

Article

Latent-Space Dynamics for Prediction and Fault Detection in Geothermal Power Plant Operations

Yingxiang Liu ¹, Wei Ling ² , Robert Young ², Jalal Zia ³, Trenton T. Cladouhos ³ and Behnam Jafarpour ^{2,*}

¹ Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089, USA; yingxian@usc.edu

² Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA 90089, USA; wling@usc.edu (W.L.); youngrob@usc.edu (R.Y.)

³ Cyrq Energy Inc., Salt Lake City, UT 84101, USA; jalal.zia@cyrqenergy.com (J.Z.); trenton.cladouhos@cyrqenergy.com (T.T.C.)

* Correspondence: behnam.jafarpour@usc.edu; Tel.: +1-(213)-740-2228

Abstract: This paper presents a latent-space dynamic neural network (LSDNN) model for the multi-step-ahead prediction and fault detection of a geothermal power plant's operation. The model was trained to learn the dynamics of the power generation process from multivariate time-series data and the effects of exogenous variables, such as control adjustment and ambient temperature. In the LSDNN model, an encoder–decoder architecture was designed to capture cross-correlation among different measured variables. In addition, a latent space dynamic structure was proposed to propagate the dynamics in the latent space to enable prediction. The prediction power of the LSDNN was utilized for monitoring a geothermal power plant and detecting abnormal events. The model was integrated with principal component analysis (PCA)-based process monitoring techniques to develop a fault-detection procedure. The performance of the proposed LSDNN model and fault detection approach was demonstrated using field data collected from a geothermal power plant.

Keywords: fault detection; neural network; latent space dynamics; geothermal operations; power plant



Citation: Liu, Y.; Ling, W.; Young, R.; Zia, J.; Cladouhos, T.T.; Jafarpour, B. Latent-Space Dynamics for Prediction and Fault Detection in Geothermal Power Plant Operations. *Energies* **2022**, *15*, 2555. <https://doi.org/10.3390/en15072555>

Academic Editors: Maruti Kumar Mudunuru, Kalyana Babu Nakshatrala and Zhao Hao

Received: 17 February 2022

Accepted: 28 March 2022

Published: 31 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As transitions from fossil fuel-dependent to renewable sources of energy gain traction, technologies are being developed to improve the efficiency of energy production by geothermal power plants [1]. With the increase in the utilization of geothermal energy, it is projected that, by 2050, geothermal power plants will generate 8.3% of the world's power and will serve 17% of the world population [2]. Geothermal power plants recover heat energy from hot underground rocks and convert it to electrical energy. The reliability and efficiency of operating geothermal power plants can be improved by optimizing the design of the power plants and by monitoring and controlling their operations. Modeling and predicting a power plant's performance under different operating conditions are important enabling tools for controlling and optimizing power plant design and operations. Physics-based simulation models have traditionally been used to model the dynamics of geothermal power plants [3,4]. However, these models are typically complex to construct, since significant efforts and in-depth knowledge and expertise are required to model the behavior of the underlying components and their interactions accurately [5]. In addition, since the configurations and characteristics of geothermal power plants depend on the field conditions and performance requirements [6], many uncertain parameters are involved when building simulation models for different power plants.

A good alternative to physics-based simulation models is data-driven modeling, where the model is trained using existing monitoring and performance data to learn the statistical patterns and relationships among different components to predict their future behavior and performance. Recently, the success of deep learning has renewed interest in the use of

artificial neural networks (ANN) as data-driven proxy models for prediction in physical systems. Various neural network architectures have been applied to model and predict the performance of different energy systems. Multilayer perceptron (MLP) is used to forecast the output power of a photovoltaic plant based on 48 h ahead weather forecasts [7]. The autoencoder structure has been combined with Long Short-Term Memory (LSTM) to forecast the energy output of 21 solar power plants [8]. To model a large-scale supercritical boiler plant, individual recurrent neural network (RNN) models have been built for different subsystems and combined based on the subsystems' input and output relationships [9]. Convolutional neural networks (CNNs) that offer significant efficiency in processing images have also been applied to two-channel two-dimensional images containing information related to the states and changes of states in a nuclear power plant for the classification of abnormal events [10]. In the geothermal domain, MLP has been used to aid the design and optimization of binary geothermal power plants by predicting the generated power of the system and the required circulation pump power [11]. In addition, a deep neural network model has been used to predict geothermal reservoir temperatures based on hydrogeochemical parameters [12]. In [13], LSTM and MLP were used to predict the productivity of a multilateral-well geothermal system, where the LSTM was used to learn the trend in historical productions, and the MLP was used to make predictions based on the output of the LSTM and the constraints of reservoir properties.

Although there have been many applications of ANN related to geothermal power generation, they mainly focused on subsurface processes. For the data modeling of geothermal power plants, studies were focused on the steady-state surface processes. The drawback of steady-state modeling is that it fails to capture the autocorrelation in time-series data and the dynamics of the power generation process affected by ambient disturbances and control adjustments. In addition, the underlying physics in the power generation process serve as a constraint that relates measured quantities, such as temperature and pressure. As a result, the existing relationships in the data imply that the main variations can be captured and represented in a low-dimensional latent space. To achieve latent representation, the autoencoder (AE) structure has been used for applications, such as dimensionality reduction [14] and system identification [15]. To capture the autocorrelations in the data and to describe the nonlinear dynamics of a system, neural networks with a single hidden layer have been used [16]. In addition, recurrent neural networks that use nonlinear autoregressive model with exogenous inputs (NARX) have been implemented with a multilayer perceptron. It was shown that the resulting network is often better at discovering long-term dependencies than conventional recurrent neural networks, mainly due to the delays in the network acting as jump-ahead connections during training [17]. In [18], the authors showed that, by using output feedback, the NARX neural network was able to predict complex time series.

In geothermal power plants, faults, such as working fluid leakage, ingress of non-condensable gases into working gas, and production pump failure, can lead to poor performance or even catastrophic failure. Due to the time scale of the faults, some of them are difficult to be spotted by the plant operators. In these cases, statistical process monitoring (SPM) techniques can help by constantly monitoring data. SPM has been used in many industrial processes, such as chemicals and semiconductor manufacturing [19], but it has not been used for applications related to geothermal power generation. Fault detection is an essential part of SPM. It is used to detect abnormal events within the process, which is crucial for maintaining normal operating conditions and preventing catastrophic failures. Principal component analysis (PCA) has been widely used for static process monitoring [19–21]. However, as PCA assumes that the data samples are independent in time, directly applying PCA-based monitoring techniques to time-series data may lead to poor fault detection results due to the presence of autocorrelation. To deal with this issue, fault-detection procedures were proposed in [22,23], in which the authors first extracted the dynamics from multivariate time-series data, which removed the autocorrelations within the data. As a result, the residuals only contain static variations, which can be modeled using the PCA and lend themselves to detect faults.

In this paper, we present a novel latent-space dynamic neural network (LSDNN) architecture to exploit the characteristics of the data collected from geothermal power plants. The LSDNN model combines the advantages of autoencoder structure and the NARX neural network model to effectively capture the cross-correlation and autocorrelation from multivariate time-series measurements. The dynamic model was trained to learn the interactions and statistical relationships between different measured and exogenous variables in the power generation process. The trained neural network was then used to make multi-step-ahead predictions about the performance of each component in the power plant. Furthermore, the predictions were used to formulate a fault detection algorithm for the plant, where the autocorrelations were first removed through prediction, and static process monitoring techniques were applied to monitor the prediction errors for detecting abnormal events in the power generation process. To the best of our knowledge, this is the first work to use neural network predictions and SPM to perform fault detection in geothermal power plants.

The remainder of this article is organized as follows. Section 2 presents the structure of the proposed LSDNN model. In Section 3, the prediction results of the LSDNN on field data collected from a geothermal power plant are shown. The prediction performances are compared with another commonly used RNN encoder–decoder neural network model. In addition, some interpretations of the LSDNN model are represented. In Section 4, we propose a fault-detection procedure using the LSDNN model to detect abnormal events in a geothermal power plant. The performance of the fault-detection procedure was tested using real data collected from a power-generation unit and data from a production well. Section 5 presents the discussion and conclusion of the paper.

2. Methodology

In this section, we discuss the LSDNN model. Figure 1 shows the schematic of the proposed LSDNN architecture, which consists of two parts: an encoder–decoder structure for mapping the original data to latent variables (and vice versa) and a latent-space dynamic model. In the LSDNN model, the encoder–decoder structure captures cross-correlations among measurements and enables a latent-space representation. Let $x_t \in \mathbb{R}^m$ denote a vector of m measured variables at time t , $z_t \in \mathbb{R}^h$ denote a vector of h dimensional latent variables with $h < m$ at time t , and $u_t \in \mathbb{R}^n$ denote a vector of exogenous variables such as control adjustments and ambient temperature at time t . In the beginning, the measurement encoder brings the past measured variables into the h -dimensional latent space in which the latent dynamics are propagated, and predictions are made. In addition, since the dimensionality of the exogenous variables may be different from h , an input encoder is used to map the exogenous variables u to a vector $u' \in \mathbb{R}^h$.

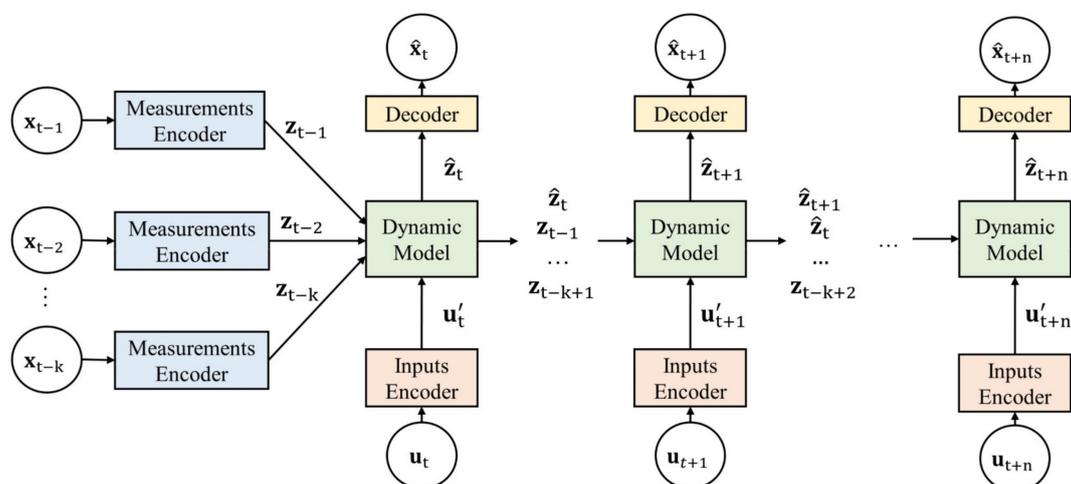


Figure 1. Latent-space dynamic neural network model structure.

Once the predictions are made through the latent-space dynamic model, the decoder maps them back to the original data space. For each time step, the model uses the same measurement encoder, input encoder, and decoder. In our model, the encoders are represented using dense neural network layers followed by a hyperbolic tangent activation function as follows:

$$\mathbf{z}_t = f^{me}(\mathbf{x}_t) = \tanh(\mathbf{W}^{me}\mathbf{x}_t + \mathbf{b}^{me}) \quad (1)$$

$$\mathbf{u}'_t = f^{ie}(\mathbf{u}_t) = \tanh(\mathbf{W}^{ie}\mathbf{u}_t + \mathbf{b}^{ie}) \quad (2)$$

where $\mathbf{W}^{me} \in \mathbb{R}^{h \times m}$, $\mathbf{b}^{me} \in \mathbb{R}^h$, $\mathbf{W}^{ie} \in \mathbb{R}^{h \times n}$, and $\mathbf{b}^{ie} \in \mathbb{R}^h$. The decoder is also a dense layer that decodes the predicted latent state $\hat{\mathbf{z}}_t$ at time t to the predicted measured variables $\hat{\mathbf{x}}_t$ at time t . As a result, the decoder can be represented as

$$\hat{\mathbf{x}}_t = \mathbf{W}^{de}\hat{\mathbf{z}}_t + \mathbf{b}^{de} \quad (3)$$

where $\mathbf{W}^{de} \in \mathbb{R}^{m \times h}$ and $\mathbf{b}^{de} \in \mathbb{R}^m$.

To capture the autocorrelation, describe the evolution of the latent states, and account for the effect of exogenous variables on the latent dynamics, a latent-space dynamic model was used in the LSDNN model. The propagation of latent-space dynamics is represented using a NARX, which has the form

$$\hat{\mathbf{z}}_t = f(\mathbf{z}_{t-1}, \dots, \mathbf{z}_{t-D_z}, \mathbf{u}'_t, \dots, \mathbf{u}'_{t-D_u}) \quad (4)$$

where $\hat{\mathbf{z}}_t$ is the one-step-ahead prediction at time t , D_z is the order of the latent state, and D_u is the input order. To perform multi-step-ahead prediction in the latent space given the trajectories of future exogenous variables, a recursive prediction strategy was used [24], where the one-step-ahead prediction from the dynamic model was treated as a true latent state and fed into the dynamic model recursively.

In this study, we considered a NARX network with an input order of k and set the D_z to be k . In the neural network representations of NARX, the nonlinear function f is approximated using a dense layer, where the input layer receives the concatenated vector of past measured variables and exogenous variables. As a result, with this representation, the number of weights of the dense layer is $(kh + (k+1)h)h$. If the latent dimension h is large, the large number of weights may lead to overfitting issues and take longer to train the neural network. We decided to alleviate this problem by taking advantage of the characteristics of the time-series data collected from the geothermal power plant. The exogenous variables were highly correlated with the measured variables in the data, which meant that they were also correlated with the series of encoded latent states. To take advantage of this characteristic, we separated the effect of exogenous variables on the latent states and the residual dynamics of the latent states, which made the model more interpretable and less complex. This can be achieved by first forming residual states $\mathbf{z}' \in \mathbb{R}^h$ by removing \mathbf{u}' from \mathbf{z} , then concatenating past k residual states as an input to the dense layer. After the dense layer produces a one-step-ahead prediction, the effect of the exogenous variables at the new time step is added back. Following this approach, the dense layer is used to approximate an autoregressive model of \mathbf{z}' , representing the evolution of the residual dynamics that is not accounted for by the encoded exogenous variables \mathbf{u}' . As a result, the number of weights in the dense layer is reduced to $(kh)h$. Furthermore, since the effects of exogenous variables are already included in the encoded measured variables, the effect of \mathbf{u}' must be removed from the encoded variables \mathbf{z} prior to being fed to the dynamic model. As a result, the model is able to properly separate the effects of the exogenous variables and the residual dynamics.

In a geothermal power plant, there may be multiple power generation units that share the brine produced by the production wells. Shutting down one of the production wells or one power generation unit may cause sudden changes in brine flow that affect other measured variables. In addition, these changes affect the exogenous variables, such as

control setpoints, accordingly. As a result, in the time-series data collected from geothermal power plants, sudden significant changes may occur in some measured variables and exogenous variables. For our proposed dynamic model, oscillations in prediction can occur when sudden changes are made due to the autoregressive nature of the dynamic model. Furthermore, when significant changes happen, the next time step prediction may not require all of the past k latent states, which act as the memory associated with the past dynamics. Hence, we designed an attention mechanism to allow the model to make adjustments, forget some of the past stored latent states, and focus on relevant latent states based on the changes in exogenous variables. The attention mechanism has been an essential part of sequence modeling and machine translation, where it has been used to adaptively select a subset of hidden states while decoding the translation [25,26]. The attention mechanism has also been applied for time-series-forecasting tasks. An attention mechanism is used to combine hidden states to model nonseasonal dependency [27]. In [28], an input attention mechanism was designed to extract relevant input features at each timestep based on the previous encoder's hidden states. In addition, a temporal attention mechanism was used to select relevant encoder hidden states across all time steps. The attention mechanism can be expressed using dot product, similarity functions, or a multilayer perceptron. In this study, we used a dense layer to implement the attention mechanism. At the prediction step t , the input to the dense layer is the concatenated vector of \mathbf{u}_t' and \mathbf{u}_{t-1}' , which are the encoded exogenous variables. The output of the dense layer is a k dimensional vector $[a'_{t-1}, \dots, a'_{t-k}]^T \in \mathbb{R}^k$ in which each element corresponds to a residual state vector. A softmax function is then applied to the output to produce a vector of attention scores $[a_{t-1}, \dots, a_{t-k}]^T \in \mathbb{R}^k$. Finally, the adjusted residual states are $a_{t-i}z'_{t-i} \in \mathbb{R}^h$ for $i = 1, 2, \dots, k$. As a result, the attention mechanism allows the dynamic model to scale the stored past k residual states based on the changes in encoded exogenous variables between two consecutive time steps. When the exogenous variables change suddenly, the model can focus on relevant past residual states for prediction.

Figure 2 shows the details of the latent-space dynamic model. Given the past k encoded measured variables and the encoded exogenous variables, the one-step-ahead prediction can be represented as

$$[z'_{t-1}, \dots, z'_{t-k}] = [z_{t-1}, \dots, z_{t-k}] - [\mathbf{u}'_{t-1}, \dots, \mathbf{u}'_{t-k}] \quad (5)$$

$$[a'_{t-1}, \dots, a'_{t-k}]^T = \tanh(\mathbf{W}^a [\mathbf{u}_t'; \mathbf{u}_{t-1}'] + \mathbf{b}^a) \quad (6)$$

$$a_{t-i} = \frac{\exp(a'_{t-i})}{\sum_{j=1}^k \exp(a'_{t-j})}, \quad i = 1, 2, \dots, k \quad (7)$$

$$\hat{z}'_t = \tanh(\mathbf{W}^d [a_{t-1}z'_{t-1}; \dots; a_{t-k}z'_{t-k}] + \mathbf{b}^d) \quad (8)$$

$$\hat{\mathbf{z}}_t = \hat{z}'_t + \mathbf{u}_t' \quad (9)$$

where $\mathbf{W}^a \in \mathbb{R}^{k \times 2h}$, $\mathbf{b}^a \in \mathbb{R}^k$, $\mathbf{W}^d \in \mathbb{R}^{h \times kh}$ and $\mathbf{b}^d \in \mathbb{R}^h$. Once the model generates the one-step-ahead prediction in latent space, the stored past k latent states are updated by removing the last latent state and adding the newly predicted latent state. Further predictions in the latent space can be made by recursively feeding the new prediction back into the model and updating the past latent states.

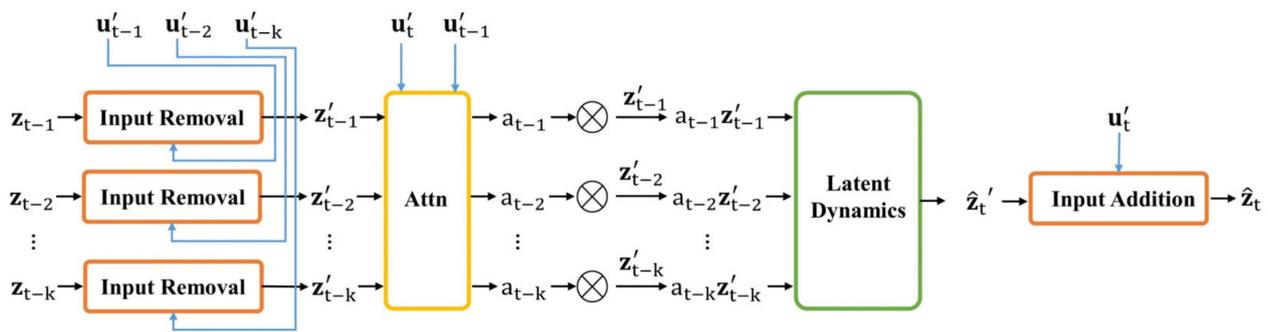


Figure 2. Details of the dynamic latent-space model for predicting in latent space.

3. Prediction Results

In this section, we first demonstrate the prediction capability of the proposed LSDNN using real data collected from a geothermal power plant and discuss the design of the LSDNN structure. The field data were collected from a binary cycle geothermal power plant operated by Cyrq Energy Inc. (Salt Lake City, UT, USA). The data consist of five years of hourly time-series measurements. After removing the missing data points and data collected during shutdown periods, around 20,500 data points were used for training and validation, and the remaining 4000 data points were used for testing. There were 19 measurements collected from the primary cycle, secondary cycle, and the turbine of the power generation unit. Among the 19 variables, we selected the brine outlet flow, the setpoint of the turbine inlet guide vane (IGV), and the R134a pump speed to be part of the exogenous variables for incorporating the changes in the operational settings. In addition, we used the ambient temperature as the fourth exogenous variable, since the efficiency of the plant is highly related to the ambient temperature because of the air-cooling system [4,29]. As a result, multi-step-ahead predictions were performed on the remaining 15 measured variables with given future operation settings and weather forecasts.

Before implementing the final LSDNN structure in PyTorch, sensitivity analysis was first performed to determine the structure of the LSDNN model. There were two hyperparameters: the latent space dimension h and the number of past data points used for predictions D_z . For each hyperparameter value, 10 models were trained, and the final prediction accuracy was evaluated based on the validation dataset using the average of the root mean squared errors. It can be observed from Figure 3 that, with a small latent space dimension, the model could not capture all the dynamical patterns in the data, which led to underfitting. On the other hand, if the latent states dimension was too large, the number of parameters in the LSDNN model also increased, leading to overfitting. The sensitivity analysis with respect to D_z showed that a small window size could not effectively encode the information from the past data points to initialize the latent state used for prediction, resulting in large prediction errors. In addition, beyond a window size of 12, further increasing the window size did not help in improving the prediction performance. As a result, after performing the sensitivity analysis, the latent space dimension h was selected to be 10 and the order D_z was determined to be 12, which meant that the model used information from the past 12 timesteps to make predictions. In addition, the prediction horizon N was selected to be 12 during training. After normalizing the measured variables and exogenous variables collected during normal operation between 0 and 1, sequences of data with a length of 24 timesteps were generated for training. The measured variables in the first 12 samples were used to initialize the encoded latent states, and 12 samples of the exogenous variables were also included. The last 12 samples corresponded to the future exogenous variables and measured variables. The model used the exogenous variables to make multi-step-ahead predictions, while the measured variables were used to calculate the prediction errors for backpropagation. We used 15,800 sequences for training

and 4000 sequences for validation. For the i th data sequence, we defined the objective function as

$$\min J_i = \frac{1}{N} \sum_{k=1}^N \|x_k - \hat{x}_k\|^2 + \frac{\lambda}{N} \sum_{k=1}^N \|z_k - \hat{z}_k\|^2 \tag{10}$$

where N is the length of the prediction horizon, x is the vector of the measured variables in the future, \hat{x} is the predicted measured variables, z is the future latent state, which is encoded from the future measured variables by the measurement encoder, \hat{z} is the predicted latent state from the latent-space dynamic model, and λ is a hyperparameter used to adjust the penalty on the second term. The first term in the objective function is the mean-square-error of the N -step-ahead prediction. The second term penalizes the derivation of the predicted latent states from the true latent states calculated using the measurements encoder. When only the first term of the objective function is used, \hat{x} is forced to stay close to x , while \hat{z} may be different from z , given \hat{x} is the output from the decoder with independent weights to be adjusted. As a result, the second term was used to separate the latent variables from the decoder so that the predicted latent states remained in the latent space, and the decoder was only to map the latent variables back to the original data space. During training, the parameters of the neural network were trained together through standard backpropagation using Adam optimizer [30] with a learning rate of 0.001 and batch size of 128. With the loss function defined for each data sequence in Equation (10), the overall objective function with M data sequences is defined as

$$\min J = \frac{1}{M} \sum_{k=1}^M J_k \tag{11}$$

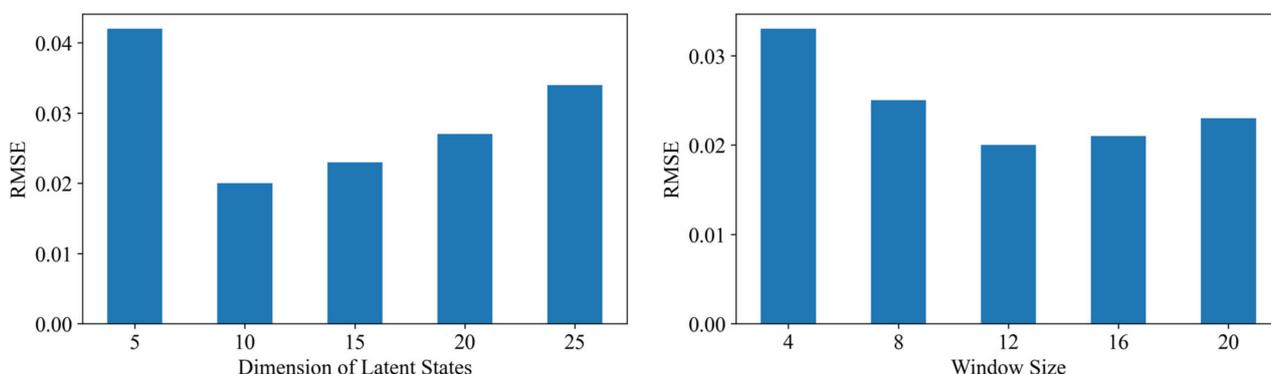


Figure 3. Sensitivity analysis for the LSDNN model with respect to the dimension of the latent states and the look-back window size.

Figure 4 shows the one-step-ahead prediction from the LSDNN on the testing dataset. For better presentation, we plotted the prediction of eight variables among the 15 predicted variables. Similar plots are shown in Figure 5 for 12-step-ahead prediction. The top eight subplots in each figure correspond to eight measured variables, and the bottom four subplots show the exogenous variables. To better display the 12-step-ahead prediction results, Figure 5 was generated in the following way: the model used 12 samples from the past to perform 12-step-ahead prediction. After 12 timesteps, the model received the real measurements from the past and used the most recent 12 samples for forward prediction. As a result, during the 12-step-ahead prediction, the model did not continuously integrate the incoming data at each timestep. It can be observed from the figures that the predictions from the model followed the general trend in the data closely and showed consistent responses to the changes in the operational settings and ambient conditions.

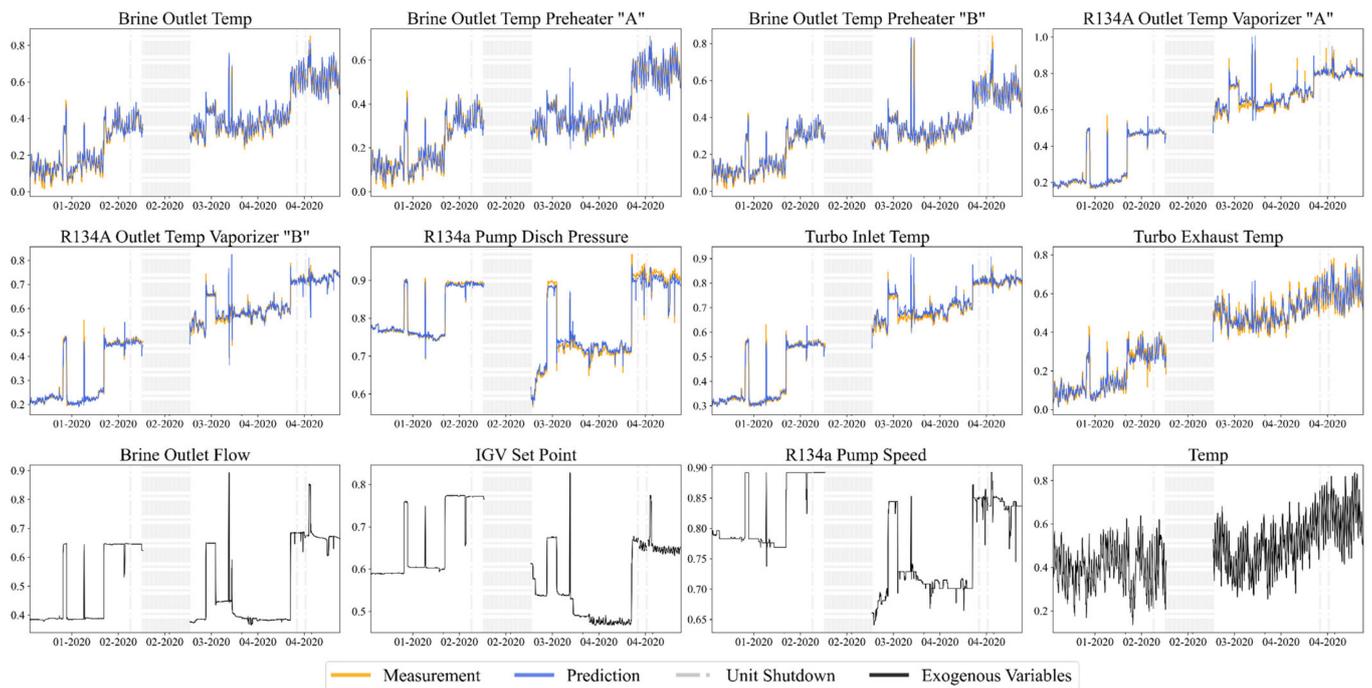


Figure 4. One-step-ahead prediction results of testing data. Top eight subplots: predicted measurements. Bottom four subplots: exogenous variables, including control inputs and ambient temperature.

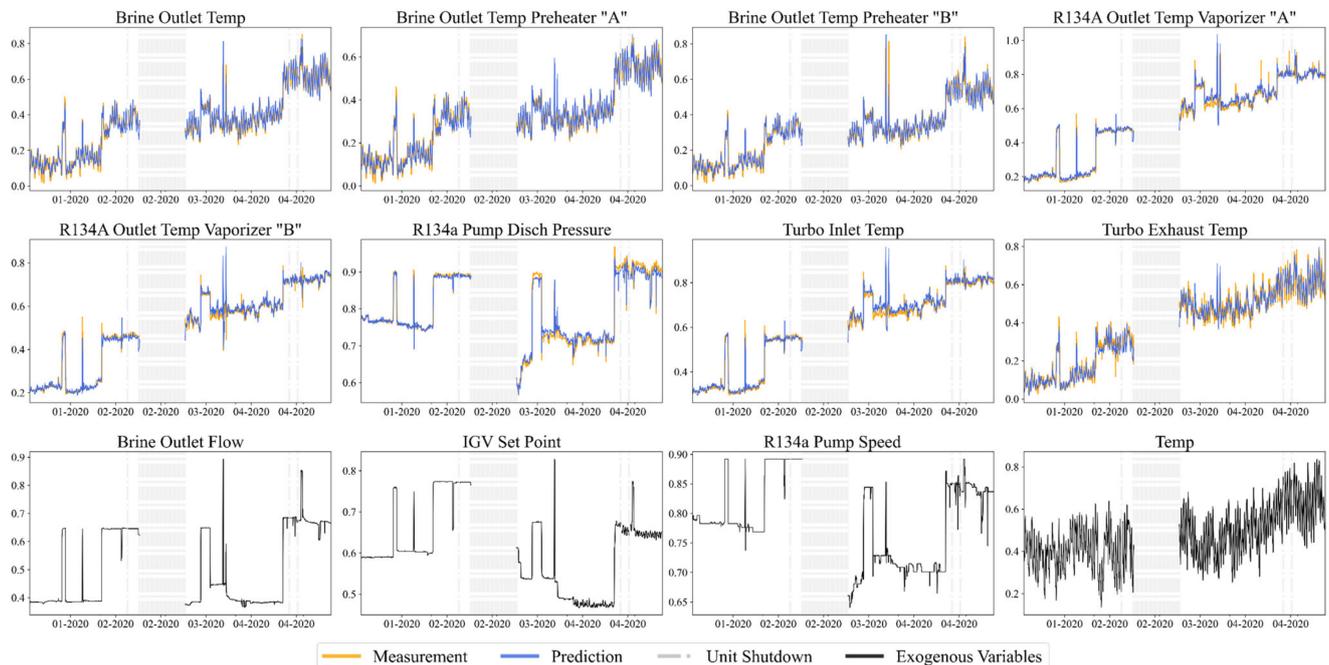


Figure 5. Twelve-step-ahead prediction results of testing data. Top eight subplots: predicted measurements. Bottom four subplots: exogenous variables, including control inputs and ambient temperature.

For comparison, we also trained the widely used RNN encoder–decoder model that was proposed in [25], where one LSTM is used to map the input sequence to a vector of fixed dimensionality and another LSTM is used to decode the vector for multi-step-ahead prediction. During training, the inputs to the encoder were the measured and exogenous variables in the past 12 timesteps, and the decoder received the future 12 timesteps of exogenous variables for prediction. The dimension of the hidden vector in the RNN encoder–decoder model was chosen to be 10. To measure the effectiveness of the LSDNN

and RNN encoder–decoder for time-series prediction, we considered both the root mean squared error (RMSE) and mean absolute percentage error (MAPE) as evaluation metrics. Let L be the length of the time series. For each measured variable, the RMSE is defined as $\sqrt{\frac{1}{L} \sum_{i=1}^L (x_i - \hat{x}_i)^2}$ and the MAPE is denoted as $\frac{1}{L} \sum_{i=1}^L \left| \frac{x_i - \hat{x}_i}{x_i} \right| \times 100\%$. We took the average of the prediction errors of 15 variables to obtain the overall prediction performance for each model. We trained the LSDNN and RNN encoder–decoder 10 times and reported their average performance and standard deviations on the test data. The prediction errors of the 1, 6, 12, and 24-step-ahead predictions are shown in Table 1. The prediction errors indicate that these two models have similar performances, and both can make accurate multi-step-ahead predictions. In addition, due to the recurrent nature of LSDNN and RNN encoder–decoder models and the similar number of parameters within the models, there was not much difference in the training time. With the selected structures and the same training data, the average time for training both models was around 50 min using one NVIDIA Tesla P100 GPU. However, compared to the RNN encoder–decoder model, the LSDNN model was more interpretable and flexible. It allowed the user to define the dynamics in a low-dimensional latent space and offered the flexibility to use different structures to represent the dynamic model. The use of gate mechanisms in the RNN encoder–decoder model limited the ability to clearly separate the contribution of the exogenous inputs from the hidden states. The design of the LSDNN’s structure lends itself to a more flexible representation of the effect of exogenous variables on the latent states.

Table 1. Prediction results of the LSDNN and RNN encoder–decoder.

Prediction Horizon	LSDNN (MAPE)	RNN Encoder–Decoder (MAPE)	LSDNN (RMSE)	RNN Encoder–Decoder (RMSE)
1	4.1 ± 0.2%	4.2 ± 0.2%	0.016 ± 0.001	0.017 ± 0.001
6	4.7 ± 0.2%	4.7 ± 0.1%	0.019 ± 0.001	0.02 ± 0.001
12	4.9 ± 0.2%	4.9 ± 0.1%	0.02 ± 0.001	0.02 ± 0.001
24	5.3 ± 0.3%	5.3 ± 0.1%	0.022 ± 0.001	0.022 ± 0.001

To further interpret the LSDNN model, Figure 6 shows the contribution of residual dynamics and exogenous variables to the latent states when performing one-step-ahead prediction. The blue line represents the contribution related to predicted residual dynamics \hat{z}_t' propagated using only the previous states without any effect of exogenous variables. The red line represents the effect of exogenous variables u_t' on the latent states. The final predicted latent state \hat{z}_t was obtained by adding these two values. With a latent variable denoting a sequence of latent states, for latent variables 1, 4, 5, 8, and 9, most of the variations are explained by u_t' , whereas, for the rest of the latent variables, both \hat{z}_t' and u_t' contribute to the predictions, and the variations are split into the red line and blue line. The grey dotted line is the true latent states that are mapped using the measurement encoder $f^{me}(x_t)$. It can be observed that the grey line stayed close to the black line, indicating that, after linearly removing the effects of exogenous variables, making a prediction using the residuals, and adding back the future effects of exogenous variables, the predicted latent states did not deviate from the states encoded from the real measurements.

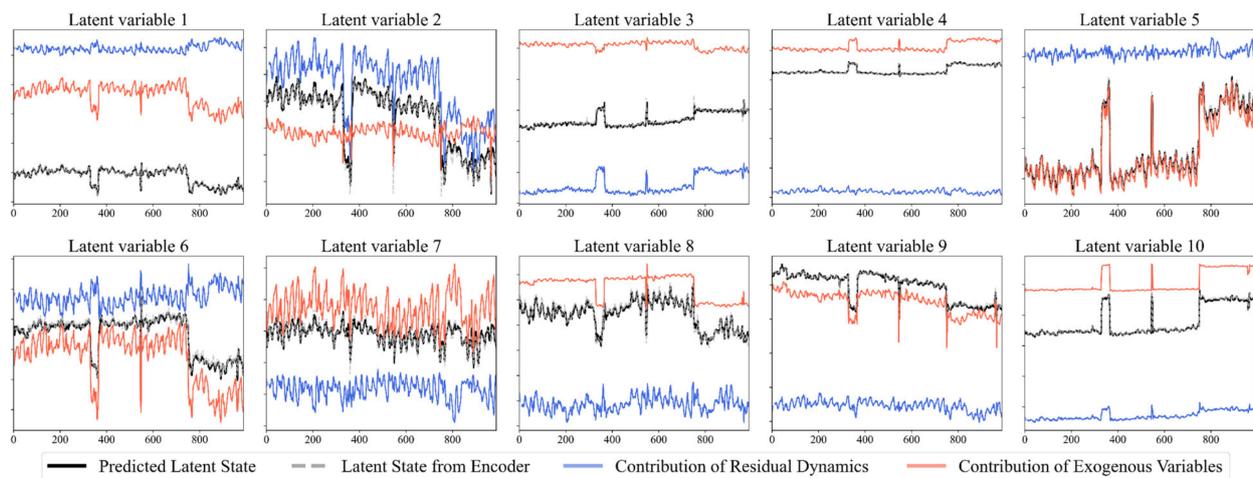


Figure 6. Contributions of exogenous variables and residual dynamics to predicted latent states. The predicted latent states stay close to the encoded latent states.

We used one latent variable as an example to visualize the role of the attention mechanism during one-step-ahead prediction, as shown in Figure 7. The top left subplot shows the past 12 latent states $[z'_{t-1}, \dots, z'_{t-12}]$ at each timestep before the attention mechanism is applied. Each row in the subplot represents the value of one past latent state. Due to the nature of one-step-ahead prediction, one state value appears 12 times in the plot. For example, if z'_{t-1} is the most recent state value at time t , it would show up as the last state value z'_{t-12} after 12 timesteps. The subplot on the top right is the updated past 12 latent states $[a_{t-1}z'_{t-1}, \dots, a_{t-12}z'_{t-12}]$ after including the attention mechanism. The subplot on the bottom left shows the attention weights $[a_{t-1}, \dots, a_{t-12}]$ at each timestep. Since the attention weights are related to the changes in the exogenous variables, we also show the control adjustments, which contain sudden significant jumps. It can be observed that the LSDNN model could adjust the attention weights based on the changes and the magnitude of the exogenous variables. For example, for the first 300 time steps in which the controls were not adjusted much, the attention mechanism put more weight on the second and sixth past states. When large changes occurred between timestep 300 and timestep 400, the weights were adjusted accordingly. The same behavior could also be observed when the controls were adjusted at around timestep 550. As a result, the stored past latent states were also scaled based on the attention weights.

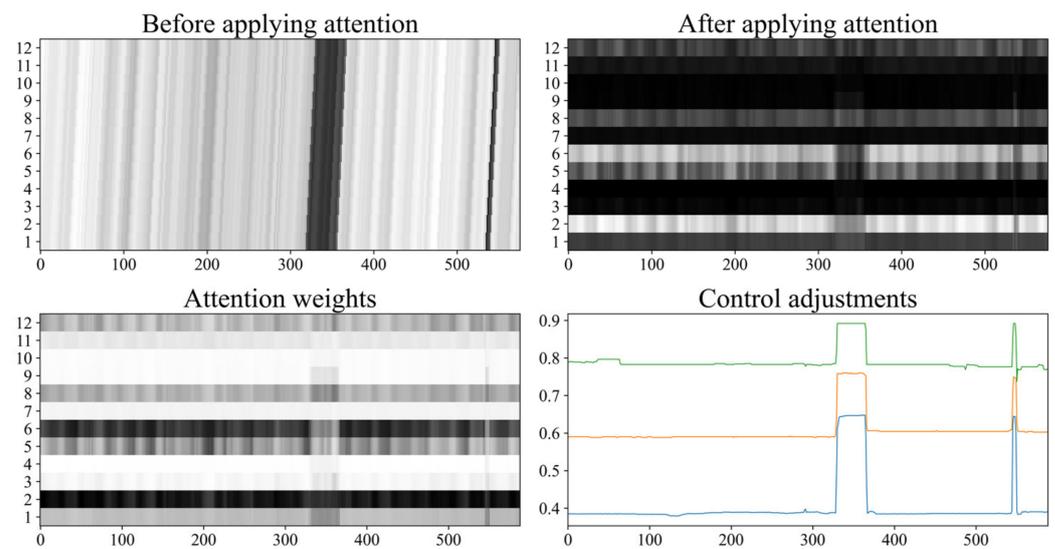


Figure 7. Changes between the input and output of the attention mechanism for one of the latent variables. The attention weights are adjusted based on control adjustments. The latent state values after applying attention are scaled by the attention weights.

4. Fault Detection in a Geothermal Power Plant

To perform fault detection, the general idea is to first build models from data collected during normal operations. Then, control limits are established to define normal operation regions. Finally, the models and the control limits are applied to new data for online fault detection [22,31]. An abnormal event is detected if the output of the models is not within the normal operation regions defined by the control limits. In this paper, the LSDNN model was first trained using data collected from normal operations to extract dynamic variations. As a result, after performing prediction, only normal static variations were present in the prediction errors, where a principal component analysis model could be built to establish the normal control limits on the residuals. When performing monitoring and fault detection on new data, the same LSDNN model, PCA model, and control limits were used.

In this paper, we adopted the well-established PCA-based processing monitoring techniques in [19] to monitor the LSDNN prediction errors. Denoting the one-step-ahead prediction errors of the LSDNN as $\tilde{\mathbf{X}}^{NN} \in \mathbb{R}^{L \times m}$, each row of the error matrix represents a sample at one timestep. Using PCA, the error matrix can be decomposed as

$$\tilde{\mathbf{X}}^{NN} = \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{X}}^{PCA} = \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{T}}\tilde{\mathbf{P}}^T \tag{12}$$

where \mathbf{T} is the score matrix and \mathbf{P} is the loading matrix corresponding to the first l leading principal components (PC). As a result, $span\{\mathbf{P}\}$ is the principal component subspace (PCS) and $span\{\tilde{\mathbf{P}}\}$ is the PCA residual subspace (RS). To monitor the variability in the PCS, Hotelling’s T^2 index can be used. It measures the distance to the origin in the principal component subspace, which contains normal variations. For each $\tilde{\mathbf{x}}^{NN} \in \mathbb{R}^m$, the index is defined as

$$T^2 = \tilde{\mathbf{x}}^{NNT} \mathbf{P}\mathbf{\Lambda}^{-1} \mathbf{P}^T \tilde{\mathbf{x}}^{NN} \tag{13}$$

$$\mathbf{\Lambda} = \text{diag}\{\lambda_1, \dots, \lambda_l\} \tag{14}$$

$$\lambda_i = \frac{1}{L-1} \mathbf{t}_i^T \mathbf{t}_i \tag{15}$$

where $\mathbf{t}_i \in \mathbb{R}^L$ is the i^{th} score vector. For a given significance level α , the sample is considered normal if T^2 is smaller than the corresponding control limit T_α^2 , where $T_\alpha^2 = \chi_{l,\alpha}^2$

if L is large [19]. To monitor the variability in RS, the squared prediction error (SPE) index can be used [19]:

$$SPE \equiv \| \tilde{\mathbf{x}}^{PCA} \|^2 = \| (\mathbf{I} - \mathbf{P}\mathbf{P}^T) \tilde{\mathbf{x}}^{NN} \|^2 \quad (16)$$

The SPE index measures variability that breaks the normal process correlation. The sample is considered normal if $SPE \leq \delta_\alpha^2 = g\chi_{h;\alpha}^2$, where δ_α^2 is the control limit for the SPE index and

$$g = \frac{\sum_{i=l+1}^m \lambda_i^2}{\sum_{i=l+1}^m \lambda_i}, \quad h = \frac{(\sum_{i=l+1}^m \lambda_i)^2}{\sum_{i=l+1}^m \lambda_i^2} \quad (17)$$

In addition, a combined index φ can be used as a global index that combines the SPE and T^2 indices [32]:

$$\varphi = T^2(\tilde{\mathbf{x}}^{NN}) + g^{-1}SPE(\tilde{\mathbf{x}}^{NN}) \quad (18)$$

The sample is normal if $\varphi \leq \chi_{h+l;\alpha}^2$.

We applied the fault detection approach to the field data in which two faults occurred in the geothermal power plant. One of them was a fault that occurred in one of the power generation units, and the other was a pump failure at one of the production wells. We trained two LSDNN models for the power generation unit and the production pump using data collected during normal operations. The LSDNN models were used to extract the dynamics and the effects from exogenous variables from available data. Then, the one-step-ahead prediction errors of the fault-free data were calculated. The prediction errors were scaled to have zero mean and unit variance to apply PCA for monitoring the static variations. We selected the number of leading PCs l so that the first l PCs captured 90% of the variances. The SPE, T^2 , combined index, and their corresponding control limits were then calculated. To monitor test data that contained faults, the trained LSDNN models were applied to the test data and obtain the prediction errors of the test data. With the loadings \mathbf{P} and $\mathbf{\Lambda}$ from the PCA model built from the normal data, the monitoring indices of the test data were calculated using Equations (13), (16), and (18). Finally, the indices were compared with the control limits to detect the faults.

Figure 8 shows the monitoring results for the test data from the power generation unit. The gaps between indices were due to the removal of data during the shutdown periods. At the end of the plot, the power plant operator discovered the fault and stopped the unit for repair. It can be observed that, before 11–19, most of the monitoring indices were below the control limits for all three indices, indicating that the unit was operating normally and there were no abnormal dynamics. A few samples had monitoring indices above the control limits then dropped below the control limits quickly when there were sudden adjustments in the control settings, or the unit was restarted. Since the adjustments were made by the operators, they could easily distinguish these false alarms from the detection of the real faults. In addition, when real faults occurred, the monitoring indices showed strong and consistent signatures, which were different from the false detections. The fault document that was provided by the operator showed that the unit was underperforming around 11–19, which could also be observed in the monitoring indices. Around 12–15, all three monitoring indices rose above the control limits and stayed above the limits until the plant was shut down for repair, which meant that a fault occurred around 12–15 and persisted until the operators discovered the fault.

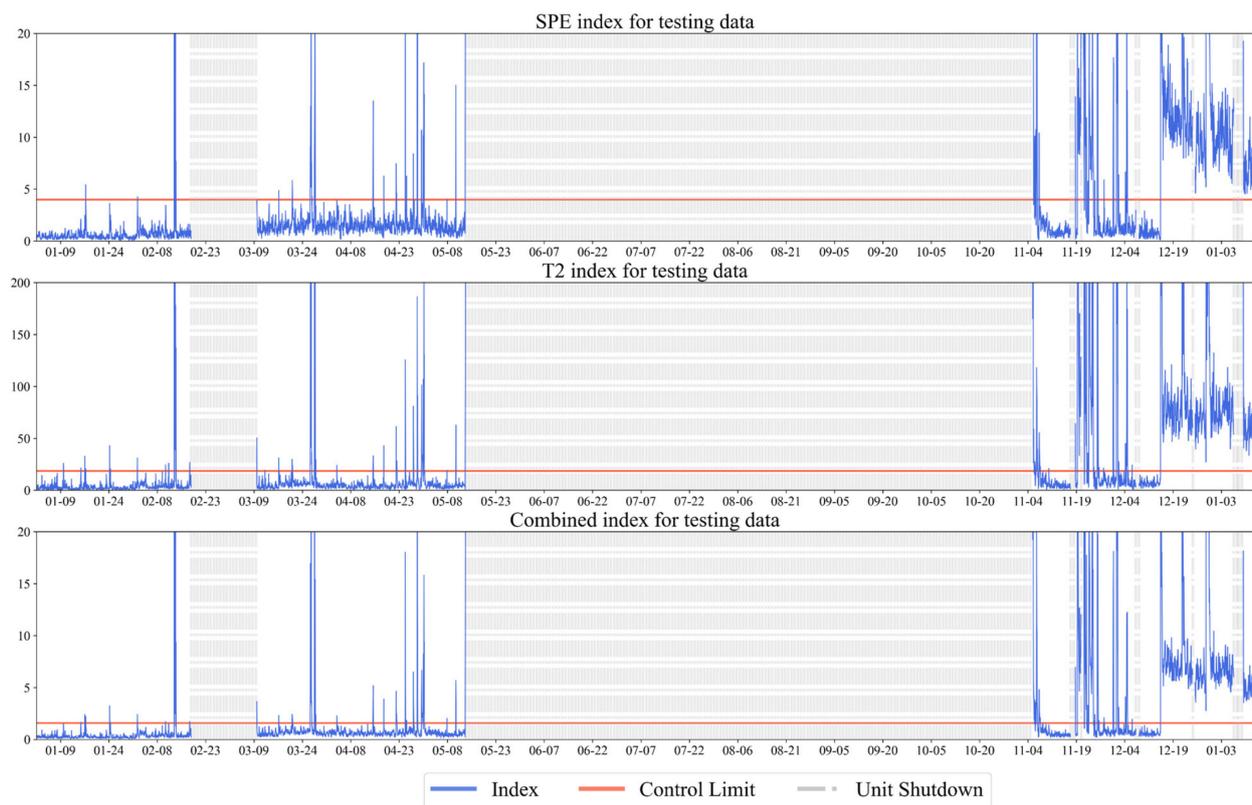


Figure 8. Monitoring indices of the power generation unit. All three indices rise above the control limits around 12–15, indicating the occurrence of faults.

After a fault is detected, the next step is to identify the location of the fault. Contribution plots are well-known tools for fault identification [20,33,34]. From the definition of SPE,

$$SPE(\tilde{x}^{PCA}) = \sum_{i=1}^m (\tilde{x}_i^{PCA})^2 = \sum_{i=1}^m SPE_i \quad (19)$$

where SPE_i is the contribution of the i th variable in the contribution plot for SPE. For better visualization, we plotted a 2D contribution plot that collected each variable's contribution for multiple timesteps, shown in Figure 9. On around 12–15, the variable turbo inlet superheat showed the greatest contribution to the SPE. After 12–15, the variables turbo inlet superheat, R134A outlet temperature of vaporizer B, and brine outlet temperature at vaporizer B had the largest contributions, indicating that they were either the major contributors to the fault or affected by the fault.

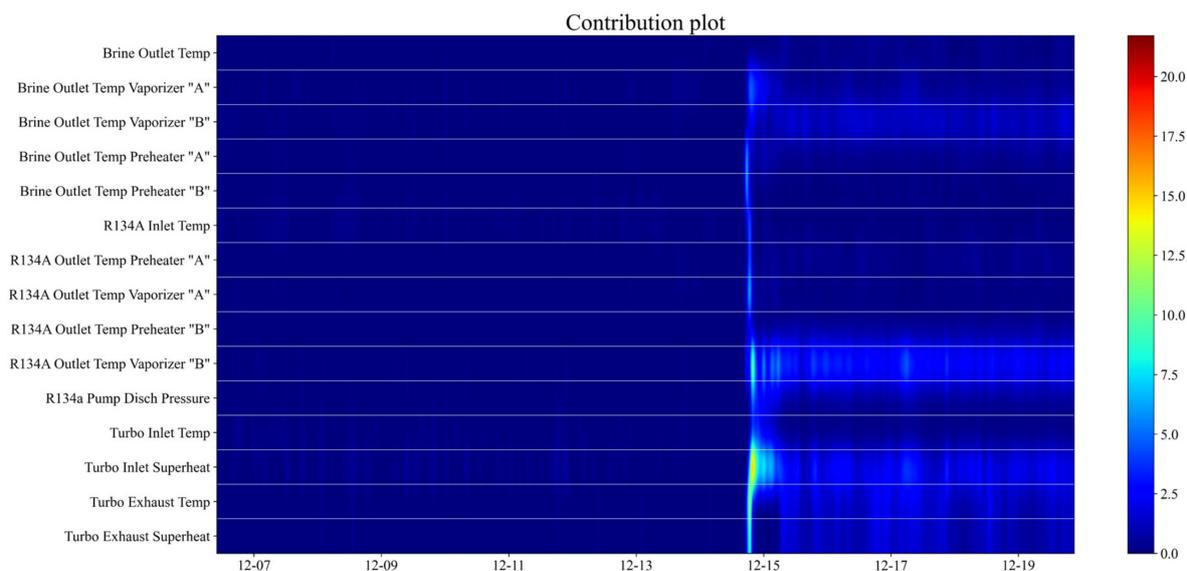


Figure 9. Contribution plot of the power generation unit. “Turbo inlet superheat” and “R134A outlet temperature vaporizer B” have the largest contributions, indicating that they are the possible fault locations.

We followed the same fault detection procedure for monitoring a production well pump. From the fault document, maintenance was performed on the pump from 03–20 to 03–27, and pump failure occurred on 09–26. Figure 10 shows the one-step-ahead prediction results of the measured variables of the pump. Before maintenance, the trained LSDNN model provided accurate predictions. However, offsets in the predictions could be observed in some of the variables after maintenance, indicating changes in the pump’s performance. While the pump components and the nature of the repair are unknown, maintenance, such as sensor recalibration and replacing new components, can cause offsets in the prediction results. However, the underlying physical relationships between different measured and exogenous variables are expected to remain the same before and after maintenance. It can be observed in the prediction results that, even though the offsets existed, the predictions followed similar trends to the measured variables, indicating that the LSDNN model learned the underlying dynamics. Since the dynamics remained the same and the fault monitoring relied on the dynamic relationships and static correlations, we performed the same fault monitoring procedure to determine if the proposed procedure could detect a fault without retraining the LSDNN model. Figure 11 shows the three monitoring indices. It can be observed that the T^2 and the combined index rose above their corresponding control limits immediately after maintenance was performed. However, the SPE index gradually grew above the control limit between 08–06 and 08–21. Since the T^2 index was used to monitor the leading PCs, it measured the distance of the sample to the origin in the PCS. If a sample exceeded the T^2 control limit, it meant that the sample shifted away from the origin in the PCS, but the correlation structure was not violated. On the other hand, the SPE was used to monitor the RS and measure the variability that breaks the correlation relationships. As a result, if the T^2 index of a sample is above the T^2 control limit but the SPE index lies below the SPE control limit, it could indicate a fault or shift in the operating region [19]. In this pump-related application, we know that maintenance caused the offsets in the prediction errors, which then caused the shift in the PCs. As a result, the T^2 index exceeding the control limit did not indicate a fault. Since the combined index was also related to the T^2 index, we should only use the SPE for fault detection in this case. Hence, the SPE index exceeding the control limit between 08–06 and 08–21 indicated a fault occurred which broke the correlation structure.

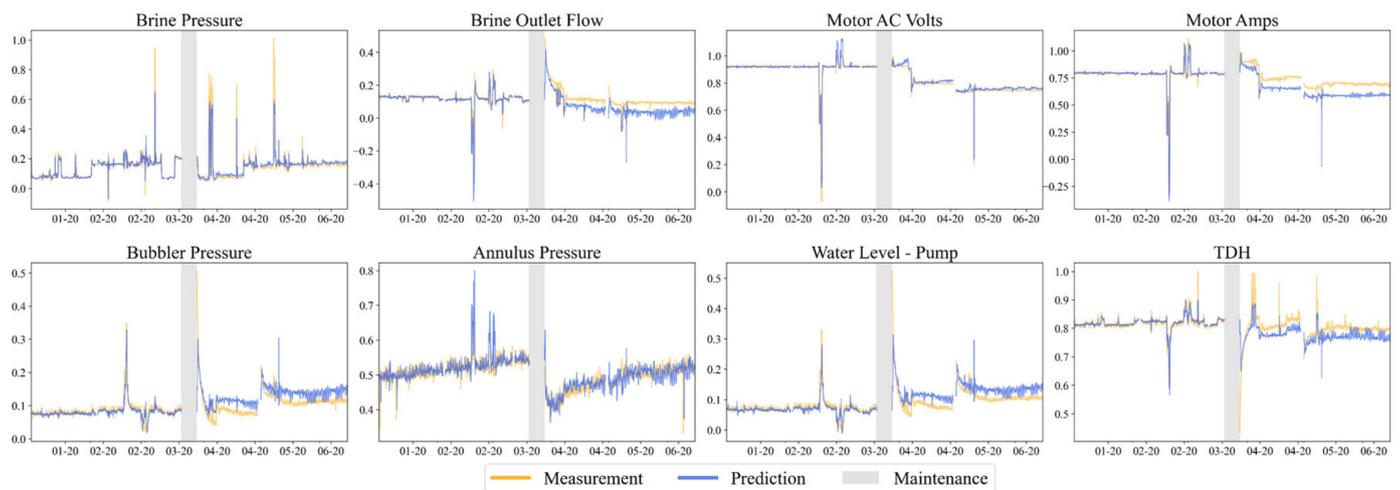


Figure 10. One-step-ahead prediction results of the production pump.

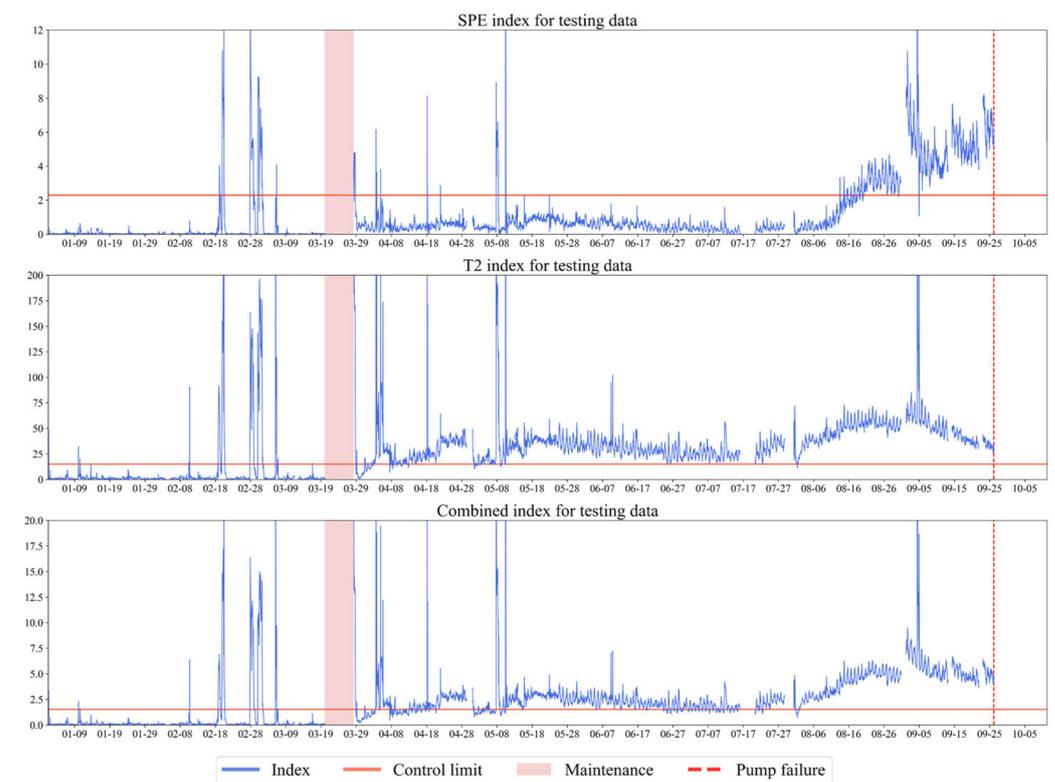


Figure 11. Monitoring indices of the production pump. The T^2 index rises above the control limit after maintenance, indicating a shift in the operation region. The SPE index exceeds the control limit after 08–06, indicating a fault that breaks the correlation structure.

Both the fault detection results for the power generation unit and production pump showed that the proposed fault detection procedure could detect abnormal dynamics before the faults were discovered in the field by the operators. As a result, if the fault detection procedure can be implemented in a geothermal power plant, it is able to detect faults earlier, possibly preventing unscheduled shutdowns or catastrophic failures of equipment.

5. Conclusions

In this paper, a latent space dynamic neural network model was proposed for multi-step-ahead prediction. The model structure was designed based on the characteristics of

time-series data collected from geothermal power plants. An encoder–decoder structure was implemented to map the original data to a reduced-dimension latent space in which the dynamics of the power generation process were represented. The predictions were made in the latent space using a dynamic model. In the dynamic model, the effects of the exogenous variables were linearly separated from the latent states to reduce the model complexity and improve interpretability. We also used an attention mechanism to adaptively change the stored latent states based on changes in exogenous variables, which gave the model more flexibility when predicting sudden changes in the measured variables. The prediction performance of the LSDNN on a geothermal power plant dataset was compared with a popular RNN encoder–decoder structure with LSTM. We also proposed a fault monitoring procedure based on the one-step-ahead prediction results of the LSDNN model. First, the LSDNN model was used to extract the dynamics and remove autocorrelations from the time-series data. Then, PCA-based fault detection techniques were applied to the residuals to monitor the static variations. The fault detection procedure was applied to two known faults corresponding to a power generation unit and a production pump. The results show that, in both cases, signatures of abnormal dynamics can be detected before the faults are discovered in the field.

The proposed LSDNN model can be extended to include other structures for the dynamic model in the latent space. Additional studies are needed to explore fault prediction and the possibility of an advance fault warning system based on multi-step-ahead prediction. In addition to fault prediction and detection, the model can be extended to include fault diagnosis to inform possible causes of a potential failure or abnormal events. Another topic that was not explored in this paper was retraining and updating the model with incoming data, which is expected to improve the predictive power of the model. Furthermore, in the case study of the production pump failure, the predictions were affected by maintenance, and retraining could be used to adapt the dynamics to the changes due to maintenance. Finally, the LSDNN model could also be used as a data-driven predictive model in broader applications, such as model predictive control and optimization of the geothermal power plants.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L.; validation, W.L.; formal analysis, Y.L.; investigation, Y.L.; resources, J.Z. and T.T.C.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, R.Y. and B.J.; visualization, Y.L.; supervision, R.Y. and B.J.; project administration, B.J.; funding acquisition, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under the Geothermal Technologies Office award number DE-EE0008765.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to the ownership of the data sets used in this paper, not all information may be publicly disclosed at the time of publication.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; or in the writing of the manuscript.

References

1. Moya, D.; Aldás, C.; Kaparaju, P. Geothermal energy: Power plant technology and direct heat applications. *Renew. Sustain. Energy Rev.* **2018**, *94*, 889–901. [[CrossRef](#)]
2. Bertani, R. Geothermal power generation in the world 2010–2014 update report. *Geothermics* **2016**, *60*, 31–43. [[CrossRef](#)]
3. Manente, G.; Toffolo, A.; Lazzaretto, A.; Paci, M. An Organic Rankine Cycle off-design model for the search of the optimal control strategy. *Energy* **2013**, *58*, 97–106. [[CrossRef](#)]
4. Ghasemi, H.; Paci, M.; Tizzanini, A.; Mitsos, A. Modeling and optimization of a binary geothermal power plant. *Energy* **2013**, *50*, 412–428. [[CrossRef](#)]

5. Imran, M.; Pili, R.; Usman, M.; Haglind, F. Dynamic modeling and control strategies of organic Rankine cycle systems: Methods and challenges. *Appl. Energy* **2020**, *276*, 115537. [[CrossRef](#)]
6. Wang, X.; Liu, X.; Zhang, C. Parametric optimization and range analysis of Organic Rankine Cycle for binary-cycle geothermal plant. *Energy Convers. Manag.* **2014**, *80*, 256–265. [[CrossRef](#)]
7. Leva, S.; Dolara, A.; Grimaccia, F.; Mussetta, M.; Ogliari, E. Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power. *Math. Comput. Simul.* **2017**, *131*, 88–100. [[CrossRef](#)]
8. Gensler, A.; Henze, J.; Sick, B.; Raabe, N. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 002858–002865.
9. Lee, K.Y.; Heo, J.S.; Hoffman, J.A.; Kim, S.-H.; Jung, W.-H. Neural Network-Based Modeling for A Large-Scale Power Plant. In Proceedings of the 2007 IEEE Power Engineering Society General Meeting, Tampa, FL, USA, 24–28 June 2007. [[CrossRef](#)]
10. Lee, G.; Lee, S.J.; Lee, C. A convolutional neural network model for abnormality diagnosis in a nuclear power plant. *Appl. Soft Comput.* **2020**, *99*, 106874. [[CrossRef](#)]
11. Arslan, O.; Yetik, O. ANN based optimization of supercritical ORC-Binary geothermal power plant: Simav case study. *Appl. Therm. Eng.* **2011**, *31*, 3922–3928. [[CrossRef](#)]
12. Haklıdır, F.S.T.; Haklıdır, M. Prediction of Reservoir Temperatures Using Hydrogeochemical Data, Western Anatolia Geothermal Systems (Turkey): A Machine Learning Approach. *Nonrenew. Resour.* **2019**, *29*, 2333–2346. [[CrossRef](#)]
13. Shi, Y.; Song, X.; Song, G. Productivity prediction of a multilateral-well geothermal system based on a long short-term memory and multi-layer perceptron combinational neural network. *Appl. Energy* **2020**, *282*, 116046. [[CrossRef](#)]
14. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [[CrossRef](#)] [[PubMed](#)]
15. Masti, D.; Bemporad, A. Learning nonlinear state–space models using autoencoders. *Automatica* **2021**, *129*, 109666. [[CrossRef](#)]
16. Chen, S.; Billings, S.A.; Grant, P.M. Non-linear system identification using neural networks. *Int. J. Control* **1990**, *51*, 1191–1214. [[CrossRef](#)]
17. Lin, T.; Horne, B.; Tino, P.; Giles, C.L. Learning Long-Term Dependencies in NARX Recurrent Neural Networks. *IEEE Trans. Neural Netw.* **1999**, *7*, 1329–1338. [[CrossRef](#)]
18. Menezes, J.M.P., Jr.; Barreto, G.A. Long-term time series prediction with the NARX network: An empirical evaluation. *Neurocomputing* **2008**, *71*, 3335–3343. [[CrossRef](#)]
19. Qin, S.J. Statistical process monitoring: Basics and beyond. *J. Chemom.* **2003**, *17*, 480–502. [[CrossRef](#)]
20. MacGregor, J.; Kourti, T. Statistical process control of multivariate processes. *Control Eng. Pr.* **1995**, *3*, 403–414. [[CrossRef](#)]
21. Wise, B.M.; Gallagher, N.B. The process chemometrics approach to process monitoring and fault detection. *J. Process Control* **1996**, *6*, 329–348. [[CrossRef](#)]
22. Dong, Y.; Qin, S.J. New Dynamic Predictive Monitoring Schemes Based on Dynamic Latent Variable Models. *Ind. Eng. Chem. Res.* **2020**, *59*, 2353–2365. [[CrossRef](#)]
23. Chen, J.; Liao, C.-M. Dynamic process fault monitoring based on neural network and PCA. *J. Process Control* **2002**, *12*, 277–289. [[CrossRef](#)]
24. Sorjamaa, A.; Hao, J.; Reyhani, N.; Ji, Y.; Lendasse, A. Methodology for long-term prediction of time series. *Neurocomputing* **2007**, *70*, 2861–2869. [[CrossRef](#)]
25. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv* **2014**, arXiv:1409.3215.
26. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
27. Lai, G.; Chang, W.-C.; Yang, Y.; Liu, H. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In Proceedings of the SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104. [[CrossRef](#)]
28. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G.W. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *IJCAI* **2017**, 2627–2633. [[CrossRef](#)]
29. Zarrouk, S.J.; Moon, H. Efficiency of geothermal power plants: A worldwide review. *Geothermics* **2014**, *51*, 142–153. [[CrossRef](#)]
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Qin, S.J.; Dong, Y.; Zhu, Q.; Wang, J.; Liu, Q. Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring. *Annu. Rev. Control* **2020**, *50*, 29–48. [[CrossRef](#)]
32. Yue, H.H.; Qin, S.J. Reconstruction-Based Fault Identification Using a Combined Index. *Ind. Eng. Chem. Res.* **2001**, *40*, 4403–4414. [[CrossRef](#)]
33. Kourti, T. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemom. Intell. Lab. Syst.* **1995**, *28*, 3–21. [[CrossRef](#)]
34. Miller, P.; Swanson, R.E.; Heckler, C.E. Contribution plots: A missing link in multivariate quality control. *Appl. Math. Comput. Sci.* **1998**, *8*, 775–792.