ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

Deep Koopman Neural Network Based Process Monitoring for Stochastic Production System

Yanhui Liu[1,a,*], Xin Jin[1,b], Saiwei Wang[1,c]

[1]*Key Laboratory of Smart Manufacturing in Energy Chemical Process, East China University of Science and Technology, Shanghai, China*
[a]*yanhui_liu1998@163.com*, [b]*jx_9810@163.com*, [c]*842376017@qq.com*
[*]*Corresponding author*

Abstract: *Stochastic production system (SPS) refers to a production process that is influenced by a large number of random factors, typical examples including industrial biosystem, composite material production system, and batch chemical reaction system. Notably, SPS is notorious for significant uncertainty and stochasticity, thereby making implementing process monitoring to ensure product quality a daunting task. One of the major underlying obstacles is how to accurately detect anomalies thereof in real time. To resolve so, this paper proposes a deep Koopman neural network based approach, wherein two deep neural networks constitute a bijective mapping between original data space and a linear high-dimensional space, and a linear operator describes dynamic evolution in the linear space. The performance of the proposed method is tested on two examples of SPS, which are of significant intrinsic stochastic dynamics, hence arguably constituting a novel class of benchmarks for performance comparing of various process monitoring algorithms, and becoming another contribution of this paper.*

Keywords: *Koopman Operator Theory, Stochastic Production System, Deep Learning, Anomaly Detection, Process Monitoring*

1. Introduction

Stochastic production system (SPS) has shown remarkable potential in various scenarios, such as fermentation, pharmaceutical industry, and composite material production [1]-[7]. The quality constraints imposed on the products in these scenarios are mostly stringent, thereby process monitoring as an effective means to ensure product quality becoming necessary. Unfortunately, SPS is notorious for its both significant intrinsic stochasticity and measurement uncertainty in the context of control theory. These are attributed to two reasons: (i) The system states of SPS are sensitive to a myriad of exogenous factors, such as inputs, environmental factors, and equipment conditions, which can affect the quality and performance of the final product. (ii) The lack of accurate in-situ measuring means introduces extra level of noisiness to the available data of SPS. Altogether, process monitoring for SPS is indispensable yet challenging.

In past decades, there are a few methods already developed for SPS process monitoring. For instance, multiway Principal Component Analysis (PCA) has been widely applied to serve the purpose, due to its simple projection structure, low dimension of computing space and fast processing of high dimensional data collectively [8]. As multiway PCA is essentially a linear method, it is incapable of handling nonlinearity in dynamics, which motivated the use of the kernel method to map data into a high-dimensional feature space where data is linearly separable [9]. Other notable efforts include an improved Independent Component Analysis (ICA) method [10], a two-step modeling strategy named kernel ICA-PCA method [11] and a multiway kernel entropy ICA method [12] developed for capturing nonlinear and non-Gaussian features embedded in SPS data. Additionally, Support Vector Machines (SVM) integrated with PCA or fuzzy reasoning are able to obtain robust decision functions for anomaly detection in SPS [13], [14]. Nevertheless, such methods only have mild capabilities to handle nonlinearity in data, while heavy tail and multimodal are common in SPS data. Besides, the hyperparameters tuning in the aforementioned methods is cumbersome.

The well-known universal approximation theorem confers neural network the capability of representing any function between inputs and outputs [15], thus showing grand promise for SPS process monitoring. It is the case. Amid all deep learning methods, auto-associative neural network [16], [17] and deep neural network [18]-[22] are two categories widely studied for SPS process monitoring. Sometimes these

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

methods become problematic in practice, as they assume that the samples are independently distributed and the dynamic correlation is overlooked. The recurrent neural networks, Long Short-Term Memory (LSTM) [23] in particular, are good at predicting future evolutions given the current and historical data and suitable for anomaly detection as well [24], [25]. However, they are also criticized for poor interpretability, as physical insights can hardly be gleaned.

Koopman Operator Theory (KOT) developed by Bernard O. Koopman in 1930s recently emerged to a curer for the acute interpretability issue [26]. The underlying idea is that there exists a bijective mapping such that the original complex dynamics are equivalently represented by evolutions in an infinite-dimensional linear space, thus providing a route for global linearization in a stark contrast to the predominant Jacobian linearization method [27]. However, finding such a bijective mapping is not trivial. Notable endeavors include Dynamic Mode Decomposition (DMD) [28], [29] and its extension (eDMD) [30], both of which use data-driven method to find an approximation to Koopman eigenfunctions in a finite-dimensional space. Despite the successes in many reported cases [31], [32], its effectiveness heavily depends on the intricacy of selecting candidate functions. To resolve so, deep neural networks were employed to approximate the candidate functions as its universal approximation capability was used [15]. Some encouraging results of applying the neat method to cast predictions for fluid dynamics and chaotic dynamics are reported in [33], [34]. Nevertheless, KOT is not massively studied in the field of process monitoring except a few efforts. Anomaly in power network was detected by comparing the maximal norm of Koopman eigenvectors to some thresholds in [35], while KOT was used to reconstruct mechanical signals for fault detection in [36]. Alternatively, Reference [37] integrated KOT with k-Nearest Neighbor (KNN) to monitor machinery health. Note that deep neural networks are used in none of the aforementioned.

In this paper, we present a KOT based method for anomaly detection in SPS. Deep neural networks are used as an autoencoder to approximately establish a bijective mapping that maps original complex dynamics onto a space, where dynamics are evolving linearly. After well trained on history data, the residue is analyzed by Support Vector Data Description (SVDD) method, and further used to determine a threshold for anomaly detection [38]. The anomaly is detected by comparing the prediction cast by the established KOT model against the threshold yielded. The contributions of the paper are three-fold as summarized below:

- We introduce a novel stochastic system, which is distilled from SPS but captures its essential dynamic characteristics. The stochasticity is neither additive (noise added to process variables) nor multiplicative (process variables scaled randomly); both cases have been well studied in process monitoring literature, but neither is our focus here. The stochasticity de facto stems from the randomness of reaction time, which has been overlooked in the field of process monitoring but prevalently discussed in systems biology. As such, the system arguably constitutes a new class of benchmarks for evaluating various process monitoring methods.

- As the data is inherently noisy, the ensemble method is used for data curation. It is found that the system dynamics cannot be sufficiently characterized by the mean values, thus demanding high-order moments. Practically, the combination of mean, variance, and third-order moment becomes the best tradeoff between performance and complexity.

- Integrated with the ensemble method, a Deep Koopman Neural Network (DKNN) model is developed for the purpose for anomaly detection in SPS. Unlike the approach in [33], the linear dynamics are captured by a linear layer instead of an auxiliary network, which is much simpler for implementation.

The rest of this paper is organized as follows: Section 2 presents the problem statement; Section 3 elaborates the methods; Section 4 discusses the results; and Section 5 draws the conclusion.

2. Problem Statement

The most representative example of SPS is biochemical systems, and hence we will focus on it to showcase the developed method in the rest of the paper. The essence of biochemical systems is to convert substrates into high-value-added metabolites by living organisms (mostly cells). One of the major impediments for biochemical system production in high quality and quantity stems from the existence of a subpopulation of cells showing remarkably reduced production efficiency and capacity, which is termed as population heterogeneity in synthetic biology [1]. Such heterogeneity is an inevitable consequence of stochastic gene expression, which is solidly supported by massive single-cell experiments [39], [40]. In the context of biochemistry, gene expression indeed consists of a set of biochemical reactions with the

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

participation of various macromolecules harbored in microscopic reactors (cells). The scarce of such macromolecules and the random molecular collision in the crowding reaction compartment of limited volume collectively lead to the stochasticity of intracellular biochemical reaction, particularly gene expression. As such, it is plausible to focus on gene expression process, which is the most critical and representative part. Without any loss of generality, any intracellular biochemical reaction can be described by

$$\sum_{i=1}^{N} s_{ir} X_i \xrightarrow{k_r} \sum_{i=1}^{N} p_{ir} X_i, \quad r = 1, 2, \cdots, R, \tag{1}$$

where X_i stands for species i $(i = 1, 2, \cdots, N)$, the stoichiometric coefficients s_{ir} and p_{ir} are nonnegative integers specifying the molecule numbers of reactants and products involved in reaction r respectively, and k_r is the rate constant of reaction r. In the stochastic sense, k_r is inversely proportional to the mean time of two successive reactions. The propensity of reaction r is

$$f_r(\mathbf{n}) = k_r \Omega \prod_{i=1}^{N} \frac{n_i!}{(n_i - s_{ir})! \Omega^{s_{ir}}}, \tag{2}$$

with Ω being the compartment volume and n_i being the molecule number of reactant i. Indeed, the propensity can be loosely understood as the probability of reaction occurrence. For instance, the transcription can be compactly described by

$$G \xrightarrow{\rho} G + M, \tag{3}$$

where G, M stand for gene and messenger RNA (mRNA) respectively, and ρ is the transcription rate constant.

Besides, there are various exogenous factors perturbing the normal operation of biochemical system, such as temperature fluctuation and contamination. The temperature impacts the reaction through reaction constants according to Arrhenius law. Arguably, so is the mechanism of contamination, as contamination may affect the catalytic efficiency of some enzyme. Hence, within the framework, when an anomaly takes place, it is reflected through the change of one or a group of reaction rate constants k_r. The goal of process monitoring then becomes detecting anomaly from the data of reaction species X_i if some reaction rate constant k_r changes.

3. Methods

3.1. Data Acquisition

The dynamics of system (1) can be simulated by the renowned Stochastic Simulation Algorithm (SSA), also known as Gillespie algorithm in systems biology [41]. The basic idea is to draw two random numbers, one for calculating the next reaction time, and the other for determining next reaction type. The pseudocode for SSA is presented as follows.

Algorithm 1 Stochastic Simulation Algorithm
1: **Initialization:** $t \leftarrow 0, \mathbf{n}, t_{max}$
2: **Repeat**
3: Calculate propensities according to (2)
4: Obtain the time step to the next reaction event $\tau = -\ln(u_1)/\lambda$,$\lambda = \sum_{r=1}^{R} f_r(\mathbf{n})$
5: Determine the next reaction event $r = \text{smallest integer satisfying} \sum_{i=1}^{r} f_i(\mathbf{n}) > u_2\lambda$
6: Update time $t \leftarrow t + \tau$
7: Update $\mathbf{n}$ according to (1)
8: **Until** $t > t_{max}$
Output: $\mathbf{n}$

Notably, there is a Julia implementation developed by our group and available on Github as *DelaySSAToolkit*. The package is based on *DiffEqJump*, but more powerful as it is even able to simulate delayed reactions [42].

3.2. Koopman Operator Theory

Here we present a brief summary of Koopman operator theory. For more details, readers are encouraged to refer to [43]. Considering a discrete-time system, whose dynamics are governed by

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k). \tag{4}$$

The state $\mathbf{x}_k$ is only observable through some function φ such that

$$\mathbf{y}_k = \varphi(\mathbf{x}_k) = [\varphi_1(\mathbf{x}), \ \cdots, \ \varphi_n(\mathbf{x})]^\top. \tag{5}$$

As shown in Figure 1, the Koopman operator $\mathbf{K}$ is an infinite-dimensional linear operator acting on observing function φ such that

$$\mathbf{K}\varphi = \varphi \circ \mathbf{F} \Leftrightarrow \mathbf{K}\mathbf{y}_k = \mathbf{y}_{k+1}, \tag{6}$$

where $\circ$ is the composition operator.

Suppose that in some Hilbert space spanned by a set of basis functions ϕ_i termed Koopman eigenfunctions satisfying that

$$\phi_i(\mathbf{x}_{k+1}) = \mathbf{K}\phi(\mathbf{x}_k) = \lambda_i\phi(\mathbf{x}_k). \tag{7}$$

It follows that observing function can be compactly decomposed into

$$\varphi(\mathbf{x}) = \textstyle\sum_{i=1}^{\infty} \phi_i(\mathbf{x})\mathbf{v}_i, \tag{8}$$

with the Koopman mode being $\mathbf{v}_i = [\langle\phi_i, \varphi_1\rangle, \cdots, \langle\phi_j, \varphi_n\rangle]^\top$. As per (7), the evolution of the measurement dynamics can be presented as

$$\varphi(\mathbf{x}_{k+1}) = \mathbf{K}\varphi(\mathbf{x}_k) = \textstyle\sum_{i=1}^{\infty} \lambda_i\phi_i(\mathbf{x}_k)\mathbf{v}_i, \tag{9}$$

which is referred as Koopman mode decomposition and tightly connected to DMD. DMD is indeed a finite truncation of Koopman mode decomposition for a linear system (φ is a linear function) [28], [43].

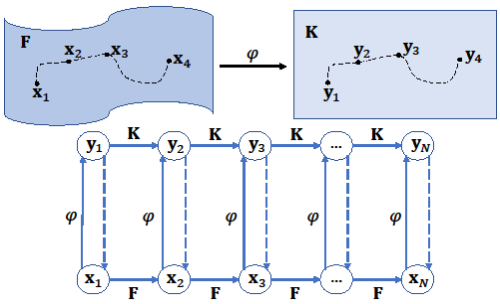

Figure 1: Schematic of Koopman operator theory. An observing function φ maps system states $\mathbf{x}_k$ into a high-dimensional space where measurements $\mathbf{y}_k$ evolving linearly governed by Koopman operator $\mathbf{K}$, while the evolution operator $\mathbf{F}$ of states $\mathbf{x}_k$ is usually nonlinear [43].

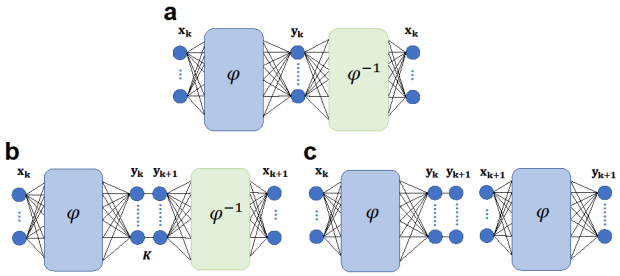

Figure 2: Schematics of deep Koopman neural network. (a) shows an autoencoder φ and φ^{-1} establishing a bijective static mapping between the original space $\mathbf{x}_k$ and the high-dimensional linear space $\mathbf{y}_k$. (b) shows how the DKNN performs one-step prediction. (c) interprets the loss function $\mathcal{L}_c$. The left panel corresponds to $\mathbf{K}\varphi(\mathbf{x}_k)$, while the right stands for $\varphi(\mathbf{x}_{k+1})$.

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

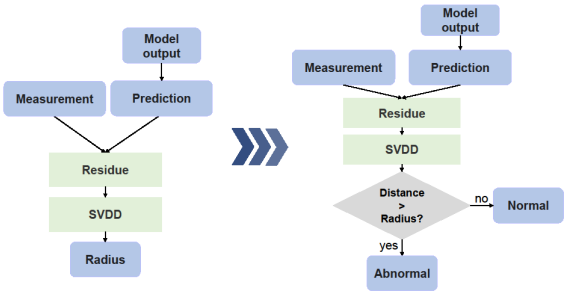

Figure 3: DKNN based anomaly detection protocol. Left: SVDD calculates the radius associated with 90% confidence interval based on the residues between predictions and measurements. Right: An anomaly is detected if the residue of a newly cast prediction is larger than an established radius. Otherwise, the system is still working in normal.

3.3. Deep Koopman Neural Network

KOT is a seemingly elegant theory enabling global linearization but rather difficult to perform, as solving the triplet of the eigenfunction ϕ_i, the eigenvalue λ_i and the mode $\mathbf{v}_i$ is a daunting task. The choice of eigenfunctions is non-trivial and calls for intricate tricks. In stark contrast, deep neural network provides a convenient way to seek the eigenfunctions. Reference [33] reported a neat approach based on a deep autoencoder which constitutes a bijective mapping between the original space and the high-dimensional linear space and approximates the set of the valid eigenfunction bases (see Figure 2a). Note that [33] needs an auxiliary neural network to perform the Koopman operator, and it substantially increases the complexity. As such, we revise the neural network presented in [33] by removing the auxiliary neural network and identifying the linear operator $\mathbf{K}$ directly, which is modeled by a linear network (see Figure 2b). Subsequently, we specify the loss function for the DKNN training. The loss function is composed of five parts, the first three of which is specified as follows

$$\begin{aligned}\mathcal{L}_a &= \left\|\mathbf{x}_k - \varphi^{-1}\left(\varphi\left(\mathbf{x}_k\right)\right)\right\|_{\mathrm{MSE},}\\ \mathcal{L}_b &= \left\|\mathbf{x}_{k+1} - \varphi^{-1}\left(\mathbf{K}\varphi\left(\mathbf{x}_k\right)\right)\right\|_{\mathrm{MSE},}\\ \mathcal{L}_c &= \left\|\varphi\left(\mathbf{x}_{k+1}\right) - \mathbf{K}\varphi\left(\mathbf{x}_k\right)\right\|_{\mathrm{MSE}.}\end{aligned} \tag{10}$$

Here $\mathcal{L}_a$ and $\mathcal{L}_b$ represent the reconstruction error and one-step prediction error in the original space respectively, and $\mathcal{L}_c$ is the one-step prediction error in the high-dimensional linear space (see Figure 2c). The subscript MSE stands for mean squared error.

An $\mathcal{L}_\infty$ term is also used to penalize the data point with the largest loss

$$\begin{aligned}\mathcal{L}_\infty = &\left\|\mathbf{x}_k - \varphi^{-1}\left(\varphi\left(\mathbf{x}_k\right)\right)\right\|_\infty\\ &+ \left\|\mathbf{x}_{k+1} - \varphi^{-1}\left(\mathbf{K}\varphi\left(\mathbf{x}_k\right)\right)\right\|_{\infty.}\end{aligned} \tag{11}$$

Additionally, l_2 regularization is imposed on the neural network weights $\mathbf{W}$ to prevent overfitting

$$\mathcal{L}_{\mathbf{W}} = \|\mathbf{W}\|_2^2. \tag{12}$$

Hence, the total loss function is the weighted summation of all the five parts

$$\mathcal{L} = \alpha_1\mathcal{L}_{\mathrm{a}} + \alpha_2\mathcal{L}_{\mathrm{b}} + \alpha_3\mathcal{L}_{\mathrm{c}} + \alpha_4\mathcal{L}_\infty + \alpha_5\mathcal{L}_{\mathbf{W}}, \tag{13}$$

where α_i for $i = 1, \cdots, 5$ stands for the weight for each parts in the loss function. The DKNN is then determined by solving the optimization problem $\min_{\mathbf{K},\mathbf{W},\mathbf{b}} \mathcal{L}$. For SPS process monitoring, the input $\mathbf{x}_k$ can be the moments (mean, variance, etc.) of molecule counts of interest.

3.4. Anomaly Detection Protocol

With the DKNN model well trained, it is possible to calculate the residues between the model predictions and measurements. Given the residues yielded, the SVDD is used to compute the 90% confidence threshold, which is termed as radius thereafter (see Figure 3Left). In practice, given the historical data, DKNN casts one-step predictions, which are used to compute the residues. The yielded residues are compared with the radius obtained before. If a residue is larger than the radius, an anomaly is

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

detected. Otherwise, the system is still running normally (see Figure 3Right).

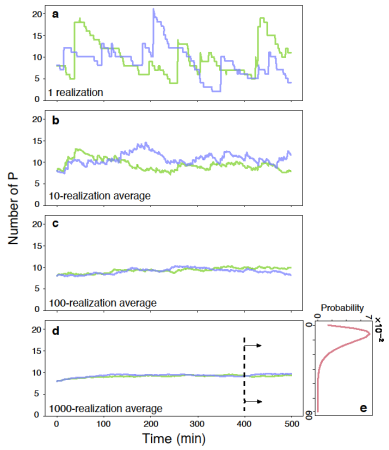

Figure 4: Stochastic simulations for Example 1.

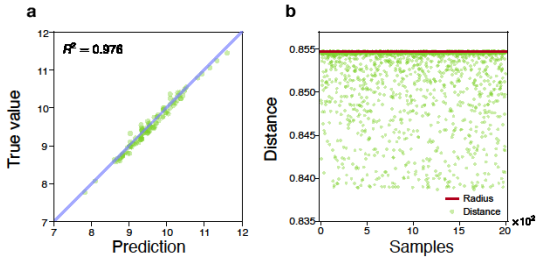

Figure 5: DKNN process monitoring for Example 1 based on mean-value data. (a) shows the DKNN model based on mean-value data cast precise one-step predictions, as predictions (green dots) are close to the line $y = x$ (purple). (b) SVDD calculates the radius (red) for anomaly detection and most samples (green dots) are contained within the radius.

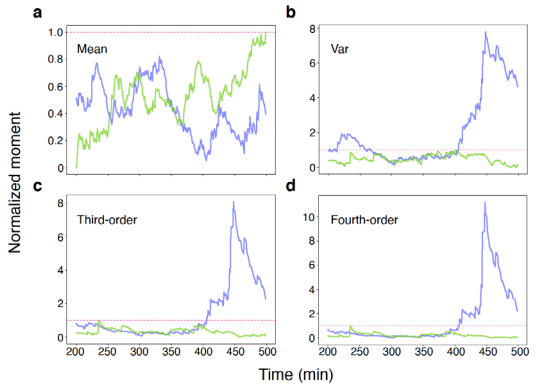

Figure 6: Sensitivity of moments against anomaly. The anomaly occurs at time $t = 401$. All the moments are normalized for visual convenience, and the normalization methods are stated in Appendix 6.2. Moments of order higher or equal to 2 are sensitive to anomaly, while the mean value is not.

Table 1: Anomaly detection F-scores test result for mean-valued data of example 1.

Confidence	90%			
Time (min)	401	420	450	480
F-score (%)	15.365	11.645	14.155	12.744

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

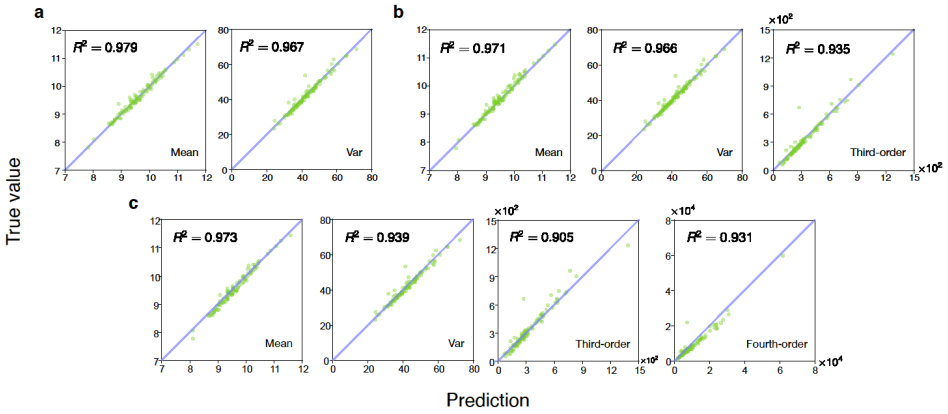

Figure 7: Accuracy of DKNN models one-step prediction for different orders of moments. DKNN model trained on a dataset containing (a) mean and variance; (b) mean, variance and third-order moment; (c) mean, variance, third-order moment and fourth-order moment.

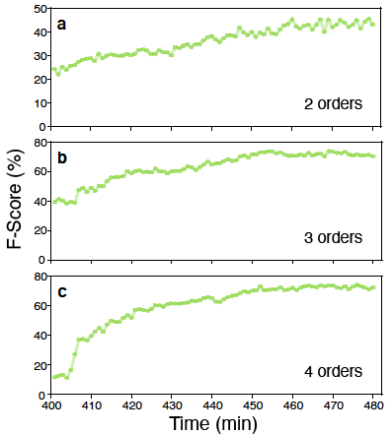

Figure 8: F-score of temporal anomaly detection of three DKNN model trained on dataset containing moments of order up to 2, 3 and 4.

4. Results

Next we unfold the process monitoring protocols on two canonical examples with both firmly rooted in SPS.

4.1. Example 1

The first canonical example considered comprises the following set of biochemical reactions:

$$\varnothing \xrightarrow{\frac{\alpha\beta^i}{(1+\beta)^{i+1}}} iP, \qquad P \xrightarrow{d} \varnothing, \tag{14}$$

where P stands for a protein of interest. The first reaction in (14) in fact represent a group of reactions, and means that the protein is produced in bursts, whose size i conforms to a geometric distribution parametrized by $1/(1+\beta)$, while the second stands for the degradation of protein or its loss of functionality. The system (14) is known as *bursty* system in literature, and was found to adequately characterize the stochastic dynamics of most genes in mammalian or human cells [40]. The burst frequency α is selected as 0.0282 min^{-1}, the mean burst size β is 3.46, and the degradation rate constant d is 0.01 min^{-1}. These kinetic parameters correspond to those associated with gene *Nanog* in mouse embryonic stem cells [40].

We first simulate the system (14) by means of SSA for 1, 10, 100 and 1000 realizations and each for two sets. In either set, the protein numbers are averaged for all realizations at each time point. The results in Figure 4 show that the single-realization data is remarkably noisy and thus poses challenges for establishing a robust process monitoring model (see Figure 4a). The distribution of protein numbers at

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

$t = 400$ min is shown in Figure 4e, and is indeed a negative binomial distribution [40]. The fluctuations are substantially attenuated as the number of averaged realizations increase (see Figures. 4b, 4c, 4d). It suggests that ensemble method is a simple but effective approach for data curation. However, precautions should be taken for large number of realizations for two reasons: (i) the anomaly may be averaged out so that its detection becomes challenging; (ii) the large number of realizations is tantamount to that of cells, whose sampling may be difficult in practice. Here we choose the number to be 100.

Next we show that the mean is not adequate for process monitoring on SPS. To this end, we simulate a fault by decreasing α to a third ($\alpha = 0.0094$ min^{-1}) and increasing β by three times ($\beta = 10.38$) at time $t = 401$. First, we trained a DKNN model with mean values at two successive time points as input and output. The training dataset comprises 2000 data points collected at time $t = 400$ and $t = 401$ corresponding to the steady state (see Figure 4e), while a test set is of size 100, on which an accuracy test is performed. The accuracy of the trained DKNN model is shown in Figure 5a. The predictions are distributed close to the line $y = x$, indicating that these predictions are accurate. By means of SVDD, a radius for anomaly detection is computed and shown as red line in Figure 5b. Most of the residues (~ 90%) are contained within this radius. Within the help of the DKNN model and the radius, we perform the test to detect the aforementioned anomaly occurring at time $t = 401$. The F-scores averaged over 20 independent ensemble samples at 4 different time points are presented in Table 1. It clearly shows that the detection accuracy is low and cannot be improved over time, thereby solidly advocating our statement that mean value is not sufficient for SPS process monitoring. The unsatisfactory result is attributed to the anomaly we specially chose. As stated previously, the steady state distribution of the system (14) is negative binomial parametrized as $\mathrm{NB}(\frac{\alpha}{d}, \frac{1}{1+\beta})$ with the mean being $\alpha\beta/d$. The mean is not altered for the specially selected anomaly. Hence, it is a vivid example showing that the mean value is not adequate to characterize the SPS dynamics and calls for high-order moments. It is also evidenced by Figure 6a that the difference between the faulted and normal trajectories can hardly be discerned, whereas Figures. 6b, 6c, 6d show that high-order moments are much more sensitive to the anomalies.

Given the observation, it is necessary to incorporate high-order moments in datasets for SPS anomaly detection. As such, we create another three pairs of training and test datasets, and each has the moments up to order 2, 3 and 4 respectively. The methods of moments calculation are stated in Appendix 6.1. After training DKNN models on the three training datasets, three independent accuracy tests on the corresponding test dataset are carried out, and the results are shown in Figure 7. It shows that the accuracy R^2 degrades as the order of moment of prediction interest increases as expected. Generally, the fluctuations in higher-order moments are more intense than that in lower-order moments.

Subsequently, we use the three well-trained DKNN models to detect the aforementioned anomaly. It shows in Figure 8 that the detection becomes more accurate as the anomaly effects accumulate in time. Besides, the models based on moments of order 3 and order 4 outperform that of order 2, while the performance of the former two are comparable. Hence, it is concluded that the combination of moments of order up to 3 probably suits best for DKNN model performing anomaly detection in SPS.

Furthermore, we compare the DKNN model and DMD model both trained on the dataset containing moments of order up to 3. The accuracy comparison is summarized in Figure 9a. It shows that DKNN outperforms DMD on the predictions of all the moments. However, the DKNN's advantage is mitigating as the stochasticity gets stronger in higher-order moment data. As for anomaly detection, the F-scores of DKNN are higher than that of DMD by 15% ~ 50% (see Figure 9b).

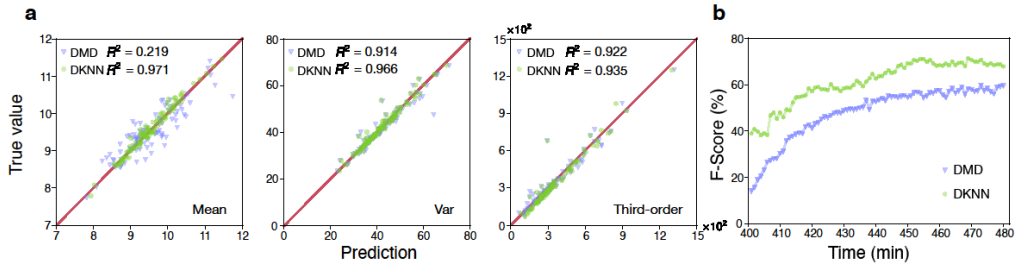

Figure 9: Comparison of DKNN and DMD on (a) prediction accuracy and (b) anomaly detection of Example 1.

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

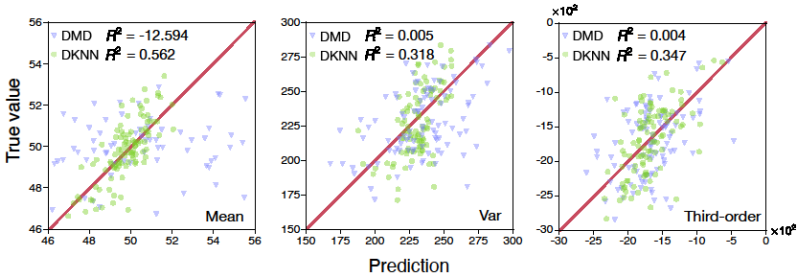

Figure 10: Comparison of DKNN and DMD on prediction accuracy of Example 2.

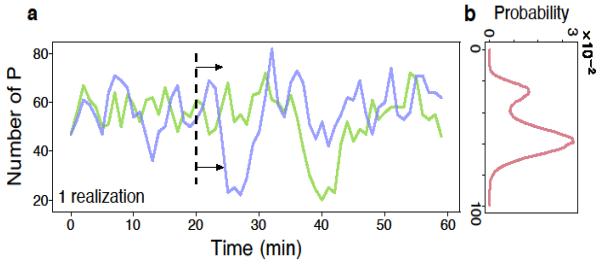

Figure 11: Stochastic simulations for Example 2.

Table 2: Comparision of DKNN and DMD on detection of anomalies case 1 & case 2 in example 2

Case	Case 1		Case 2	
Method	DKNN	DMD	DKNN	DMD
Time (min)	21	21	21	21
F-score (%)	92.44	66.67	23.91	16.39

Table 3: DKNN technical details

Case	Case 1		Case 2	
Method	DKNN	DMD	DKNN	DMD
Time (min)	21	21	21	21
F-score (%)	92.44	66.67	23.91	16.39

4.2. Example 2

Next we consider a more complicated example, which is of great biological interest as well. The SPS consists of five biochemical reactions:

$$\begin{aligned} &\mathrm{G} \xrightarrow{\rho_1} G+P, \quad G^* \xrightarrow{\rho_2} G^*+P, \quad P \xrightarrow{d} \varnothing, \\ &G \xrightarrow{\sigma_b} G^*, \quad G^* \xrightarrow{\sigma_u} G. \end{aligned} \tag{15}$$

The system as a whole is named *telegraph* model, which is a renowned model for gene expression in [44]. The symbols G and G^* stand for two gene states that are actively expressing proteins (usually referred as ON state) and less active (referred as OFF state with leakage). The first two reactions in (15) mean protein P being expressed, the third stands for protein degradation, the fourth and fifth mean that the gene is hopping between ON and OFF states. The kinetic parameters we use here are:$\rho_1 = 60\ \mathrm{min}^{-1}$, $\rho_2 = 25\ \mathrm{min}^{-1}$, $d = 1\ \mathrm{min}^{-1}$, $\sigma_b = 0.1\ \mathrm{min}^{-1}$, $\sigma_u = 0.25\ \mathrm{min}^{-1}$. By using SSA, we collect data at time $t = 20,\ 21$ min to create a training dataset of size 2000 and a test dataset of size 100. Both datasets contain the moments of order up to 3. By training DKNN and DMD model on the training dataset and comparing both on the test dataset, it is found in Figure 10 that DKNN is remarkably better than DMD for predicting all the moments, despite a loss in accuracy compared to the result of Example 1. However, it is with expectation, since the distribution for the kinetic parameters selected is bimodal suggesting the protein number is fluctuating between two disparate levels (see Figure 11). In the following, we further compare both models on detecting two different types of anomalies.

4.2.1. Case 1

The rate ρ_1 is changed to 40 at time $t = 21$ min, which corresponds to gene expression process of state ON changed. Based on the yielded models and the associated residues, SVDD computes the radii of

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

90% confidence interval for anomaly detection. The detection result is reported in Table 2, where the F-scores strongly support the superiority of DKNN.

4.2.2. Case 2

The rate σ_u is changed to 0.1 at time $t = 21$ min, which corresponds the gene is more often switching to OFF state. By applying the same process monitoring protocol again, the results in Table 2 again confirms DKNN's supremacy against DMD. However, the F-scores are lower than that of Case 1. It may be related to that Case 2 corresponds to a perturbation on the upstream of gene expression, while Case 1 corresponds to the downstream. The upstream perturbation may be buffered by a multitude of downstream processes, and thus becomes more challenging to detect. Nevertheless, Case 2 provides an excellent arena for benchmarking various process monitoring methods.

5. Conclusions

In this paper, we discuss the process monitoring for SPS and develop an integrated method of Koopman operator theory and deep neural network to solve it. The method uses a deep autoencoder structure to establish a bijective mapping between original space and a high-dimensional linear space, where the Koopman operator operates. An anomaly detection threshold is computed by SVDD on the basis of unmodeled residues. It is also argued that given the novel type of stochasticity—intrinsic noise, the SPS in the form of biochemical systems simulated by SSA can serve as an excellent arena for benchmarking various process monitoring methods. As SPS data is remarkably noisy, we propose to use ensemble method to tackle it and conclude that high-order moments have to be incorporated for robustness.

6. Appendix

6.1. Moment calculation

The moments in data are calculated as central moments

$$S^k = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^k, \tag{16}$$

where n is the number of samples, X_i stands for the value of sample at a certain time, and $\overline{X}$ is the mean of sample.

6.2. Moment normalization

The moments in the normal case is normalized by the min-max method as follows

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{17}$$

where X is the raw moment data, X_{norm} stands for the normalized moment, and X_{min}, X_{max} stand for the minimum and maximum of the raw data. The moments in the faulted case are normalized as per the minimum X_{min} and maximum X_{max} of the normal case.

6.3. Neural network details

The Koopman operator is implemented as a linear network. All the technical details of DKNN including network structure and hyperparameters are summarized in Table 3. All the weights of neural network are initialized as per a truncated normal distribution $\mathcal{N}(0, 0.1)$, while the biases are set to 0. The training optimizer is Adam with a learning rate equal to 0.001.

References

[1] Wang, G., Haringa, C., Noorman, H., Chu, J., and Zhuang, Y. (2020). Developing a Computational Framework to Advance Bioprocess Scale-Up. Trends in Biotechnology, 38(8), 846-856.
[2] Lu, J., Cao, Z., Zhao, C., and Gao, F. (2019). 110th Anniversary: An Overview on Learning-Based Model Predictive Control for Batch Processes. Industrial & Engineering Chemistry Research, 58(37), 17164-17173.
[3] Jiang, Q., Wang, Z., Yan, S., and Cao, Z. (2022). Data-Driven Soft Sensing for Batch Processes

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

Using Neural Network-Based Deep Quality-Relevant Representation Learning. IEEE Transactions on Artificial Intelligence.
[4] Cao, Z., Yu, J., Wang, W., Lu, H., Xia, X., Xu, H., and Zhang, L. (2020). Multi-Scale Data-Driven Engineering for Biosynthetic Titer Improvement. Current Opinion in Biotechnology, 65, 205-212.
[5] Gao, J., Feng, E., and Zhang, W. (2022). Modeling and Parameter Identification of Microbial Batch Fermentation under Environmental Disturbances. Applied Mathematical Modelling, 108, 205-219.
[6] Soukoulis, C., Panagiotidis, P., Koureli, R., and Tzia, C. (2007). Industrial Yogurt Manufacture: Monitoring of Fermentation Process and Improvement of Final Product Quality. Journal of dairy science, 90(6), 2641-2654.
[7] Sriramula, S., and Chryssanthopoulos, M. K. (2009). Quantification of Uncertainty Modelling in Stochastic Analysis of FRP Composites. Composites Part A: Applied Science and Manufacturing, 40(11), 1673-1684.
[8] Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N. (2008). MPCA: Multilinear Principal Component Analysis of Tensor Objects. IEEE transactions on Neural Networks, 19(1), 18-39.
[9] Lee, J. M., Yoo, C., and Lee, I. B. (2004). Fault Detection of Batch Processes Using Multiway Kernel Principal Component Analysis. Computers & Chemical Engineering, 28(9), 1837-1847.
[10] Jia, Z. Y., Wang, P., and Gao, X. J. (2012). Process Monitoring and Fault Diagnosis of Penicillin Fermentation Based on Improved MICA. Advanced Materials Research, 591, 1783-1788.
[11] Zhao, C., Gao, F., and Wang, F. (2009). Nonlinear Batch Process Monitoring Using Phase-Based Kernel-Independent Component Analysis−Principal Component Analysis (KICA−PCA). Industrial & Engineering Chemistry Research, 48(20), 9163-9174.
[12] Peng, C., Chunhao, D., and Qiankun, Z. (2020). Fault Diagnosis of Microbial Pharmaceutical Fermentation Process with Non-Gaussian and Nonlinear Coexistence. Chemometrics and Intelligent Laboratory Systems, 199, 103931.
[13] Yang, C., and Hou, J. (2016). Fed-Batch Fermentation Penicillin Process Fault Diagnosis and Detection Based on Support Vector Machine. Neurocomputing, 190, 117-123.
[14] Ding, J., Cao, Y., Mpofu, E., and Shi, Z. (2012). A Hybrid Support Vector Machine and Fuzzy Reasoning Based Fault Diagnosis and Rescue System for Stable Glutamate Fermentation. Chemical Engineering Research and Design, 90(9), 1197-1207.
[15] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. Neural Networks, 2(5), 359-366.
[16] Shimizu, H., Yasuoka, K., Uchiyama, K., and Shioya, S. (1997). On-Line Fault Diagnosis for Optimal Rice a-Amylase Production Process of a Temperature-Sensitive Mutant of Saccharomyces Cerevisiae by an Autoassociative Neural Network. Journal of Fermentation and Bioengineering, 83(5), 435-442.
[17] Lopes, J. A., and Menezes, J. C. (2004). Multivariate Monitoring of Fermentation Processes with Non-Linear Modelling Methods. Analytica Chimica Acta, 515(1), 101-108.
[18] Yu, J., Zhang, C., and Wang, S. (2021). Multichannel One-Dimensional Convolutional Neural Network-Based Feature Learning for Fault Diagnosis of Industrial Processes. Neural Computing and Applications, 33, 3085-3104.
[19] Chen, S., Yu, J., and Wang, S. (2020). One-Dimensional Convolutional Auto-Encoder-Based Feature Learning for Fault Diagnosis of Multivariate Processes. Journal of Process Control, 87, 54-67.
[20] Peng, C., Lu, R., Kang, O., and Kai, W. (2020). Batch Process Fault Detection for Multi-Stage Broad Learning System. Neural Networks, 129, 298-312.
[21] Chen, H., Liu, Z., Alippi, C., Huang, B., and Liu, D. (2022). Explainable Intelligent Fault Diagnosis for Nonlinear Dynamic Systems: From Unsupervised to Supervised Learning. IEEE Transactions on Neural Networks and Learning Systems.
[22] Chen, H., Chai, Z., Dogru, O., Jiang, B., and Huang, B. (2021). Data-Driven Designs of Fault Detection Systems Via Neural Network-Aided Learning. IEEE Transactions on Neural Networks and Learning Systems, 33(10), 5694-5705.
[23] Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. Physica D: Nonlinear Phenomena, 404, 132306.
[24] Zhang, M., Li, X., and Wang, R. (2021). Incipient Fault Diagnosis of Batch Process Based on Deep Time Series Feature Extraction. Arabian Journal for Science and Engineering, 1-12.
[25] Ren, J., and Ni, D. (2020). A Batch-Wise LSTM-Encoder Decoder Network for Batch Process Monitoring. Chemical Engineering Research and Design, 164, 102-112.
[26] Koopman, B. O. (1931). Hamiltonian Systems and Transformation in Hilbert Space. Proceedings of the National Academy of Sciences, 17(5), 315-318.
[27] Brunton, S. L. (2019). Notes on Koopman Operator Theory. Universität Von Washington, Department of Mechanical Engineering, Zugriff, 30.

ISSN 2616-5767 Vol.6, Issue 9: 31-42, DOI: 10.25236/AJETS.2023.060906

[28] Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S. (2009). Spectral Analysis of Nonlinear Flows. Journal of Fluid Mechanics, 641, 115-127.
[29] Schmid, P. J. (2010). Dynamic Mode Decomposition of Numerical and Experimental Data. Journal of Fluid Mechanics, 656, 5-28.
[30] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W. (2015). A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. Journal of Nonlinear Science, 25, 1307-1346.
[31] Korda, M., and Mezić, I. (2018). Linear Predictors for Nonlinear Dynamical Systems: Koopman Operator Meets Model Predictive Control. Automatica, 93, 149-160.
[32] Brunton, S. L., Brunton, B. W., Proctor, J. L., and Kutz, J. N. (2016). Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control. Plos One, 11(2), e0150171.
[33] Lusch, B., Kutz, J. N., and Brunton, S. L. (2018). Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics. Nature Communications, 9(1), 4950.
[34] Yeung, E., Kundu, S., and Hodas, N. (2019, July). Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems. In 2019 American Control Conference (ACC) (pp. 4832-4839). IEEE.
[35] Dubey, R., Samantaray, S. R., Panigrahi, B. K., and Venkoparao, V. G. (2016). Koopman Analysis Based Wide-Area Back-Up Protection and Faulted Line Identification for Series-Compensated Power Network. IEEE Systems Journal, 12(3), 2634-2644.
[36] Dang, Z., Lv, Y., Li, Y., and Wei, G. (2018). Improved Dynamic Mode Decomposition and Its Application to Fault Diagnosis of Rolling Bearing. Sensors, 18(6), 1972.
[37] Cheng, C., Ding, J., and Zhang, Y. (2020). A Koopman Operator Approach for Machinery Health Monitoring and Prediction with Noisy and Low-Dimensional Industrial Time Series. Neurocomputing, 406, 204-214.
[38] Liu, B., Xiao, Y., Cao, L., Hao, Z., and Deng, F. (2013). Svdd-Based Outlier Detection on Uncertain Data. Knowledge and Information Systems, 34, 597-618.
[39] Larsson, A. J., Johnsson, P., Hagemann-Jensen, M., Hartmanis, L., Faridani, O. R., Reinius, B., and Sandberg, R. (2019). Genomic Encoding of Transcriptional Burst Kinetics. Nature, 565(7738), 251-254.
[40] Cao, Z., and Grima, R. (2020). Analytical Distributions for Detailed Models of Stochastic Gene Expression in Eukaryotic Cells. Proceedings of the National Academy of Sciences, 117(9), 4682-4692.
[41] Gillespie, D. T. (1977). Exact Stochastic Simulation of Coupled Chemical Reactions. The Journal of Physical Chemistry, 81(25), 2340-2361.
[42] Fu, X., Zhou, X., Gu, D., Cao, Z., and Grima, R. (2022). DelaySSAToolkit. jl: Stochastic Simulation of Reaction Systems with Time Delays in Julia. Bioinformatics, 38(17), 4243-4245.
[43] Brunton, S. L., Budišić, M., Kaiser, E., and Kutz, J. N. (2021). Modern Koopman Theory for Dynamical Systems. arXiv preprint arXiv:2102.12086.
[44] Cao, Z., and Grima, R. (2018). Linear Mapping Approximation of Gene Regulatory Networks with Stochastic Dynamics. Nature Communications, 9(1), 3305.