IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 August 2022, accepted 12 September 2022, date of publication 15 September 2022,
date of current version 22 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206950

RESEARCH ARTICLE

SSAE and GRU Based Joint Modeling for Nonlinear Distributed Parameter Systems

LING AI[1], JUNZHE GAN[1], XIANJIE FENG[2], AND XUEQIN CHEN[3]
[1]Department of Automation, Harbin University of Science and Technology, Harbin 150086, China
[2]Department of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150086, China
[3]Research Center of Satellite Technology, Harbin Institute of Technology, Harbin 150040, China

Corresponding author: Ling Ai (ailing@hrbust.edu.cn)

This work was supported in part by National Natural Science Foundation of China under Grant 61673141 and Grant 61806060, in part by the Natural Science Foundation of Heilongjiang Province of China under Grant LH2019F024, in part by the Harbin Science and Technology Bureau of China under Grant 2017RAQXJ006, and in part by the China Scholarship Council under Grant 201608230319.

ABSTRACT The modeling and control issues for distributed parameter systems (DPSs) have received a great deal of attention. Because linear model order reduction (MOR) methods may ignore the nonlinear dynamics and lose some details, it is difficult to describe DPS accurately by common modeling methods. To effectively model such systems, a sparse stacked auto-encoder and gated recurrent unit (SSAE-GRU) model is proposed in this paper. Under the time/space separation theory, it is the mainstream way to perform MOR and identification of time series respectively. In the SSAE-GRU model, this practice is still adhered to but joint learning is recommended. SSAE can be used as an excellent MOR technique. A sparse activation strategy that is introduced makes its model space simple and easy to train. GRU has the ability to represent such complex temporal properties because the information stored by previous neurons can be transmitted to the current moment selectively. The joint training method allows them to be responsible and consider the connection between adjacent moments and spatial energy transfer overall. Then, we use L2 regularization in back-propagation to reduce the difficulty of model optimization and prevent overfitting. The modeling scheme is simulated on two typical chemical thermal processes. This article demonstrates the effectiveness of the proposed method as well as outstanding performance compared to existing methods.

INDEX TERMS Distributed parameter systems, model order reduction, sparse stacked auto-encoder, gated recurrent unit, joint learning.

I. INTRODUCTION

A. BACKGROUND

Intelligent manufacturing is the combination of advanced sensing, detection, control, and process optimization technologies and practices that fuse information and the manufacturing environment to enable precise management of energy, production efficiency, and costs in factories [1]. The snap curing oven during chip packaging [2], the thermal monitoring of lithium-ion battery charge and discharge experiments [3], and the thermal management of chemical engineering reactors are all topics of intelligent manufacturing research. They have infinite-dimensional properties described by partial differential equations (PDEs). These industrial processes with complex spatiotemporal distributed nature are commonly referred to as distributed parameter systems (DPSs). Modeling and controlling issues for DPSs are a challenging task [4].

Model order reduction (MOR) techniques are essential for modeling DPSs [5]. It transforms the origin DPSs described by infinite-dimensional partial differential equations (PDEs) into finite-dimensional lump parameter systems (LPSs) described by ordinary differential equations (ODEs). Although spatiotemporal discretization methods such as the finite difference technique and the finite element method (FEM) can be used for general DPSs with irregular boundaries, the high reduction orders impose a huge computing load on subsequent control applications. We are accustomed to simplifying the systems before controller design. The low-dimensional representation of parabolic systems can be produced using spatiotemporal decomposition theory [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei.

B. IITERATURE REVIEW

Due to the lack of in-depth information on the physicochemical backdrop, data-driven approaches are frequently used in the modeling of unknown DPSs. The spatiotemporal dataset can be gathered using a large number of spatial sensors, which are subsequently processed using data-based linear and nonlinear MOR algorithms to yield low-dimensional time series [7]. Principal component analysis (PCA) [8], also called Karhunen-Loeve decomposition (KLD) [9] or proper orthogonal decomposition (POD) [9], is one of the most famous linear algorithms applied to MOR for DPSs. PCA is a global linear projection method [10]. It uses a linear approximation for the nonlinear problem that would not ensure the minor components never contain the important information [6]. Since linear MOR algorithms cannot preserve the nonlinear spatial structure of the complex system, a range of measures have been created to enhance MOR performance. By splitting the original dataset into tractable subsets, a novel multimode spatiotemporal modeling technique based on the locally weighted PCA (LW-PCA) method is created for large-scale highly nonlinear DPSs with parameter fluctuations [11]. By incorporating information entropy, adaptive PCA adjusts the weight matrixes of reconstructing error [12]. Both of them enhance boosting linear PCA accuracy to a certain extent. In addition, nonlinear approaches such as isometric mapping (ISOMAP) [13] and kernel support vector machine (SVM) [14] have been employed for DPSs. Nonlinear MOR techniques including locally linear embedding (LLE) [15] have greatly enriched the practice of DPS modeling issues. These algorithms are shallow learning networks with a single hidden layer structural model. As we all know, shallow networks are prone to fall into local optimum and have poor generalization ability.

In recent years, multi-layer deep networks are more efficient at extracting features from high-dimensional data [16]. A MOR framework for DPSs was designed to utilize a deep auto-encoder (AE) embedded in Restricted Boltzmann Machine (RBM) with a layer-wise pre-trained learning strategy [17]. A multi-layer AE architecture with direct training for DPSs has been developed [18]. Although the deep networks-based MOR techniques mentioned above have shown significant promise in improving the reduced models' performance, there are still several specific constraints such as being cumbersome or difficult to converge. These deep network-based DPS modeling techniques rely on multiple training epochs on the dataset to increase the modeling capacity, and there is still room for improvement.

Additionally, numerous identification techniques, such as extreme learning machine (ELM) [19], least-squares support vector machine (LS-SVM) [20], have been applied to the related low-dimensional time-series obtained by MOR. A Dual ELM model is developed for the two nonlinearities embedded in industrial thermal processes [21]. A spatiotemporal LS-SVM model is designed to compensate for modeling errors due to truncation and unknown nonlinear dynamics [22]. A modified High-Order SVD that takes into account the interaction across several spatial modes is applied to model DPSs [23]. A fast incremental learning-based modeling approach for thermal process modeling of lithium batteries is developed [24]. A finite sensing optimization technique with recursive temperature field estimation for pouch cells is devised [25]. For time series identification in wireless sensor networks, a distributed spatiotemporal Volterra model (DS-Volterra) with enhanced Wiener is used [26], [27], [28]. A reduced model via multilayer perceptron and long short term memory (MLP-LSTM) is proposed to approximate the DPS situation of two coupled nonlinear dynamics [29].

We note that some new networks, for instance, Alexnet [30] with rectified linear units (ReLU) [31] achieved a low test error rate of image classification. ReLU with L1 regularization trick [32] has been proved that sparsity operating in a deep neural network is more biologically plausible. Gated recurrent unit (GRU) [33], which is an elaborate recurrent neural network (RNN), is designed to model time series. It has comparable accuracy with the long short term memory (LSTM) [34], and meanwhile, the parameters that need to be trained are less by one-fourth.

In this paper, a novel SSAE-GRU-based modeling approach is presented for the nonlinear DPSs. The intrinsic features are extracted using sparse stacked auto-encode (SSAE) approach with the sparse activation functions. The SSAE can fit networks without pre-training. Considering practicality and ease of implementation, the sparsity constraints by L2 penalty and exponential linear unit (ELU) activation function are applied. Then, the proper evolution law of low-dimensional representation and control signal are establish by GRUs. The capacity of time-series prediction to generalize has been aided through regularization. Finally, the proposed model adopts a joint learning approach. Unlike existing methods, the proposed method only requires optimization for one objective function because we are most interested in high-dimensional reconstruction. It has the potential to lower modeling errors. The main contributions and novelty of this paper are summarized as follows:

1) A sparse form of stacked auto-encoder is introduced to resolve the MOR issue of DPSs. Sparse representation is closer to the system reality, which alleviates computationally intensive, makes the network easier to train, and ultimately improves the performance of modeling.

2) Considering the features of DPSs are related between the time and space dimensions, a joint learning approach is adopted. MOR and time series prediction are performed in one step for gathering a model with higher accuracy.

3) Simulations on two representative chemical thermal processes verify the effectiveness of the proposed method.

The rest of the paper is structured as follows. The problem description is in Sections II. In Section III, the SSAE-GRU algorithm is presented. Section IV gives the experiment result of two typical chemical thermal processes to confirm the effectiveness of the proposed method. A summary is given in Section V.

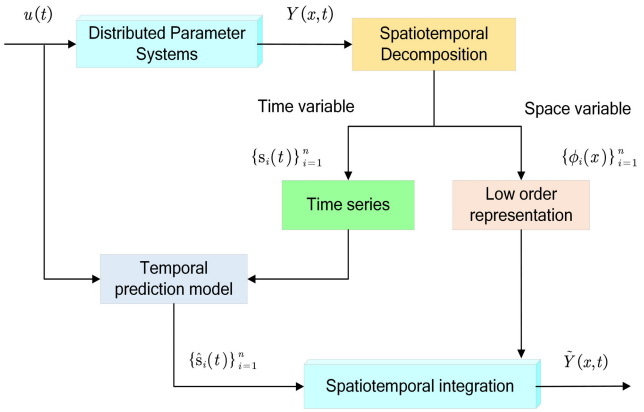


FIGURE 1. Schematic representation of the spatiotemporal modeling.

II. RELATED WORK

A. MATHEMATICAL FOUNDATION OF DPS

Consider a general DPS can be described by the following nonlinear PDEs:

$$\frac{\partial Y(x,t)}{\partial t} = \mathscr{L}\Big(Y(x,t), \frac{\partial Y(x,t)}{\partial x}, \ldots, \frac{\partial^n Y(x,t)}{\partial^n x}\Big) + \big(u(t)\big) \tag{1}$$

subject to the boundary condition:

$$\mathscr{Q}\Big(Y(x,t), \frac{\partial Y(x,t)}{\partial x}, \ldots, \frac{\partial^{n-1} Y(x,t)}{\partial^{n-1} x}\Big)\Big|_{x=x_0, x=x_n} = 0 \tag{2}$$

and the initial condition:

$$Y(x,0) = Y_0(x). \tag{3}$$

where, t is the time variable, $x \in \Omega$ is the spatial variable, and Ω is the spatial domain, $Y(x,t)$ is the controlled output variable, $u(t)$ is the control input variable, $\mathscr{L}$, , and $\mathscr{Q}$ are the continuous differentiable functions in Hilbert space $\mathscr{H}$. It includes two nonlinear time law: a block is from system $Y(\cdot,t)$ and the other is from $u(t)$. The basic steps of DPS spatiotemporal modeling are shown in Fig. 1.

System (1) is applicable to model a variety of physical and biochemical processes, such as catalytic reaction rods, steel casting, and the tubular reactor. To obtain accurate information about such a system, a sufficient number of sensors need to be placed along with the spatial location. Only a small number of actuators are allowed to be mounted for observing the state in actual physical conditions. The input-output datasets are obtained from the actual production process under random signal excitation. The modeling algorithm is developed in two stages: the output $\{Y(x_m,t_n)\}_{m=1,n=1}^{M,N}$ excited by the input $\{u(t_n)\}_{n=1}^{N}$ is used by the SSAE to reduce the dimensionality of the approximate model. M, N, and K are the number of the sensors, the sampling duration, and the actuators, respectively.

B. SPATIOTEMPORAL DECOMPOSITION THEORY

According to spatiotemporal decomposition theory [5], the variables of DPSs which is controlled by PDE can be expanded as:

$$Y(x,t) = \sum_{i=1}^{n} s_i(t) \cdot \phi_i(x) \tag{4}$$

The spatiotemporal variables can be decomposed into two parts: a set of low-dimension representations $\phi_i(x)$ and a temporal model $s_i(t)$. To reduce the $Y(x,t)$ to $s_i(t)$, a number of MOR techniques are listed in literature review.

$$s_i(t) = T_e\left(Y(x,t)\right) \tag{5}$$

where T_e denotes the dimensionality reduction function.

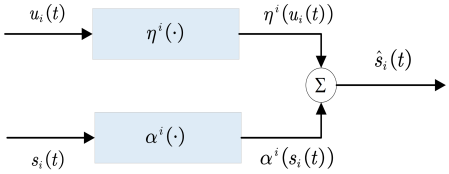


FIGURE 2. Schematic representation of the low-order time evolution law.

C. ESTABLISH LOW-ORDER TEMPORAL SERIES

It is critical to create an appropriate representation and identify the corresponding time series. Some scholars have scientifically proven that time series is decomposed into two nonlinear units with different regularities $s_i(t)$ [21], [35]. This law has depicted in Fig. 2.

$$\hat{s}_i(t) = \alpha^i\left(s_i(t)\right) + \eta^i\left(u_i(t)\right) \tag{6}$$

where the $\alpha^i(\cdot)$ and $\eta^i(\cdot)$ are nonlinear modules. $\hat{s}_i(t)$ is temporal prediction value.

D. SPATIOTEMPORAL INTEGRATION

The system prediction outputs can be acquired by spatiotemporal integration.

$$\tilde{Y}(x,t) = T_d(\hat{s}_i(t)) \tag{7}$$

where $\hat{s}_i(t), T_d, \tilde{Y}(x,t)$ are forecasted temporal series, integral function, spatiotemporal integration predictions, respectively.

Though conventional modeling methods have an acceptable accuracy on nonlinear processes, there are still some practices required to improve, which can be summarized as follow:

1) DPSs are nonlinear and spatiotemporal-varying. Linear projections ignore nonlinear variation among the data. Shallow networks are deficient in learning ability, decreasing their effectiveness in nonlinear DPS modeling.

2) The randomness of system inputs makes it difficult to identify temporal dynamics.

3) Considering the ease of implementation and practice, some reported methods have high calculation costs thus need to make a balance between accuracy and time-consuming.

III. SSAE-GRU SPATIOTEMPORAL MODEL

A. THE FRAMEWORK OF THE PROPOSED MODEL

To solve the above problem, the framework of the proposed model will be described in detail as follow:

1) The SSAE technique is applied for extracting low-dimensional features and forming an essential representation to characterize high-dimensional PDEs. Details of the SSAE technique can be found in Section III.B.

2) GRU which is multivariable time series forecasting algorithm is set up to build the temporal dynamics and deal with nonlinearities. Details of the GRU are presented in Section III.C.

3) According to spatiotemporal reconstruction, the high dimensional temperature distribution model can be constituted. Details of spatiotemporal reconstruction and joint learning are presented in Section III.D.

B. SSAE MOR TECHNIQUE

By training a multilayer neural network with a small bottleneck layer to reconstruct high-dimensional input vectors, high-dimensional data can be transformed into low-dimensional codes. The essential features of the system are acquired from the bottleneck layer by stacked auto-encoder (SAE).

SAE has the composition of an encoding function T_e and a decoding function T_d. The encoder is created using a multiple-layer neural network. At each discretized time step, the vector $\mathrm{Y} = \left[\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_p\right]^T \in R^p$ represents the p input. k is the encoder network layer. The encoder projects Y from the input layer to low order representation $y = [y_1, y_2, \ldots, y_c]^T \in R^c$. Decoder function T_d has a symmetrical structure with T_e. Hence, the input-output of the SAE can be expressed as follows:

$$\begin{aligned} y(t) &= T_e(\mathrm{Y}(\cdot, t)) \\ &= \varphi(\mathrm{W}_k \ldots \varphi(\mathrm{W}_1 \mathrm{Y}(\cdot, t) + b_1) \ldots + b_k) \quad (8) \\ \tilde{Y}(\cdot, t) &= T_d(y(t)) \\ &= \varphi\left(\mathrm{W}_1^T \ldots \varphi\left(\mathrm{W}_k^T y(t) + b_k\right) \ldots + b_1\right) \quad (9) \end{aligned}$$

where $\varphi(\cdot)$ are nonlinear activation functions that act element-wise on its inputs, which have many different forms such as step function, sigmoid and tanh, etc. W are $p \times c$ matrixs. $b \in R^c$ are the bias vectors. The superscript T means matrix transposition. Fig. 3 is a normal, flexible and adjustable architecture of SAE. Though SAE can achieve high precision since Hinton trained networks by layer-wise pre-training. We may discover interesting structures, by imposing other constraints and tricks on the network.

A neuron is defined to be "active" if its output value is close to 1, or to be "inactive" if its output value is close to 0. The sparse function in this study constraints the neurons to be inactive for most of the sampling time. A Sparsity constraint imposes on the hidden units, mainly by changing the activation strategy.

ELU which is chosen as the activation function has the characteristics of unilateral inhibition and fast convergence.

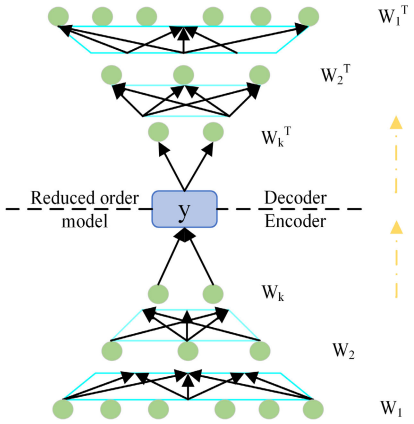


FIGURE 3. Schematic representation of the SAE.

It combines the advantages of sigmoid and ReLU. The left side has soft saturation like a sigmoid and the right side has no saturation like ReLU. Thus the neurons involved in a calculation have limited activation. ELU has negative values which allow it to push mean unit activation closer to zero. It achieves sparsity in a low-cost way to speed up learning.

$$\mathrm{ELU}(x) = \begin{cases} (e^x - 1) & x < 0 \\ x & x \geq 0 \end{cases} \quad (10)$$

Our model is activated by ELU. It endows network with the ability to fit nonlinearity. ELU also allows the network to limit the activity of neurons during error back-propagation so that the gradient does not explode or vanish.

C. GRU SERIES FORECASTING MODEL

RNN can be used in many works in natural language processing (NLP) successfully. Long Short Term Memory (LSTM) is designed to overcome the limitations of long-term dependency. However, LSTMs have a rather complex design with three multiplicative gates, which might impair their efficient implementation. An attempt to simplify LSTMs has recently led to Gated Recurrent Units (GRUs), which are based on just two multiplicative gates. Just a while ago, the Minimal RNN is suggested. Its accuracy is not as good as LSTM and GRU though it is simple and trained easily. The GRU has two control gates, each of them is activated by a sigmoid. The gates receive a weighted sum of current input x_t and previous output h_{t-1} as the total input. The update gate z_t and reset gate r_t can be expressed as:

$$z_t = \sigma(W_{iz} \cdot x_t + b_{iz} + W_{hz} \cdot h_{t-1} + b_{hz}) \quad (11)$$

$$r_t = \sigma(W_{ir} \cdot x_t + b_{ir} + W_{hr} \cdot h_{t-1} + b_{hr}) \quad (12)$$

where W_{iz} and W_{hz} are the weight matrixes of update gate; b_{iz} and b_{hz} are the biases of update gate; W_{ir} and W_{hr} are the weight matrixes of reset gate; b_{ir} and b_{hr} are the biases of update gate. z_t controls the amount of information needs to be forgotten from the h_{t-1}. r_t controls the amount of information needs to be reserved from the h_{t-1}.

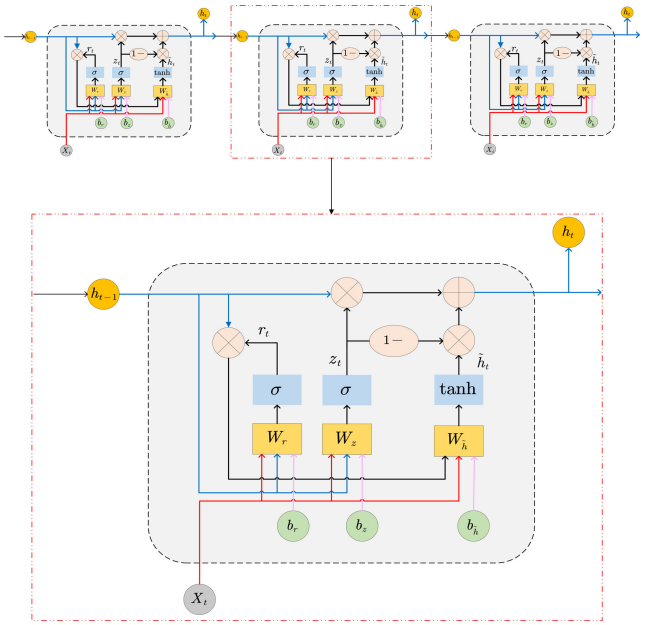


FIGURE 4. Schematic representation of the GRU.

GRU also has two states, which are called candidate hidden state n_t and current output state h_t, respectively.

$$n_t = \tanh\left(W_{in}x_t + b_{in} + r_t \odot \left(W_{hn}h_{t-1} + b_{hn}\right)\right) \quad (13)$$

$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{t-1} \quad (14)$$

Similarly, W and b are the weight and bias of the candidate hidden state. $\odot$ is the Hadamard product. σ is sigmoid function. In conclusion, h_t determines the final output according to the information of the gates and candidate state. Under the problem of time series forecasting, the GRU process can be regarded as:

$$\hat{x}_t = h_t = \varsigma\left(x_t\right) \quad (15)$$

where $\varsigma(\cdot)$ is GRU. x_t is current input. $\hat{x}_t$ is current output, namely, the input of next time. Fig. 4 illustrates the architecture of GRU.

The full operation of the GRU temporal model is listed as follows:

Step 1: Prepare the input data that comes from control inputs $u(t)$ and spatial low dimensional representation $y(t)$.

Step 2: Establish the relationship between the control input $u(t)$ and temporal prediction $\hat{y}(t)$ by GRU.

Step 3: Employ GRU to identify time dynamics between $y(t)$ and the prediction $\hat{y}(t)$.

Step 4: Build dual GRU time series model.

D. SPATIOTEMPORAL INTEGRATION AND JOINT LEARNING

Measured output data $Y(x,t)$ and random inputs $u(t)$ are taken for representations learning and model identification. First, SSAE compresses data collected by spatially distributed sensors as:

$$y = T_e\left(Y\right) \quad (16)$$

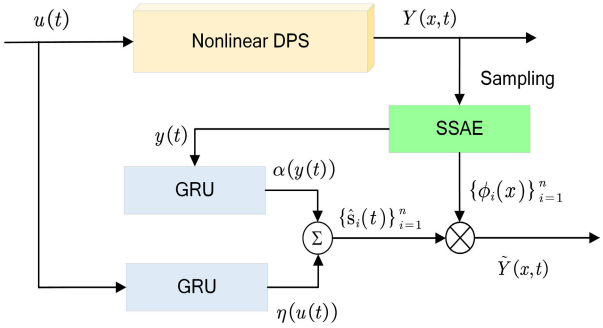


FIGURE 5. Schematic representation of the SSAE-GRU workflow.

Considering the timing rules of the external input and the system itself are completely different, we use two GRUs to learn separately. In the forward propagation of the GRU network, to simplify the mathematical description, the relationship refers to equation (6) between the future state $\hat{y}(t)$, its present state $y(t)$, and the systematic excitation $u(t)$ can be constructed as:

$$\hat{y}\left(t\right) = \alpha\left(y\left(t\right)\right) + \eta\left(u\left(t\right)\right) \quad (17)$$

Recombining the prediction of time variables with the spatial variables, high-level system predictions consistent with the original dimension can be derived as follow:

$$\tilde{Y} = T_d\left(\hat{y}\right) \quad (18)$$

where y is low dimensional representations of system. T_e, T_d are encoder, decoder function. Both α and η are temporal identification model. The overflow diagram of this work is shown in Fig. 5.

In this study, both SSAE and GRU are trained in a single stage. Two subtasks of MOR and series prediction they represent are integrated into one learning process. Both temporal and spatial variables are connected by a final loss function, which is more in line with the time/space coupling characteristics of DPS. To fight against overfitting, we add the L2 regularization term into the cost function. Regularization is also one of the sources of sparsity. Before modeling, data preprocessing is a necessary step. Here, a Min-Max Scaler method is applied to scale the data to a range of [0,1].

Based on the spatiotemporal data, considering modeling errors both in time and space, the objective function of the SSAE-GRU joint model is constructed as follows:

$$J(\theta; Y, \tilde{Y}) = \left\| \tilde{Y}(\theta) - Y \right\|_2 \quad (19)$$

where $J(\theta; Y, \tilde{Y})$ is 2-norm loss function which need to be optimized. The definition of Y and $\tilde{Y}$ is the same as before.

θ_t is the set that consists of all weights and biases which influence the error at present. The gradient back-propagation includes four parts. Details are presented as follow:

1) *Gradient of the objective function:* From (19), the gradient g_t of all the samples at the t-th time step is derived

as follow:

$$g_t = \nabla J_t(\theta_{t-1}) + \lambda\theta_{t-1} \tag{20}$$

where λ is the regularization hyperparameter which restrained the model fit.

2) *Calculate first moment vector:* According to (20), m_t is determined and is also controlled by decay coefficient β_1 as follow:

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1)\cdot g_t \tag{21}$$

The initial value of m_t is zero. $\beta_1 = 0.9$.

3) *Calculate second moment vector:* With respect to (20), V_t can be decided and decay coefficient β_2 also have an influence as follow:

$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2)\, g_t^2 \tag{22}$$

where v_t is initialized as zero. $\beta_2 = 0.999$.

4) *Update parameters:* As a result, the latest values is given by:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \frac{\zeta \cdot m_t}{\sqrt{v_t}} \\ &= \theta_t - \frac{\zeta \cdot (\beta_1 \cdot m_{t-1} + (1-\beta_1)\cdot g_t)}{\sqrt{\beta_2 \cdot v_{t-1} + (1-\beta_2)\, g_t^2}}\end{aligned} \tag{23}$$

Here learning rate $\zeta = 0.001$. g_t^2 indicates the element-wise square $g_t \odot g_t$. The above-mentioned hyperparameters set as constants are all referenced in [36] and [37].

E. COMPUTATIONAL COMPLEXITY

In deep learning, the index of floating-point operations (FLOPs) is often used to measure the computational complexity. The key to obtain FLOPs is to find out the trainable parameters. As mentioned before, M is the number of sensors, k is the encoder layer number. There are k_1 neurons in the first layer. According to equations (9) and (9), total trainable parameters of the encoder and decoder are $2 \times [(M \times k_1 + k_1) + (k_1 \times k_2 + k_2) + \ldots + (k_{n-1} \times k_n + k_n)]$. GRU has three layers like a normal neural network. Each layer is linked by three sets of weight matrixes and bias vectors, corresponding to two gates and the candidate state. The number of neurons in the hidden layer is determined by k_n. According to (11), (12), (13), (17), temporal model has $4 \times (3 \times k_n^2 + k_n)$ variables need to be trained. For the proposed algorithm, all trainable variables have $2 \times [(M \times k_1 + k_1) + (k_1 \times k_2 + k_2) + \ldots + (k_{n-1} \times k_n + k_n)] + 4 \times (3 \times k_n^2 + k_n)$. Although the advanced regularization and activation strategies in this method can reduced the computation of gradient back-propagation to a certain extent, they do not change the order of magnitude of FLOPs overall. Using Big O notation to describe the time complexity is $O(n^2)$.

IV. EXPERIMENT

In this section, to evaluate the proposed model's effectiveness, two chemical industrial processes: catalytic rod reaction and tubular reactor with recycling are set. The numerical experiments are configured on a computer with: Intel i5 6300HQ CPU, 12GB RAM, Nvidia GTX 960M GPU, Windows 10, and Pytorch 1.7. The following indexes are given for comparison among traditionally statistical learning methods, the proposed method, and the same type of deep learning methods.

Algorithm 1 SSAE-GRU Based Modeling for Nonlinear DPSs

Input: Measured data $Y(x,t)$, control inputs $u(t)$

Output: Spatiotemporal model and its parameters θ and prediction $\tilde{Y}(x,t)$

```
Normalize Y(x,t) into range [0, 1]; split data into train set and test set
Randomly initialize encoder [W_e, b_e] and decoder [W_d, b_d] within [0,1], Δw = Δb = 0
Initialize GRU's weights and biases from U(−√k, √k), where k = 1/2, h_{p−1} = 0
Set max iteration I, GRU's neuron P
for i = 1 : I do
    Update the parameters with (9) to calculated y(t): W_e ← W_e + Δw, b_e ← b_e + Δb
    for p = 1 : P do
        z_p ← y_p, h_{p−1} by (11)
        r_p ← y_p, h_{p−1} by (12)
        n_p ← y_p, h_{p−1}, r_p by (13)
        h_p ← z_p, n_p, h_{p−1} by (14)
    end for
    α(y(t)) ← h_p
    Initial GRU's parameter and repeat steps 7-12:
    η(u(t)) ← h_p
    ŷ(t) ← α(y(t)) + η(u(t))
    Update the parameters with (9) to calculated Ỹ(x,t): W_d ← W_d + Δw, b_d ← b_d + Δb
    Calculated loss J(θ) by (19)
    Fine-tuning the parameters with back-propagation by (20), (21), (22) and (23)
    save all the trainable parameters θ in memory
    if loss < best loss then
        update θ
    end if
end for
```

Root of mean squared error:

$$RMSE = \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{t=1}^{L}\left(Y(x,t) - \tilde{Y}(x,t)\right)^2} \tag{24}$$

Spatiotemporal prediction error:

$$e(x,t) = Y(x,t) - \tilde{Y}(x,t) \tag{25}$$

Principally, the total spatiotemporal error situation of all sensors over all periods is described by RMSE. SPE indicates the error between the whole prediction process and the sampling process of each sensor at each time. The computation time required for each model is also given.

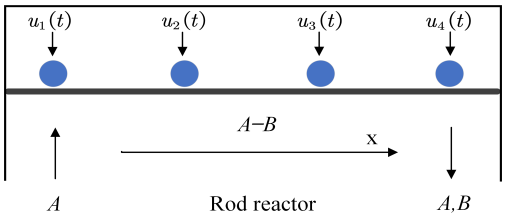


FIGURE 6. Simplified physical diagram of a catalytic rod.

A. CATALYTIC ROD

The catalytic rod reaction [38] is a benchmark experiment for testing the effectiveness of the time-space prediction model. A long thin rod in a reactor as shown in Fig. 6 is a typical transport-reaction process in the chemical industry. The reactor is fed with pure species A and a zeroth order exothermic catalytic reaction of the form A → B takes place in the rod. With the settings of constant density and heat capacity of the rod, constant conductivity of the rod, constant temperature at both sides of the rod, and excess of species A in the furnace, the mathematical expression, which interprets the spatiotemporal-varying of the rod temperature, has the following parabolic PDE:

$$\frac{\partial Y(x,t)}{\partial t} = \frac{\partial^2 Y(x,t)}{\partial x^2} + \beta_T \left(\mathrm{e}^{-\gamma/(1+Y)} - \mathrm{e}^{-\gamma}\right) + \beta_u \left(b(x)^{\mathrm{T}} u(t) - Y(x,t)\right) \quad (26)$$

subject to the boundary and initial condition:

$$Y(0,t) = 0, \quad Y(\pi,t) = 0, \quad Y(x,0) = Y_0(x) \quad (27)$$

where $Y(x,t)$, $u(t)$, $b(x)$, β_T, β_u and γ denote the temperature in the reactor, the manipulated input (temperature of the cooling medium), the actuator distribution, the heat of reaction, the heat transfer coefficient and the activation energy, respectively. The process parameters are set as: $\beta_T = 50$, $\beta_u = 2$, $\gamma = 4$. As the first step for the model identification, suitable input signals is very important for gathering informative data. Four actuators: $u(t) = [u_1(t), u_2(t), u_3(t), u_4(t)]^{\mathrm{T}}$ are employed to excite the nonlinearity of process.

$$b_i(x) = H\left(x - \frac{(i-1)\pi}{4}\right) - H\left(x - \frac{i\pi}{4}\right), \quad i \in [1,4] \quad (28)$$

$H(\cdot)$ is the standard Heaviside function. More specifically, the temporal input

$$u_i(t) = 1.1 + (6\cdot\tau)e^{(-i/5)}\sin(50\cdot t/7 + 2.5\cdot\tau) - 0.4\cdot e^{(-i/20)}\sin(50\cdot t + 2.5\cdot\tau)\,(i = 1,\ldots,4) \quad (29)$$

where τ is a uniform distributed random function on [0,1]. Twenty sensors are placed to be distributed along the rod, the sampling interval is designed as 0.01 and total simulation time is 7.5s. 750 samples are collected as the original data.

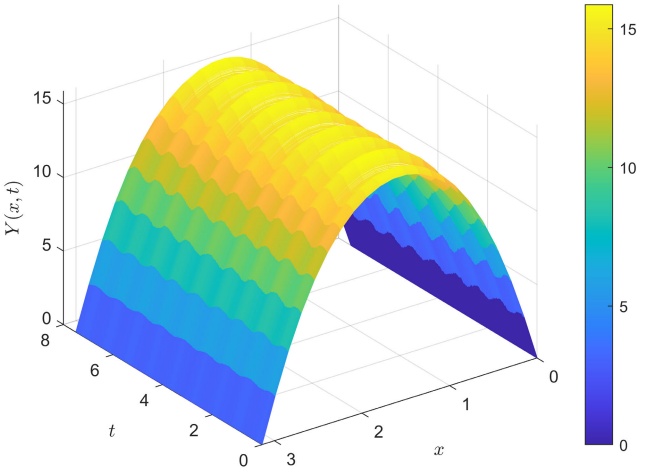


FIGURE 7. Measured outputs of catalytic rod.

The first 500 sample are used as training samples, and remaining 250 sample are testing data. Fig. 7 illustrates the measured outputs of spatiotemporal dynamics in catalytic rod.

SSAE is designed to achieve nonlinear projection and reconstruction learning. Existing research case selected 2 as the proper dimension of the systems. The whole structure of SSAE with bottleneck layer is confirmed as 20-10-2-10-20. Reduced order time series $y_1(t), y_2(t)$ computed from original system measurements $\{Y(x_m, t_n)\}_{m=1,n=1}^{20,750}$ are used as the true value (solid line in Fig. 8) to train and test the sequence model.

Two GRUs identify the temporal law based on the input signal $\{u_i(t_n)\}_{i=1,n=1}^{4,750}$ and time series $\{y_i(t)\}_{i=1,n=1}^{2,750}$. The number of hidden layer nodes of each GRU is set as 2. Then, the overall network is optimized. The calculation process and hyperparameter selection of the optimizer are shown in Section III.D. The max iterations are set as 1500. While the spatiotemporal model of whole reaction process is obtained, temperature predicted by the temporal model under the manipulated input conditions are compared with the actual measured temperature at the specified times to validate the proposed model. As indicated in Fig. 8, the predictions (dotted line) given by GRU model can track the tendency of true value.

After spatiotemporal integration, the prediction of high-dimensional temperature distribution is shown in Fig. 9. Comparing the predicted temperature with the original value, the maximum deviation does not exceed $0.1°C$, that is, the error is less than 1%. The SSAE-GRU model is qualified to reflect the spatiotemporal dynamics of the original system.

In order to quantify the average error of the overall sample, the RMSE criterion is introduced. Table 1 compares the RMSE values of PCA-RBF, NL-PCA-RBF [6], DS-Volterra [27], AE-RNN [18] and the proposed method in the catalytic rod case. PCA-RBF is used as a benchmark method. Note that the proposed method is executed 20 times using randomly initialized weights. The mean and standard deviation of RMSE values in training and testing are listed, respectively. The prediction errors given by SSAE-GRU are the

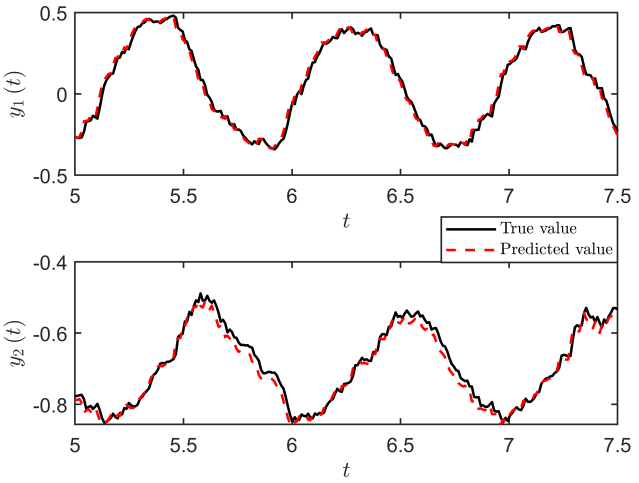


FIGURE 8. Comparison of the true value and predicted value.

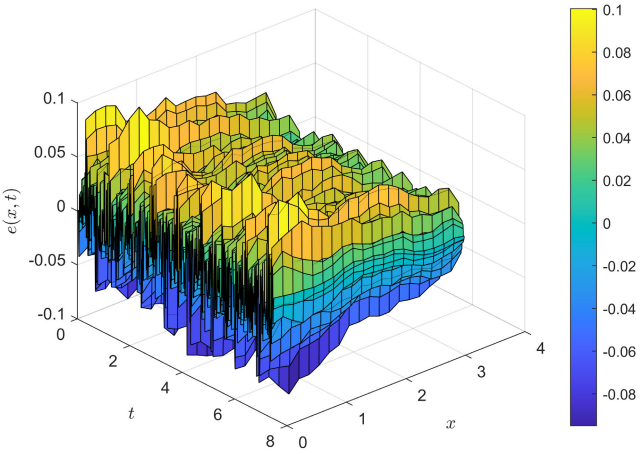


FIGURE 9. SPE of the proposed model in catalytic rod example.

TABLE 1. Comparison of RMSE in catalytic rod.

Methods	RMSE_tr	RMSE_te
PCA-RBF	0.1040	0.0956
NL-PCA-RBF [6]	0.0570	0.0591
DS-Volterra [27]	0.0510	0.0514
AE-RNN [18]	0.0452	0.0456
SSAE-GRU	0.0387	0.0389

smallest among all these models. It is 35.8% more accurate than the statistical-based NL-PCA-RBF model. When contrasted to DS-Volterra, the accuracy of SSAE-GRU is 29.5% increased. Compared with AE-RNN, which is the same type of deep learning method, the accuracy is improved by 18.3%. The results illustrate the predictive stability of the proposed method. Table 2 shows the training time of models. SSAE-GRU spends 14.3764s in training, which is close to DS-Volterra 13.5385s, far less than the training time of AE-RNN 65.1983s and NL-PCA-RBF 102.7345s.

B. TUBULAR REACTORS WITH RECYCLE

Tubular reactors are widely used for the production of a variety of industrial products and are characterized by a strong

TABLE 2. Comparison of modeling time in catalytic rod.

Methods	Training Time
PCA-RBF	1.5632s
NL-PCA-RBF [6]	102.7345s
DS-Volterra [27]	13.5385s
AE-RNN [18]	65.1983s
SSAE-GRU	14.3764s

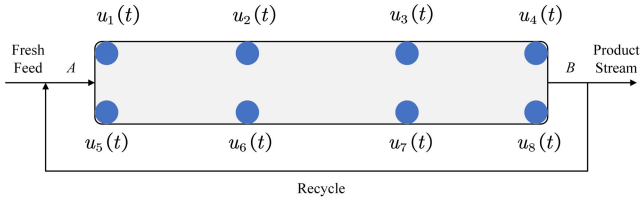


FIGURE 10. Simplified physical diagram of a tubular with recycle.

coupling of diffusive, convective, and reactive mechanisms. In tubular reactors where highly, exothermic reactions take place. To reduce the 'hot spot', a recycle loop around the reactor was used to return the unreacted reactant to the reactor. We considered a non-isothermal tubular reactor without catalyst packing, shown in Fig. 10, where an irreversible first-order reaction of the form $A \rightarrow B$ took place. Tubular reactors with recycle can be modeled by the systems of parabolic PDE [39]. Data with a small number of degrees of freedom can describe the main feature of these systems. MOR techniques incredibly reduce the complexity of the internal complex dynamical systems while maintaining the accuracy of its input and output behavior, thereby significantly saving simulation time [40]. The spatiotemporal dynamic of the tubular reactor was expressed by the following formulas:

$$\frac{\partial C}{\partial t} = -\frac{\partial C}{\partial x} + \frac{1}{P_{eC}}\frac{\partial^2 C}{\partial x^2} - f(C,Y) \tag{30}$$

$$\frac{\partial Y}{\partial t} = -\frac{\partial Y}{\partial x} + \frac{1}{P_{eY}}\frac{\partial^2 Y}{\partial x^2} + B_Y f(C,Y) + \beta_Y (b(x)u(t) - Y) \tag{31}$$

where C and Y are the dimensionless reactant concentration and temperature, respectively. $f(C,Y) = B_C C e^{\gamma Y/(1+Y)}$ is the reaction term. B_C and B_Y denote a dimensionless pre-exponential factor and a dimensionless heat of a reaction, respectively. γ and β_Y are a dimensionless activation energy and a dimensionless heat transfer coefficient. A recycle is used here to return part of the reactants in the output stream to the feed stream at a ratio r. The parameters used are $P_{eC} = 7.0$, $P_{eY} = 7.0$, $B_C = 0.1$, $B_Y = 2.5$, $\gamma = 10.0$, $r = 0.5$, and $\beta_Y = 2.0$ [18], [41]. The boundary conditions for the concentration and temperature at $x = 0$ are as follows:

$$\frac{\partial C}{\partial x} = -P_{eC}\left[(1-r)(1+C_0) + rC(t,1) - C(t,0)\right] \tag{32}$$

$$\frac{\partial Y}{\partial x} = -P_{eY}\left[(1-r)(1+Y_0) + rY(t,1) - Y(t,0)\right] \tag{33}$$

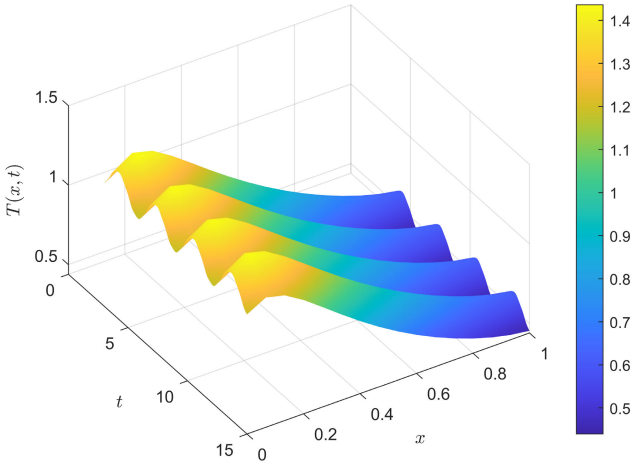


FIGURE 11. Measured outputs of tubular reactor with recycle.

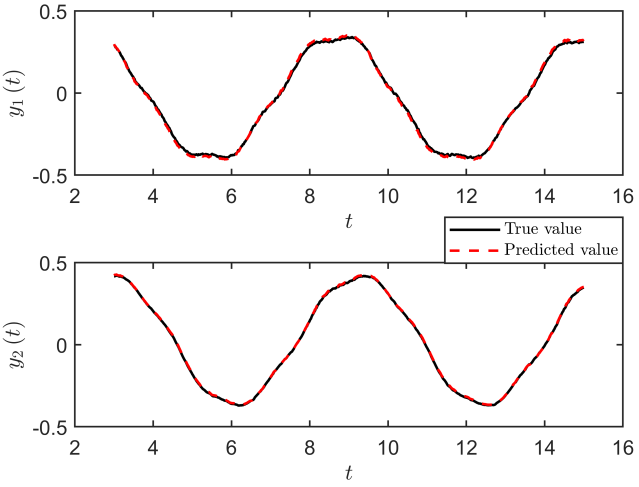


FIGURE 12. Comparison of the true value and predicted value.

The boundary conditions at $x = 1$ are $dC/dx = 0$ and $dY/dx = 0$, $u(t)$ are jacket temperature zones (actuators) and $b(x)$ is the actuator distribution function. Under these circumstances, each control input snapshot $u(t)$ consists of eight manipulated inputs $u_i(t) = [u_1(t), \ldots, u_8(t)]^T$ located based on the spatial distribution function $b(x) = [b_1(z), \ldots, b_8(x)]^T$ [42] given by the following expression:

$$b_i(x) = H(x - (i-1)/8) - H(x - i/8) \tag{34}$$

The manipulated inputs $u_i(t)$ is designed as follow:

$$u_i(t) = 0.15 + (0.2 + 0.05\tau)\exp(-i/10)\sin(2\tau + 0.2\tau) - 0.02\exp(-i/20)\sin(10\tau + 0.2\tau) \tag{35}$$

The system is detected by 16 sensors and sampled at time interval $\Delta t = 0.01$. Each temperature snapshot Y_t is collected from 16 heat exchanges of the equal surface. The total simulation time is $15s$ and the first 300 snapshots were discarded because the exothermic reaction raised the temperature quickly after $t = [0, 2]$ along the entire reactor. Our modeling method focus on forming long-term monitoring of chemical systems after stabilization. Fig. 11 shows the steady-state of the reactor. 600 snapshots are chosen as the training data while the remaining 600 snapshots are tested.

The construction of SSAE in this condition uses a 16-8-2-8-16 with bottleneck layers architecture. Control inputs are fed into the GRU through a two-layers 8-2 architecture. The other GRU has a 2-2 architecture. Two-layer GRU structure is more attractive for learning the evolution law in the time dimension. Other parameter settings are the same as in the catalytic rod reaction case.

The low dimensional time series models predicted by two GRUs have excellent performance and captured the time relationship of the low dimensional model and control inputs as indicated in Fig. 12. The SPE distribution of proposed method is illustrated in Fig. 13. The maximum does not exceed $4 \times 10^{-3}{}^{\circ}C$, 0.5%. It is satisfactory to achieve a high level of prediction accuracy. Table 3 shows the RMSE

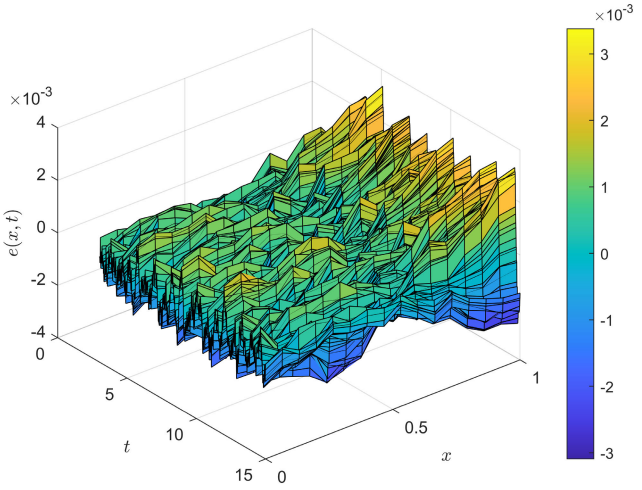


FIGURE 13. SPE distribution in tubular reactor with recycle.

TABLE 3. Comparison of RMSE in tubular with recycle.

Methods	RMSE_tr	RMSE_te
K-L decomposition	0.079	0.078
DS-Volterra [27]	0.0084	0.0085
AE-RNN [18]	0.0070	0.0074
SSAE-GRU	0.0063	0.0064

using the nonlinear methods is only one tenth than that using the K-L decomposition. The RMSE of the proposed method is 0.0064, and the best results are obtained again in the second chemical numerical experiment. Table 3 provides that SSAE-GRU is improved compared to DS-Volterra 24.7% and AE-RNN 14.8% respectively. Table 4 shows that the consuming time of training SSAE-GRU is 15.6413s, which is 77.2% less than that of AE-RNN, 67.4932s.

C. ANALYSIS AND COMPARISON

Compared with linear dimensionality reduction approaches, such as K-L decomposition, the nonlinear dimensionality reduction approaches with stronger representation ability have more advantages in dealing with DPSs with

TABLE 4. Comparison of modeling time in tubular with recycle.

Methods	Training time
K-L decomposition	2.0105s
DS-Volterra [27]	14.8231s
AE-RNN [18]	67.4932s
SSAE-GRU	15.6413s

complex nonlinear parameters and boundary conditions. Due to its deep network structure, deep learning technique has stronger nonlinear representation ability than other nonlinear approaches. Although DS-Volterra model ensures good modeling efficiency and reduces the occupation of communication resources, the multi-layer SSAE can achieve better performance with the similar amount of computation.

Although the calculation of GRU is more complicated than vanilla RNN and some simplified versions, we use GRU because we believe that prediction accuracy is supposed to consider a higher priority. Thanks to ELU and L2 regularization, neurons in the model have the property of sparse activation. The proposed method is superior to AE-RNN in model execution time. Successful training of AE-RNN requires tens of thousands of iterations. Compared with this, the proposed method only needs about 1/10, which is an important reason for the training time advantage. SSAE-GRU stacked with multi-layers of neurons means that it is impractical to compete with the linear method of computing cost. However, the key to modeling DPSs is accuracy, and then consider the calculation time under this premise. Therefore, the proposed method helps to overcome this challenge and is meaningful.

V. CONCLUSION

In this work, a novel data-driven model named SSAE-GRU is proposed for modeling of spatiotemporal-varying DPSs. By introducing deep learning, the body of knowledge on accurate modeling of DPSs is expanded. This deep learning technology-based model is trained using a jointly modular learning approach. In this way, the spatiotemporal model is inherited and updated in a relatively simple way throughout the training process. From the perspective of modeling accuracy, the proposed model is based on the precise SSAE dimensionality reduction technique and the GRU time series prediction technique to solve the long-term dependency problem. Therefore, it study the intrinsic parameters of the physical equation from the data extremely, which is close to the actual application in the manufactures. From the perspective of modeling efficiency, the introduction of sparse nature and the joint learning strategy can lead to a simple structure, simplify the model training process, and accelerate learning. Thus, this method is applied to a class of time/space coupled DPSs and is a excellent black-box model. Experiments on catalytic rods and circulating tubular reactors demonstrate the efficiency and feasibility of the proposed model. In our future study, we will focus on the following three aspects.

1) From the perspective of simulation results and DPSs itself, there is a dramatic change at the boundary. And the accuracy of low-dimensional representations may be affected. How to reduce the influence of this phenomenon on the design of MOR techniques is a key problem we need to solve.

2) Theoretically, deep networks' representational ability is stronger. We will extend the application of proposed method to other types of industrial processes represented by DPSs.

3) We will also consider incorporating low-order models into the field of predictive control.

REFERENCES

[1] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: A review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.
[2] X. Lu, W. Zou, and M. Huang, "A novel spatiotemporal LS-SVM method for complex distributed parameter systems with applications to curing thermal process," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 1156–1165, Jun. 2016.
[3] W. Shen, K. Xu, L. Deng, and S. Zhang, "A karhunen-loeve Galerkin online modeling approach for the thermal dynamics of li-ion batteries," *IEEE Access*, vol. 8, pp. 187893–187901, 2020.
[4] C. Zhu, H. Yang, Y. Fan, B. Fan, and K. Xu, "Online spatiotemporal modeling for time-varying distributed parameter systems using kernel-based multilayer extreme learning machine," *Nonlinear Dyn.*, vol. 107, no. 1, pp. 761–780, Jan. 2022.
[5] H.-X. Li and C. Qi, "Modeling of distributed parameter systems for applications—A synthesized review from time–space separation," *J. Process Control*, vol. 20, no. 8, pp. 891–901, Sep. 2010.
[6] C. Qi and H.-X. Li, "Nonlinear dimension reduction based neural modeling for distributed parameter processes," *Chem. Eng. Sci.*, vol. 64, no. 19, pp. 4164–4170, Oct. 2009.
[7] H.-X. Li and C. Qi, *Spatio-Temporal Modeling of Nonlinear Distributed Parameter Systems: A Time/Space Separation Based Approach*, vol. 50. Berlin, Germany: Springer, 2011.
[8] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 377–386, Jan. 2016.
[9] L. Ai and Y. San, "Model predictive control for nonlinear distributed parameter systems based on LS-SVM," *Asian J. Control*, vol. 15, no. 5, pp. 1407–1416, 2013.
[10] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: A comparative," *J. Mach. Learn. Res.*, vol. 10, nos. 66–71, p. 13, 2009.
[11] K. Xu, B. Fan, H. Yang, L. Hu, and W. Shen, "Locally weighted principal component analysis-based multimode modeling for complex distributed parameter systems," *IEEE Trans. Cybern.*, early access, Mar. 18, 2021, doi: 10.1109/TCYB.2021.3061741.
[12] S. Pitchaiah and A. Armaou, "Output feedback control of distributed parameter systems using adaptive proper orthogonal decomposition," *Ind. Eng. Chem. Res.*, vol. 49, no. 21, pp. 10496–10509, Nov. 2010.
[13] K.-K. Xu, H.-X. Li, and Z. Liu, "ISOMAP-based spatiotemporal modeling for lithium-ion battery thermal process," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 569–577, Feb. 2018.
[14] H. Jiang and Y. Dong, "Dimension reduction based on a penalized kernel support vector machine model," *Knowl.-Based Syst.*, vol. 138, pp. 79–90, Dec. 2017.
[15] K.-K. Xu, H.-X. Li, and H.-D. Yang, "Local-properties-embedding-based nonlinear spatiotemporal modeling for lithium-ion battery thermal process," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9767–9776, Dec. 2018.
[16] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[17] M. Wang, H.-X. Li, X. Chen, and Y. Chen, "Deep learning-based model reduction for distributed parameter systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 12, pp. 1664–1674, Dec. 2016.

[18] X. Qing, J. Jin, Y. Niu, and S. Zhao, "Time–space coupled learning method for model reduction of distributed parameter systems with encoder–decoder and RNN," *AIChE J.*, vol. 66, no. 8, Aug. 2020, Art. no. e16251.

[19] X. J. Lu, F. Yin, C. Liu, and M. H. Huang, "Online spatiotemporal extreme learning machine for complex time-varying distributed parameter systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1753–1762, Aug. 2017.

[20] L. Ai, Y. Xu, L. Deng, and K. L. Teo, "Least squares support vector machine-based multivariate generalized predictive control for parabolic distributed parameter systems with control constraints," *Symmetry*, vol. 13, no. 3, p. 453, Mar. 2021.

[21] X. Jin, H. D. Yang, K. K. Xu, and C. J. Zhu, "Dual extreme learning machines-based spatiotemporal modeling for nonlinear distributed thermal processes," *Int. J. Comput. Methods*, vol. 18, no. 1, Feb. 2021, Art. no. 2050026.

[22] X. Lu, P. He, and J. Xu, "Error compensation-based time-space separation modeling method for complex distributed parameter processes," *J. Process Control*, vol. 80, pp. 117–126, Aug. 2019.

[23] L. Chen, H.-X. Li, and S. Xie, "Modified high-order SVD for spatiotemporal modeling of distributed parameter systems," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 4296–4304, Apr. 2022.

[24] Y. Zhou, H.-X. Li, and S.-L. Xie, "Fast modeling of battery thermal dynamics based on spatio-temporal adaptation," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 337–344, Jan. 2022.

[25] Y. Zhou, H. Deng, and H.-X. Li, "Optimal-sensing-based recursive estimation for temperature distribution of pouch-type batteries," *IEEE Trans. Transport. Electrific.*, early access, May 2, 2022, doi: 10.1109/TTE.2022.3171857.

[26] S. Gupta, A. K. Sahoo, and U. K. Sahoo, "Nonlinear space–time varying parameter estimation using consensus-based in-network distributed strategy," *Digit. Signal Process.*, vol. 79, pp. 175–189, Aug. 2018.

[27] S. Gupta, A. K. Sahoo, and U. K. Sahoo, "Wireless sensor network-based distributed approach to identify spatio-temporal Volterra model for industrial distributed parameter systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7671–7681, Dec. 2020.

[28] S. Gupta, A. K. Sahoo, and U. K. Sahoo, "Volterra and Wiener model based temporally and spatio-temporally coupled nonlinear system identification: A synthesized review," *IETE Tech. Rev.*, vol. 38, no. 3, pp. 303–327, May 2021.

[29] Y. Fan, K. Xu, H. Wu, Y. Zheng, and B. Tao, "Spatiotemporal modeling for nonlinear distributed thermal processes based on KL decomposition, MLP and LSTM network," *IEEE Access*, vol. 8, pp. 25111–25121, 2020.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 1–8.

[32] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[34] Y. Yu, X. Si, C. Hu, and Z. Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[35] K. Xu, X. Tan, B. Fan, T. Xiao, X. Jin, and C. Zhu, "Dual extreme learning machine based online spatiotemporal modeling with adaptive forgetting factor," *IEEE Access*, vol. 9, pp. 67379–67390, 2021.

[36] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1—Learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[38] P. Christofides and J. Chow, "Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes," *Appl. Mech. Rev.*, vol. 55, no. 2, pp. 29–30, Mar. 2002.

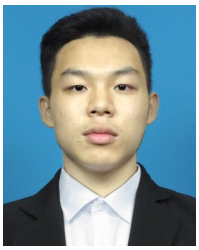
[39] C. S. Bildea, A. C. Dimian, S. C. Cruz, and P. D. Iedema, "Design of tubular reactors in recycle systems," *Comput. Chem. Eng.*, vol. 28, nos. 1–2, pp. 63–72, Jan. 2004.

[40] C. Antoniades and P. D. Christofides, "Studies on nonlinear dynamics and control of a tubular reactor with recycle," *Nonlinear Anal., Theory, Methods Appl.*, vol. 47, no. 9, pp. 5933–5944, Aug. 2001.

[41] W. Xie, I. Bonis, and C. Theodoropoulos, "Off-line model reduction for on-line linear MPC of nonlinear large-scale distributed systems," *Comput. Chem. Eng.*, vol. 35, no. 5, pp. 750–757, May 2011.

[42] I. Bonis, W. Xie, and C. Theodoropoulos, "Multiple model predictive control of dissipative PDE systems," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1206–1214, May 2014.

LING AI received the bachelor's degree in automation and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, China, in 2005 and 2014, respectively. From 2017 to 2018, he was a Visiting Postdoctoral Researcher at the Department of Mathematics and Statistics, Curtin University, Australia. He is currently with the Department of Automation, Harbin University of Science and Technology. His research interests include process control, predictive control, and intelligent modeling and control.



JUNZHE GAN received the bachelor's degree in electrical engineering and automation from the Changchun University of Science and Technology, China, in 2020. He is currently pursuing the master's degree with the Department of Automation, Harbin University of Science and Technology. His research interests include deep learning, intelligent modeling, and control.

XIANJIE FENG received the bachelor's degree in packaging engineering from the Harbin University of Commerce, China, in 2020. He is currently pursuing the master's degree in software engineering with the Department of Computer Science and Technology, Harbin University of Science and Technology. His research interests include machine learning, deep learning, quantum mechanics, natural language processing, and artificial intelligence.

XUEQIN CHEN received the B.S. degree in automation, the M.S. degree in spacecraft design, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, in 2003, 2005, and 2008, respectively. She is currently a Professor with the Harbin Institute of Technology. Her main research interests include fault diagnosis and fault-tolerant control.