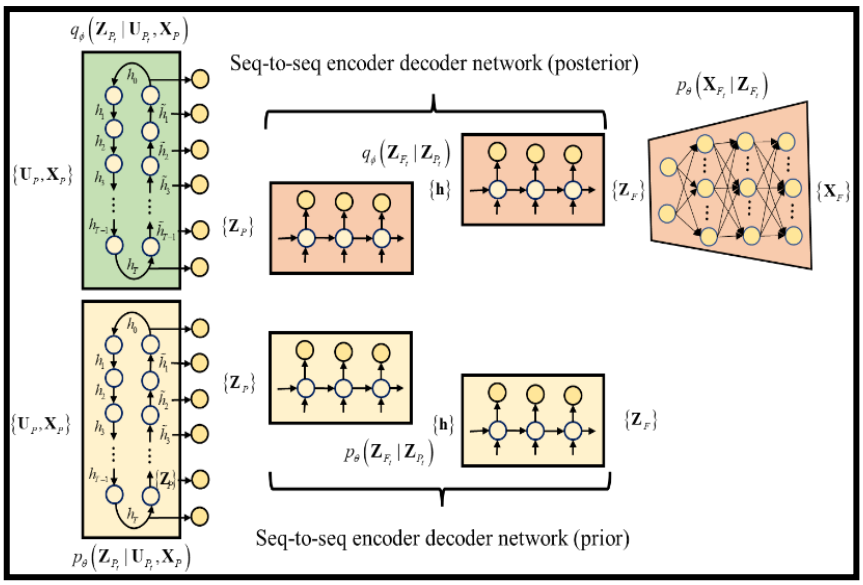


Fig. 2. Structure of Multi-NSSM model

2.1. Posterior distribution model ($q_\phi\left(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$ and $p_\theta\left(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$)

Before going through the multi-step ahead predictions in the latent space, the main task is to extract the dynamic latent features at the current and past time points. The inference posterior distribution $q\left(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$ and the true posterior distribution $p\left(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}\right)$ are used to map the process input variables from the observation space into the latent space to obtain the inherent process latent behaviors. Due to the slow dynamic behaviors in the large-scale process, the state of the process will be affected by sequential information, shown as

$$\mathbf{h}_t = \mathbf{\Lambda}(\mathbf{u}_{t-\tau_p:t}, \mathbf{x}_{t-\tau_p:t}) \tag{7}$$

also can be denoted as

$$\mathbf{h}_t = \mathbf{\Lambda}(\mathbf{h}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \tag{8}$$

represents the forward RNN transition while. (10) demonstrates the backward RNN.

$$\tilde{\mathbf{h}}_t = \mathbf{\Lambda}(\tilde{\mathbf{h}}_{t+1}, \mathbf{u}_{t-\tau_p:t}, \mathbf{x}_{t-\tau_p:t}) \tag{9}$$

The bi-directional recurrent neural network (Bi-RNN) is used to map the observation process input variables into the RNN hidden cell state $\mathbf{h}$. The forward RNN learns the dynamic transition information of sequential data from $t-1$ to t while the backward RNN captures the dynamic properties from t to $t-1$ to enhance the information of the latent features. The initial states of the backward RNN can be obtained via the final hidden state of the forward RNN; on the contrary, the initial states of the forward RNN can be obtained from the final hidden state of the backward RNN. One of the advantages of using Bi-RNN is smoothening the initial state of the model; thus, model precision increases and model

overfitting is prevented. The Bi-RNN structure of $p\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$ is almost the same as $q\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$. The only difference is that $q\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$ is changed to $p\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$.

To obtain the latent feature $\mathbf{Z}_p$, Kalman smoother is used to transform the hidden state from Bi-RNN into the latent space to form the posterior distributions $q\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$ and $p\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$. Then the similarity of the distributions between $q\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$ and $p\left(\mathbf{Z}_{P_t} \mid \mathbf{U}_{P_t}, \mathbf{X}_{P_t}\right)$ in the third term of the right side of(6) can be calculated by the KL divergence.

$$KL(q_\phi(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t}, \mathbf{X}_{P_t})| \left|p_\theta(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t}, \mathbf{X}_{P_t})\right) \tag{10}$$

2.2. Sequence-to-sequence encoder decoder network ($q_\phi\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ and $p_\theta\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$)

After the latent feature $\mathbf{Z}_{P_t}$ is obtained, the multistep transition posterior distribution $q\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ and the multi-step transition prior distribution $p\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ can be estimated through a sequence-to-sequence encoder-decoder network. In the sequence-to-sequence network, the encoder mainly captures the dynamic characteristics of $\mathbf{Z}_{P_t}$, which denote the hidden cell states $\left(\mathbf{h}_{t-\tau_p}, \cdots, \mathbf{h}_{t-1}, \mathbf{h}_t\right)$, while the decoder predicts the dynamic characteristics of $\mathbf{Z}_{F_t}$, which denote the hidden cell states $\left(\mathbf{h}_{t+1}, \cdots, \mathbf{h}_{t-1}, \mathbf{h}_{t+\tau_F}\right)$, but the final hidden cell states $\left(\mathbf{h}_t\right)$ of the encoder will be used as the initial states of the decoder. The hidden cell states presented in the sequence-to-sequence encoder decoder network are different from those mentioned in posterior distribution models. To reduce notation usage, the same notations are used.

Although the entire historical time series data in the training phase are available, the future outputs will never be known during the online implementation. If the future latent features are used as the input of the decoder in training, the predicted latent features $\left\{z_f\right\}; f = t+1 : t+\tau_f$ will only focus on getting the information that allows the predicted process outputs to be as close to the actual process outputs as possible. In the conventional iterative mechanism, the predicted latent features do not learn the dynamic transition. This can lead to an accumulation of errors during online execution. Since the future latent feature $\mathbf{Z}_{F_t}$ is unknown, the available future latent features from time t to $t+\tau_f-1$ are not used as the input of the decoder. To be closer to the actual situation, the inputs of the decoder ($\mathbf{Z}_{F_t}$) are obtained in an iterative manner. In each window data, only the latent features $\left(\mathbf{Z}_t\right)$ from the time point $t-\tau_p$ till t are available for multi-step ahead predictions.

$$\mathbf{Z}_t = \left[\mathbf{z}_{t-\tau_p}, \mathbf{z}_{t-\tau_p+1}, \ldots, \mathbf{z}_t\right] \in^{L \times V} \tag{11}$$

where V represents the number of latent variables. This means that the inputs of the decoder are obtained from the previously estimated latent features iteratively. This training idea can effectively avoid the accumulation of errors within the prediction horizon.

Two sequence-to-sequence encoder-decoder networks for distributions of $q\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ and $p\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ are separately constructed. Then the similarity of τ_f time-step ahead transition distributions between $q\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ and $p\left(\mathbf{Z}_{F_t} \mid \mathbf{Z}_{P_t}\right)$ in the second term of the right side of (6) can be calculated by the KL divergence,

$$KL(q(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t})|\left|p(\mathbf{Z}_{F_t}|\mathbf{Z}_P)\right) \quad (12)$$

2.3. Reconstructed decoder network ($p_\theta\left(\mathbf{X}_{F_t}|\mathbf{Z}_{F_t}\right)$)

With the latent samples $\mathbf{Z}_{F_t}$ of the future transition posterior distribution, the reconstructed process output ($\hat{\mathbf{X}}_F$) can be estimated through the emission network which follows the form of the Gaussian distribution. The reconstructed process output in the variational lower bound can be calculated by

$$E_{q_\emptyset(\mathbf{Z}_{F_t}|\mathbf{Z}_{F_t})}[\text{In } p\left(\mathbf{X}_{F_t}|\mathbf{Z}_{F_t}\right)] = \quad (13)$$

$$\frac{1}{S}\sum_{S=1}^{S}\left[-\frac{1}{2}\left(\mathbf{X}_{F_t}-|\boldsymbol{\mu}_t(\mathbf{Z}_{F_t}^S)\right)\sum_x(\mathbf{Z}_{F_t}^S)(\mathbf{X}_{F_t}-\boldsymbol{\mu}_x(\mathbf{Z}_{F_t}^S))-\frac{1}{2}|\sum_x(\mathbf{Z}_{F_t}^S)|-m\mu\right]$$

where $\mathbf{Z}_{F_t}^s$ is the s-th sampling point drawn from the smoothed posterior $q\left(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t}\right)$.

3. Results and Discussion

The early warning system is mainly decided by designing a fault detection index. For every product, the specification threshold boundary can be designed. The predefined threshold boundary can be used to test whether the multistep-ahead predictions are constrained in the bound of the process trend defined by the normal region. If the multistep-ahead predictions are located out of bound, the process will be operated in an unhealthy condition.

A real ammonia synthesis process is used to evaluate the effectiveness of the proposed method. NH3 is widely used as a crucial ingredient in many productions, so the ammonia synthesis process is very common in reality. In the ammonia synthesis process, the pre-decarburization unit plays an important role in absorbing the carbon dioxide (CO_2) from the process gas (PG) to reduce the concentration of CO_2 and make it eligible for subsequent production. The absorbed CO_2 can be used for further production. The flowsheet of the pre-decarburization unit is presented in Fig. 3.

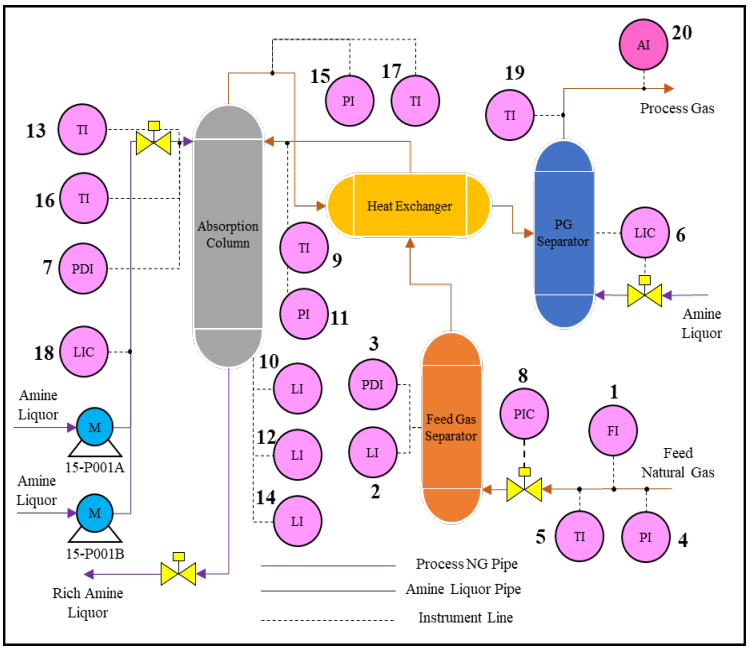


Fig. 3. Flowsheet of the pre-decarburization unit

The pre-decarburization unit consists of 4 major devices, including a feed gas separator, a PG separator, a heat exchanger, and an absorption column. The absorption of CO_2 mainly occurs in the absorption column, and the chemical reactions are given by:

$$\mathrm{RNH_2} + CO_2 \rightarrow RNH^+COO^- \tag{14}$$

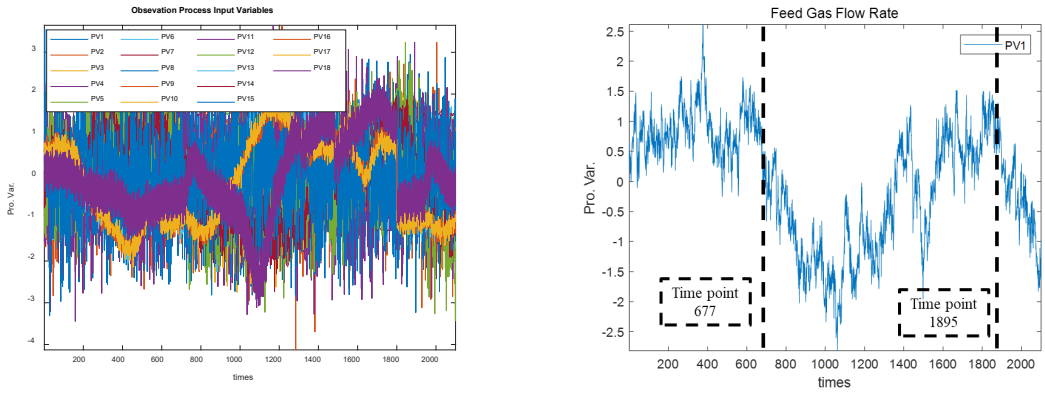
$$\mathrm{RNH_2} + RNH^+COO^- \rightarrow RNHCOO + RNH_3^+ \tag{15}$$

From the perspective of the ammonia synthesis process, its main core absorbs CO2 to make CO2 qualified for subsequent production and reduce the operation risk. Nevertheless, the uncertainty of disturbances such as environmental changes, variation of upstream and the degradations of devices will significantly affect the productivity and safety of the process. To reduce the impact of the disturbances on the concentration of CO2, a multistep ahead prediction model is highly required to demonstrate the influences of current disturbances in the ammonia synthesis process. Therefore, the ammonia synthesis process is chosen to verify the multi-step prediction ability of the proposed method. 19 crucial process variables, including the manipulated process input variables $(\mathbf{U})$ and the process output variables $(\mathbf{X})$, are used to construct the multi-step ahead prediction model. The descriptions of the selected variables are listed in Table 1.

Table 1. Description of the selected variables

Num.	Description	Variables
1	Feed Gas Flow Rate	U
2	Feed Gas Separator Liquid Level	U
3	Feed Gas Separator Pressure Diff.	U
4	Feed Gas Pressure	U
5	Feed Gas Temperature	U
6	PG Separator Liquid Level	U
7	PG Separator Pressure Difference	U
8	Pressure Control in PG Separator	U
9	PG Temperature in Absorption Column	U
10	Adsorption Column #1 Liquid Level	U
11	PG Pressure in Absorption Column	U
12	Adsorption Column #2 Liquid Level	U
13	Middle Absorption Column Temperature	U
14	Adsorption Column #3 Liquid Level	U
15	PG Pressure at the Top of Absorption Column	U
16	Liquid Ammonia Temperature in Absorption Column	U
17	PG Temperature at the Top of Absorption Column	U
18	Regenerator Liquid Level	U
19	PG Temperature in PG Separator	X

The process variables are collected by on-line sensors, so 2,100 samples, including normal and abnormal samples, are extracted from the DCS database of a real ammonia synthesis process. Fig. 4 (a) shows the behavior of the process variables. The occurrence of fault disturbance can be inferred from the feed gas flow rate (Fig. 4 (b)). The fault disturbances occur at time points 677 and 1,895, respectively.



(a) (b)

Fig. 4. The behavior of the (a) process variables and (b) the feed gas flow rate (PV1)

Fig. 5 shows the behavior of the collected process output variable. In Fig. 5, the red horizon dashed line represents the desired specification of PG temperature in PG separator while both black horizontal dashed lines are the upper and lower bounds that define the normal condition. Because of the large scale and slow dynamic transition nature of the ammonia synthesis process, the visualization of the impact of faults on process outputs emerges at time point 780 and 1,951 although the faults occur at time points 677 and 1,895. The sample points of 780-1,350 and 1951-2,070 are significantly out of specification. As the testing data start from the 1800th time point, the purpose of multi-step prediction is to observe the impact of faults on quality variables in advance (before 1,951 time points).

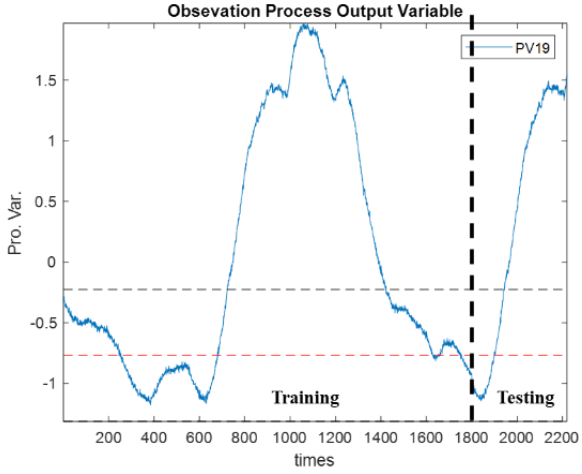


Fig. 5. The behavior of collected process output (the industrial case)

To realize early warning, both normal and abnormal situations must be included in the training set for the model to learn the process properties. About 1,800 process samples are used to arrange training and validation window data. Among the training and validation window dataset, 1,586 window data are used as a training dataset while 200 window data are used as a validation dataset. The rest of the data are used to form the testing window dataset to access the performance of the multi-step prediction model. The network structure of Multi-NSSM is designed as follows:

- The forward-backward RNN consists of only 1 layer with 5 neurons.
  $q_{\emptyset}(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}), p_{\theta}(\mathbf{Z}_{P_t}|\mathbf{U}_{P_t},\mathbf{X}_{P_t}))$
- The sequence-to-sequence RNN encoder-decoder network consists of only 1 layer with 5 neurons.
  $\left(q_{\emptyset}(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t}), p_{\theta}(\mathbf{Z}_{F_t}|\mathbf{Z}_{P_t})\right)$

Each DNN consists of 3 fully connected layers and each layer has 5 neurons. A hyperbolic tangent is used as the activation function for each layer. $\left(p_\theta\left(\mathbf{X}_{F_t} \mid \mathbf{Z}_{F_t}\right)\right)$

The initial cell state of the RNN network in the prediction network is assumed to start from zero. The window size of the data is 10 and the prediction horizon is set to be 5. The optimizer for Multi-NSSM is the Adam optimizer with a learning rate of 0.001 in TensorFlow. Cross validation and early stopping are used to prevent model overfitting.

To show the multi-step predictive ability of the proposed Multi-NSSM, multi-step RNNs with multi-input-multi-output (MIMO), iterative (Iter) and direct (Direc) strategies are included for comparisons (Table 2). To quantitatively assess the prediction performance, the R^2 performance index is considered using noise-free testing data. The R^2 formulations are given as:

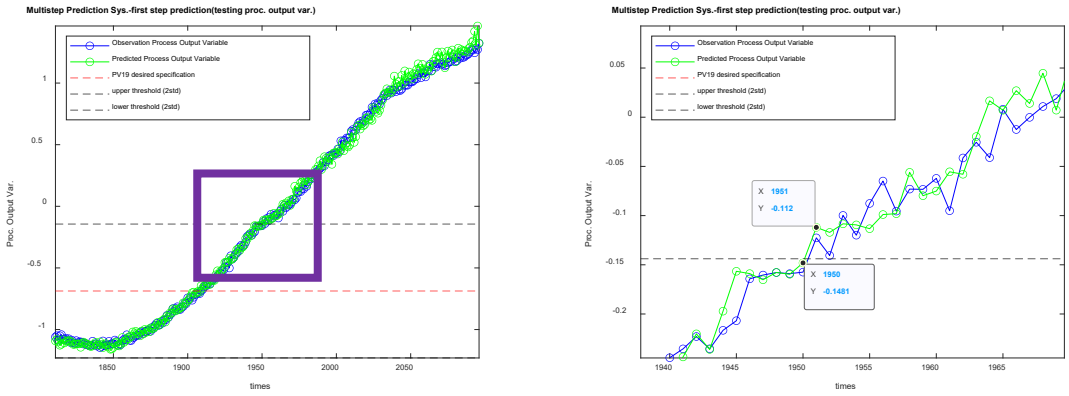
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{n=1}^{N}(x_n - \hat{x}_n}{\sum_{n=1}^{N}(x_n - \bar{x})^2} \tag{16}$$

where x_n is noise-free process output, $\hat{x}_n$ is the reconstructed process output of the model, and N is the total number of testing data. If the predicted values are close to the noise-free process output data, the R^2 would approximate 1; otherwise, the R^2 score would be away from 1. Hence, the R^2 of the comparison models are shown in Table 2.

Table 2. The R^2 of the first five-step ahead predictions in Multi-RNN-MIMO, Multi-RNN-Iter, Multi-RNN-Direc and Multi-NSSM models

R^2	Step 1	Step 2	Step 3	Step 4	Step 5
Multi-RNN (MIMO)	0.86	0.856	0.852	0.85	0.847
Multi-RNN (Iter)	0.98	0.93	0.82	0.73	0.66
Multi-RNN (Direc)	0.94	0.98	0.92	0.95	0.88
Multi-NSSM	0.997	0.995	0.995	0.994	0.993

The comparison results in Table 2 show that the R2 of Multi-NSSM are closer to 1 than those of Multi-RNNs. With either the prediction of the first step or the fifth step applied to the testing data, Multi-NSSM predicts better than Multi-RNNs because the process is usually contaminated with uncertain noise. RNN, which is a deterministic model, cannot describe the uncertainty of the process while Multi-NSSM, which is a probability model, can well describe the process uncertainty. The deterministic model can be seen as a special case of a probabilistic model because it describes only one condition in the distribution. On the other hand, if the same process input variable and hidden state variables are inputted to the RNN model, RNN will always obtain the same output. Nevertheless, the condition of the process is susceptible to uncertainties in reality, such as environmental changes, variation in flow rates, etc. Based on the influence of uncertainties, the output of the process will not always be the same. Thus, using distribution to describe the output of the process is more reasonable and this is the benefit of constructing a probabilistic model. After the multistep prediction model (Multi-NSSM) is completely trained, the multistep prediction system on the process output is ready to perform early warning. The multistep predicted process output data from Multi-NSSM in the 1st step are shown on Fig. 6(a).

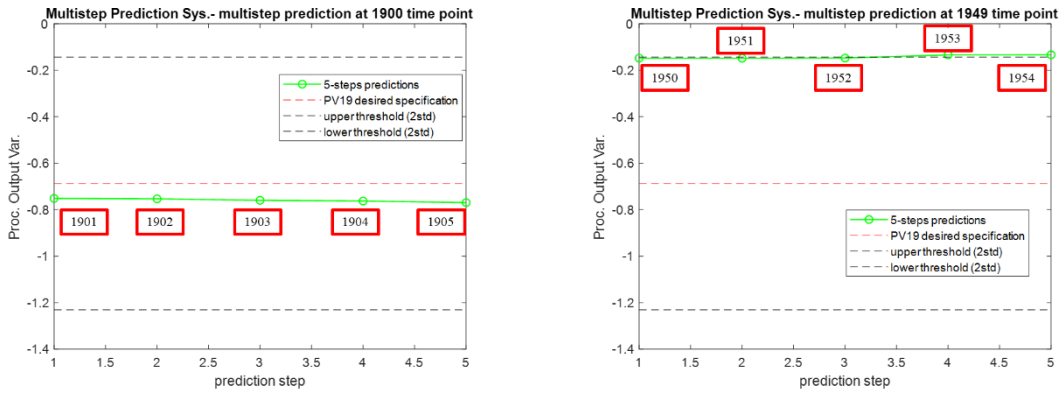


(a) (b)

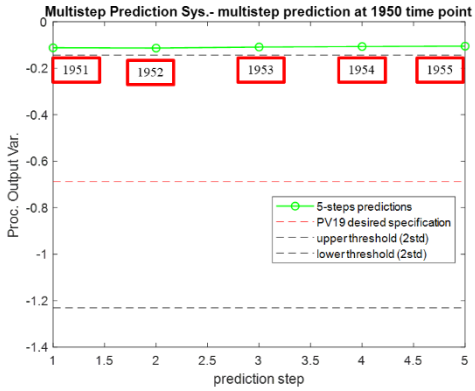
Fig. 6. The 1-step predictions of process variables of (a) the Multi-NSSM and (b) the zoomed part of the dark purple box in (a)

Because of the slow dynamic transition nature of the large-scale process, the occurrence of the fault often does not have an instantaneous effect on the process outputs. The proposed multistep prediction system can identify the influence of the fault in advance through the multistep predictions, thereby accomplishing accurate monitoring result and even early warning. As mentioned previously, the fault occurs at 1,890th time point and the impact on the process output can only be observed at the 1,951th time point. In Fig. 6, the 1st step prediction detects the fault at the 1,950th time point. Moreover, the 5-step ahead prediction of process variable at the 1900th time point is shown in Fig. 7(a). As the contribution of the disturbance are too small to affect the process and slow dynamic transition of the process, the disturbances gradually propagate to influence the process instead of affecting the process immediately. Although the 1-step prediction of the process output at 1,949th time point (Fig. 6 (b)) and the first prediction in the 5-step ahead prediction 1,949th time point (Fig. 7 (b)) maintains in the normal operating condition, the predicted future process outputs slowly deviate from the normal region. The growth of the previous fault disturbance has given an impact on the process.

As for the 1-step prediction at 1,950th time point (Fig. 6 (b)) and the first prediction in the 5-step ahead prediction at 1,950th time point (Fig. 7 (c)) indicates that the process is totally influenced by the fault disturbance. Thus, the multi-step predictions from the 1,950th time point onwards deviate from the normal region.



(a) (b)



(c)

Fig. 7. The trends of the 5-steps ahead predictions at the (a) 1,900th, (b)1,949th, and (c)1,950th time points

4. Conclusion

Because of the large scale and slow dynamic transition of chemical processes, the occurrence of fault disturbance may not influence the process instantaneously until a few time points later as the disturbance must propagate through the process. The impact of fault disturbances cannot be determined by measuring or predicting the process outputs at the current time. It will be too late to be detected after all the products are inspected. The fault disturbances already have given a huge impact on process safety and profits. Therefore, a multistep ahead prediction model is essential for estimating the impact of disturbances by predicting the future trend of the process outputs. Moreover, there are usually uncertain disturbances in chemical processes. A probabilistic model is good at describing the uncertain nature. With the aforementioned problems, the proposed method is proposed to mitigate those problems. Multi-NSSM is constructed using the measured process input and output variables to obtain its predictive ability for future process output predictions. With the multistep ahead prediction process outputs from Multi-NSSM, multi-step ahead process monitoring can be done for early warning. Meanwhile, the temperature reflection in the PVC drying process is slow and consists of process dynamics. From the result and discussion section, it is found that the proposed Multi-NSSM can identify the fault disturbance in advance to achieve early warning, although the fault disturbances do not influence the process instantaneously. The comparison results in the industrial cases show that Multi-NSSM developed as a probabilistic model can well describe the uncertainty of the process while Multi-RNN developed as a deterministic model cannot. This can be proof by the fault detection result in Fig. 7. The proposed multi-step ahead prediction model can only describe the learned fault that happened before. The meta-learning strategy can be used to merge with the multi-step ahead prediction model, the information from other faults can be used to transfer knowledge to enhance the information for new disturbances. These may be breakthrough points to solve the above problem in future research direction.

Acknowledgment

The authors would like to gratefully acknowledge Ministry of Science and Techonology, Taiwan, R.O.C. (MOST 109-2221-E-033-013-MY3). Also, they would like to thank China General Plastic Corporation, Taiwan, which provides the industrial data for the case study.

Declarations

Author contribution. Yi Shan Lee: Conceptualization, Methodology, Data curation, Writing – review and editing.
Sai Kit Ooi: Conceptualization, Methodology, Visualization, Writing – original draft.
Junghui Chen: Conceptualization, Methodology, Writing – review and editing, Supervision, Funding acquisition.

Funding statement. This work was supported by the Ministry of Science and Techonology, Taiwan, R.O.C. under Grant MOST 109-2221-E-033-013-MY3.

Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] M. H. Kaspar and W. Harmon Ray, "Dynamic PLS modelling for process control," *Chem. Eng. Sci.*, vol. 48, no. 20, pp. 3447–3461, Oct. 1993, doi: 10.1016/0009-2509(93)85001-6.

[2] Y. Dong and S. J. Qin, "Regression on dynamic PLS structures for supervised learning of dynamic data," *J. Process Control*, vol. 68, pp. 64–72, 2018, doi: 10.1016/j.jprocont.2018.04.006.

[3] M. Wozniak, J. Silka, M. Wieczorek, and M. Alrashoud, "Recurrent neural network model for IoT and networking malware threat detection," *IEEE Trans. Ind. Informatics*, vol. 17, no. 8, pp. 5583–5594, 2021, doi: 10.1109/TII.2020.3021689.

[4] M. Wozniak, M. Wieczorek, J. Silka, and D. Polap, "Body pose prediction based on motion sensor data and recurrent neural network," *IEEE Trans. Ind. Informatics*, vol. 17, no. 3, pp. 2101–2111, 2021, doi: 10.1109/TII.2020.3015934.

[5] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM – A tutorial into Long Short-Term Memory Recurrent Neural Networks," *arXiv*, pp. 1–42, 2019. [Online]. Available at: https://arxiv.org/abs/1909.09586.

[6] Y. Han, C. Fan, M. Xu, Z. Geng, and Y. Zhong, "Production capacity analysis and energy saving of complex chemical processes using LSTM based on attention mechanism," *Appl. Therm. Eng.*, vol. 160, no. July 2018, p. 114072, 2019, doi: 10.1016/j.applthermaleng.2019.114072.

[7] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowledge-Based Syst.*, vol. 181, p. 104785, 2019, doi: 10.1016/j.knosys.2019.05.028.

[8] Q. Sun and Z. Ge, "Probabilistic Sequential Network for Deep Learning of Complex Process Data and Soft Sensor Application," *IEEE Trans. Ind. Informatics*, vol. 15, no. 5, pp. 2700–2709, 2019, doi: 10.1109/TII.2018.2869899.

[9] X. Yin, Z. Niu, Z. He, Z. Li, and D. hee Lee, "Ensemble deep learning based semi-supervised soft sensor modeling method and its application on quality prediction for coal preparation process," *Adv. Eng. Informatics*, vol. 46, no. June, p. 101136, 2020, doi: 10.1016/j.aei.2020.101136.

[10] L. Feng, C. Zhao, and Y. Sun, "Dual attention-based encoder-decoder: A customized sequence-to-sequence learning for soft sensor development," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–12, 2020, doi: 10.1109/tnnls.2020.3015929.

[11] J. M. P. Menezes and G. A. Barreto, "Long-term time series prediction with the NARX network: An empirical evaluation," *Neurocomputing*, vol. 71, no. 16–18, pp. 3335–3343, 2008, doi: 10.1016/j.neucom.2008.01.030.

[12] A. Grigorievskiy, Y. Miche, A. M. Ventelä, E. Séverin, and A. Lendasse, "Long-term time series prediction using OP-ELM," *Neural Networks*, vol. 51, pp. 50–56, 2014, doi: 10.1016/j.neunet.2013.12.002.

[13] R. M. Adnan, Z. Liang, S. Trajkovic, M. Zounemat-Kermani, B. Li, and O. Kisi, "Daily streamflow prediction using optimally pruned extreme learning machine," *J. Hydrol.*, vol. 577, no. July, p. 123981, 2019, doi: 10.1016/j.jhydrol.2019.123981.

[14] H. Li, Y. He, H. Yang, Y. Wei, S. Li, and J. Xu, "Rainfall prediction using optimally pruned extreme learning machines," *Nat. Hazards*, vol. 108, no. 1, pp. 799–817, 2021, doi: 10.1007/s11069-021-04706-9.

[15] G. Shi, C. Qin, J. Tao, and C. Liu, "A VMD-EWT-LSTM-based multi-step prediction approach for shield tunneling machine cutterhead torque," *Knowledge-Based Syst.*, vol. 228, p. 107213, 2021, doi: 10.1016/j.knosys.2021.107213.

[16] Y. Duan, Y. Lv, and F. Y. Wang, "Travel time prediction with LSTM neural network," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 1053–1058, 2016, doi: 10.1109/ITSC.2016.7795686.

[17] Z. Karevan and J. A. K. Suykens, "Transductive LSTM for time-series prediction: An application to weather forecasting," *Neural Networks*, vol. 125, pp. 1–9, 2020, doi: 10.1016/j.neunet.2019.12.030.

[18] A. Zhang, X. Zhao, and L. Wang, "CNN and LSTM based Encoder-Decoder for Anomaly Detection in Multivariate Time Series," *IEEE Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2021*, vol. 5, pp. 571–575, 2021, doi: 10.1109/ITNEC52019.2021.9587207.

[19] M. Sibtain *et al.*, "Multifaceted irradiance prediction by exploiting hybrid decomposition-entropy-Spatiotemporal attention based Sequence2Sequence models," *Renew. Energy*, vol. 196, pp. 648–682, 2022, doi: 10.1016/j.renene.2022.07.041.

[20] W. Deng, Y. Li, K. Huang, D. Wu, C. Yang, and W. Gui, "LSTMED: An uneven dynamic process monitoring method based on LSTM and Autoencoder neural network," *Neural Networks*, vol. 158, pp. 30–41, 2023, doi: 10.1016/j.neunet.2022.11.001.

[21] F. Yan, C. Yang, and X. Zhang, "DSTED: A Denoising Spatial-Temporal Encoder-Decoder Framework for Multistep Prediction of Burn-Through Point in Sintering Process," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10735–10744, 2022, doi: 10.1109/TIE.2022.3151960.

[22] Z. Li, S. Riaz, M. Waqas, and M. Batool, "An Enhanced Gated Recurrent Unit-Based Adaptive Fault Diagnosis of Rotating Machinery," *Shock Vib.*, vol. 2022, p. 13, 2022, doi: 10.1155/2022/4648311.

[23] R. Guo and H. Liu, "A Hybrid Mechanism- And Data-Driven Soft Sensor Based on the Generative Adversarial Network and Gated Recurrent Unit," *IEEE Sens. J.*, vol. 21, no. 22, pp. 25901–25911, 2021, doi: 10.1109/JSEN.2021.3117981.

[24] T. Limouni, R. Yaagoubi, K. Bouziane, K. Guissi, and E. H. Baali, "Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model," *Renew. Energy*, vol. 205, no. December 2022, pp. 1010–1024, 2023, doi: 10.1016/j.renene.2023.01.118.

[25] H. Zhao, H. Liu, J. Xu, C. Guo, and W. Deng, "Research on a fault diagnosis method of rolling bearings using variation mode decomposition and deep belief network," *J. Mech. Sci. Technol.*, vol. 33, no. 9, pp. 4165–4172, 2019, doi: 10.1007/s12206-019-0811-2.

[26] Z. Geng, X. Duan, Y. Han, F. Liu, and W. Xu, "Novel variation mode decomposition integrated adaptive sparse principal component analysis and it application in fault diagnosis," *ISA Trans.*, vol. 128, pp. 21–31, 2022, doi: 10.1016/j.isatra.2021.11.002.

[27] S. Lin, R. Clark, R. Birke, and S. Sch, "Anomaly detection for time series using VAE-LSTM hybrid model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), Barcelona*, 2020, pp. 4322–4326, doi: 10.1109/ICASSP40776.2020.9053558.

[28] Q. Quan, Z. Hao, H. Xifeng, and L. Jingchun, "Research on water temperature prediction based on improved support vector regression," *Neural Comput. Appl.*, vol. 34, no. 11, pp. 8501–8510, 2022, doi: 10.1007/s00521-020-04836-4.

[29] G. F. Fan, M. Yu, S. Q. Dong, Y. H. Yeh, and W. C. Hong, "Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling," *Util. Policy*, vol. 73, no. September, p. 101294, 2021, doi: 10.1016/j.jup.2021.101294.

[30] K. Zhang, H. Cao, J. Thé, and H. Yu, "A hybrid model for multi-step coal price forecasting using decomposition technique and deep learning algorithms," *Appl. Energy*, vol. 306, no. October 2021, 2022, doi: 10.1016/j.apenergy.2021.118011.