

ESN Neural Network-Based Electrical Submersible Pump Model

Chuan Wang, Zhen Long

Energy Equipment Research Institute, Southwest Petroleum University, China

Abstract: This paper introduces an Electrical Submersible Pump (ESP) model based on the Echo State Network (ESN) neural network. ESPs are commonly used in oil extraction, and their performance is influenced by a variety of factors, making accurate modeling crucial for optimizing operations and maintenance. ESN is a type of recurrent neural network with unique dynamic memory capabilities, making it suitable for handling time-series data and modeling nonlinear dynamic systems. The article details the ESN network structure and its application in the ESP model, including how to use ESN to predict performance parameters and diagnose faults in ESPs. Compared with traditional models, the ESN model shows significant advantages in prediction accuracy and computational efficiency. The results indicate that the ESN-based ESP model can effectively simulate the dynamic behavior of ESPs, providing a powerful tool for real-time monitoring and fault prevention of ESPs.

Keywords: ESN-RLS network; PSO optimization parameters; Kalman filter method.

1. Introduction

Electric submersible pump (Electrical Submersible Pump, ESP) is a widely used equipment in the petroleum industry to transport the underground liquid to the ground. The ESN network model based on Python3.9 was constructed according to the physical model of the electric submersible pump, And wrote the corresponding update algorithm (RLS); And the parameters of optimizing the network via the PSO algorithm, Bring the network state to the current optimum, Improve the robustness of the model; Then, through the NumPy and CasADi open-source libraries in the PyCharm development environment, Write the data set used for network training; In verifying the authenticity of the model, The LSTM network and GRU network model and the ESN network model were written, The authenticity of the ESN model prediction is obtained; Finally, it was compared with the real value according to the training results, Expand the evaluation of this network model.

An artificial neural network is a machine learning technology that is inspired by the operation of neurons in the human brain. This technology excels at handling complex problems that are difficult to model precisely with traditional algorithms, demonstrating exceptional flexibility and adaptability[1]. In various fields such as automatic

handwriting recognition, object recognition, autonomous driving technology, and many others, artificial neural networks play a significant role. Their core advantage lies in their powerful learning ability, which stems from the deep analysis and utilization of large-scale training datasets. Neural networks continuously optimize their internal parameters, automatically learning and identifying potential patterns and regularities within the data, thereby achieving accurate predictions for new samples[6].

The structure of a neural network consists of neurons designed in multiple layers. Each layer of neurons undertakes specific information processing tasks. As computational units, neurons multiply the system input values by weights and accept weighted inputs from the previous layer of neurons, adding a bias term to increase the flexibility of the model. The computation process is shown in Equation 1-1.

$$\sum_{i=1}^n x_i w_i + b \tag{1-1}$$

Subsequently, through the action of the activation function, the neuron produces an output, which then serves as input for the next layer of neurons, proceeding layer by layer until the end of the network. As shown in Figure 1-1, the mechanism of information flow between neurons is depicted[7].

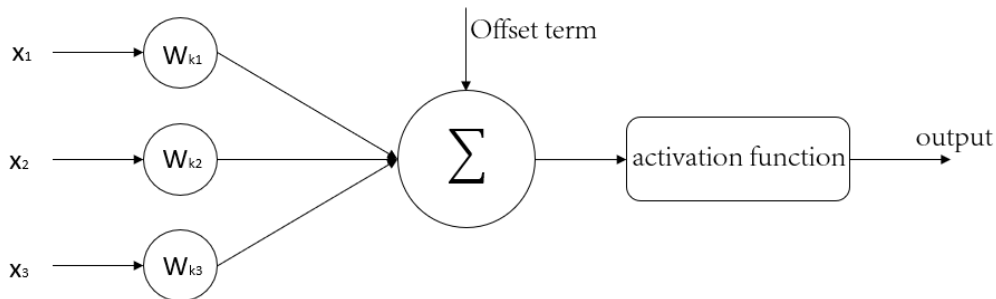


Figure 1-1. Schematic diagram of neuronal information flow

This article will start from the basic structure of artificial neural networks[10], and through the analysis of three

dimensions: the input layer, the hidden analysis layer composed of multiple neurons, and the output layer, construct

between the internal neurons of the output layer in Figure 1-3 is generally not used, while the connection weight matrix from the output layer to the hidden layer is optional. In the actual construction of the model, this thesis will not consider these two parts of the connection weight matrix, but instead adopt only the training of the connection weight matrix from the hidden layer to the output layer. The calculation method for the hidden layer output and the output layer output is changed to:

$$h_t = f\left(\left(w^i\right)^T x_t + \left(w^r\right)^T h_{t-1} + b^r\right) \quad (1-4)$$

$$y_t = \varphi\left(\left(w^o\right)^T \left[x_t^T, h_t^T\right]^T + b_{t-1}^o\right) \quad (1-5)$$

Wherein, $w^0 \in R^{(n_i+n_r) \times n_o}$ represents the connection weight matrix from the input layer and hidden layers to the output layer[17]. Unlike traditional RNN series that require training for all connection weights, ESN only requires training for the output weight matrix w^0 . The input weight matrix w^i of ESN is determined by random generation during the network initialization phase and remains fixed throughout subsequent training processes. These weights are uniformly distributed random numbers. They are also generated randomly and remain fixed, but to ensure the echo state property of ESN, the spectral radius $\rho(w^r)$ must be less than or equal to 1. The fewer network weight training requirements make the structure of ESN simpler and significantly reduce the training difficulty.

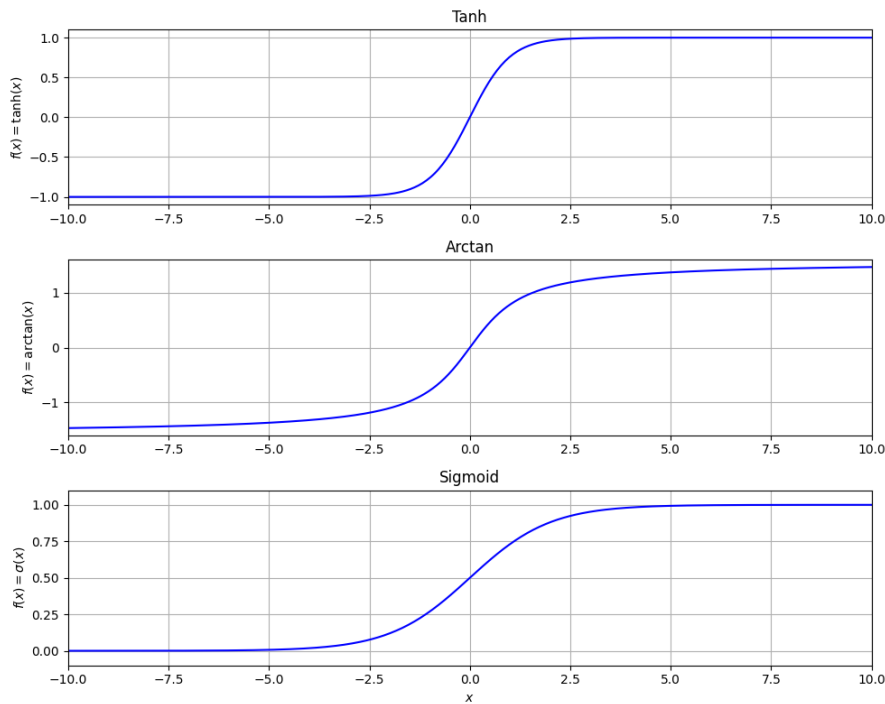


Figure 1-4. Diagram of different types of activation functions

2.2. Key parameters of ESN

Capacity of the hidden layer. The hidden layer is located in the middle of a neural network and is responsible for information processing and transmission. The capacity of the hidden layer refers to the number of neurons within it, and generally, the larger the scale, the higher the potential performance of the network. The spectral radius is the spectral radius of the connection weight matrix of the hidden layer, that is, the maximum of the absolute values of all eigenvalues. The spectral radius determines the speed at which the input causes the reservoir state to decay over time, as well as the stability of the hidden layer. The neurons inside the hidden layer are required to be sparsely connected, and the leakage rate is to measure the degree of the connection between the neurons inside the hidden layer. It is the ratio of the number of neurons in the hidden layer and the total number of neurons in the hidden layer, that is, the number of non-zero elements, and the value is between (0,1). Enter the scaling factor, Input scaling factor regulates the effect of the input weight matrix on the network.

2.3. ESN online learning

Because the ESN network model of electric submersible pump has a large amount of actual required prediction data, It is w^i and w^r fixed in the initialization stage of the network. In order to better iteratively update the ESN's output weight matrix online, the constant mode recursive least squares method (RLS) is adopted.

Let the input data at the current moment be r_t and the corresponding expected output is for \hat{y}_t , And the predicted target output, defining its objective function as:

$$L(w^0) = \frac{1}{2} \sum_{t=1}^t \lambda^{t-\tau} \left(\hat{y}_\tau - (w^0)^T r_\tau \right)^2 \quad (1-6)$$

Where $\lambda \in (0, 1]$ is the forgetting factor, when close to 1, the current model will consider the impact of historical data, the model changes slowly when adapting to the new data;

when close to 0, the model will reduce the weight of the historical data, so as to adapt to the new input faster.

Order $\nabla_{w^0} L(w^0) = 0$, can get:

$$(w^0)^* = \sum_{\tau=1}^t \lambda^{t-\tau} (r_{\tau} r_{\tau}^T)^{-1} r_{\tau} \hat{y}_{\tau} \quad (1-7)$$

Sorting out available:

$$w_t^0 = (w^0)^* = A_t^{-1} b_t \quad (1-8)$$

among:

$$A_t = \sum_{\tau=1}^t \lambda^{t-\tau} r_{\tau} r_{\tau}^T \quad (1-9)$$

$$b_t = \sum_{\tau=1}^t \lambda^{t-\tau} r_{\tau} \hat{y}_{\tau} \quad (1-10)$$

Order, form (1-9) and (1-10) write:

$$A_t = \lambda A_{t-1} + r_t r_t^T \quad (1-11)$$

$$b_t = \lambda b_{t-1} + r_t \hat{y}_t \quad (1-12)$$

On the basis of equation (1-11), by Sherman-Morrison-Woodbury formula, the recursion is rewritten as:

$$P_t = \frac{1}{\lambda} P_{t-1} - \frac{1}{\lambda} g_t u_t^T \quad (1-13)$$

among:

$$u_t = P_{t-1} r_t \quad (1-14)$$

$$g_t = \frac{u_t}{\lambda + u_t^T r_t} \quad (1-15)$$

For the gain vector g_t . from(1-11), (1-13)in(1-8), The final updated weights are:

$$w_t^0 = w_{t-1}^0 - g_t e_t \quad (1-16)$$

Inside $e_t = \left((w_{t-1}^0)^T r_t - \hat{y}_t \right)$.

3. The ESN Network Model of The Electric Submersible Pump

3.1. Data preprocessing

In the process of industrial data collection, due to the existence of various interference factors, the collected data often contains many problems. To improve the quality and usability of the data, data preprocessing is required, including but not limited to the addition of new features, removing

duplicates, transforming data formats, application of interpolation techniques, and noise processing. These operations help to clean and collate the data, making it more suitable for expression and analysis. In the preprocessing process, mastering the data characteristics, process flow and equipment operation rules of a specific field is crucial to accurately identify and handle outliers. At the same time, in order to meet the needs of the algorithm model, the scale and format of the data must be considered to ensure that the data input meets the requirements of the model, so as to improve the training and prediction efficiency of the model.

There may be a strong intercorrelation between the state data of the electric submersible pump. For the various variables being continuously monitored, the data between the different variables may show great variability due to the different means of measurement. To reduce the impact of this variability, we need to standardize and normalize the values of these scalars.

(1) Z-Score standardization

For x_1, x_2, \dots, x_n the sequence, its mathematical expression is:

$$y_i = \frac{x_i - \bar{x}}{s} \quad (2-1)$$

Inside $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$.

(2) Min-Max for normalization

Since each feature has different meanings, the 0-1 normalization should be applied to the sample set. In neural network deep learning, data normalization is a common preprocessing step that ensures the consistent numerical range of different features and guarantees the performance of the model. Maximum minimum normalization is a common method to deal with large differences between different data. Linear mapping can reduce the range of required data to [0,1], which can better adapt to various machine learning models. The mathematical expression is provided as follows:

$$x_{ij} = \frac{x_{ij} - x_{\min}}{x_{\max} - x_{\min}} \quad (2-2)$$

Inside $\begin{cases} i = 1, 2, \dots, n \\ j = 1, 2, \dots, m \end{cases}$, x_{\min} is the minimum value in the

x_{ij} , x_{\max} is the maximum value in the x_{ij} , m is the sample size of the data n is the dimension of the data characteristics. The normalized matrix Z was obtained as follows:

$$Z = \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{m1} & z_{m3} & \dots & z_{mn} \end{pmatrix}$$

After the normalized data, it is necessary to ensure that the individual features are compared at the same scale. The normalized data were divided into training, validation and test set in a certain proportion. The main purpose of the training

set is to train the model and build an effective model by learning the patterns and features of the data. The validation set plays an important role in the process of model training. It is used to evaluate whether the performance of the model meets the expectation, so as to provide a reliable reference for the subsequent debugging and optimization of the hyperparameters of the network model, avoid overfitting or underfitting problems, and ensure that the model has good generalization ability. The final test set was retained to evaluate the final performance of the model, ensuring that the model performed expected on location data.

As shown in Figure 2-1, data trends of production parameters of well A2 under different working conditions.

These data are collected continuously, so as to accurately capture the changes of each parameter under different working conditions. Obviously, there are significant differences in each parameter under different working conditions. In this study, sample data from the normal and cumulative phases were used to predict trends in the system state and draw corresponding control measures accordingly. In this way, we can take measures to adjust and optimize the system before the state deteriorates to an unacceptable degree, so as to ensure the stable operation of the system. This control mode not only improves the reliability and security of the system, but also effectively reduces the maintenance cost.

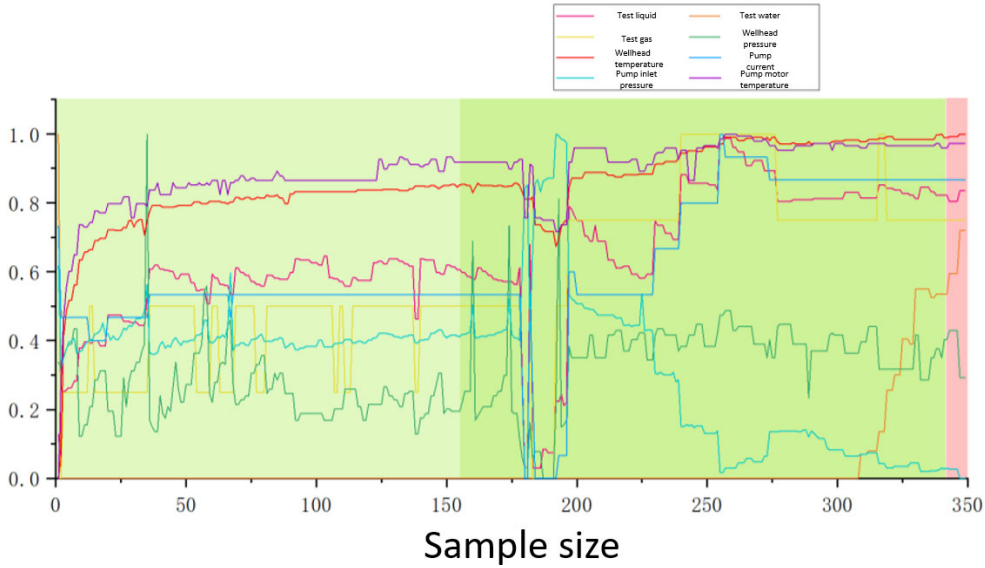


Figure 2-1. Diagram of different types of activation functions

(3) Noise reduction treatment

Noise reduction processing is an important step in data preprocessing, especially in the processing of industrial field data, due to the influence of environmental noise, equipment aging or sensor accuracy and other factors, the data often contains a lot of noise. The presence of noise will disturb the model to identify the real signal and reduce the prediction

accuracy and reliability. Therefore, sampling effective noise reduction techniques are crucial to improve data quality and ensure model performance. Here, using Matlab for joint simulation and Simulink to import real-time data from the electric submersible pump, the noise situation in the data is shown in Figure 2-2.

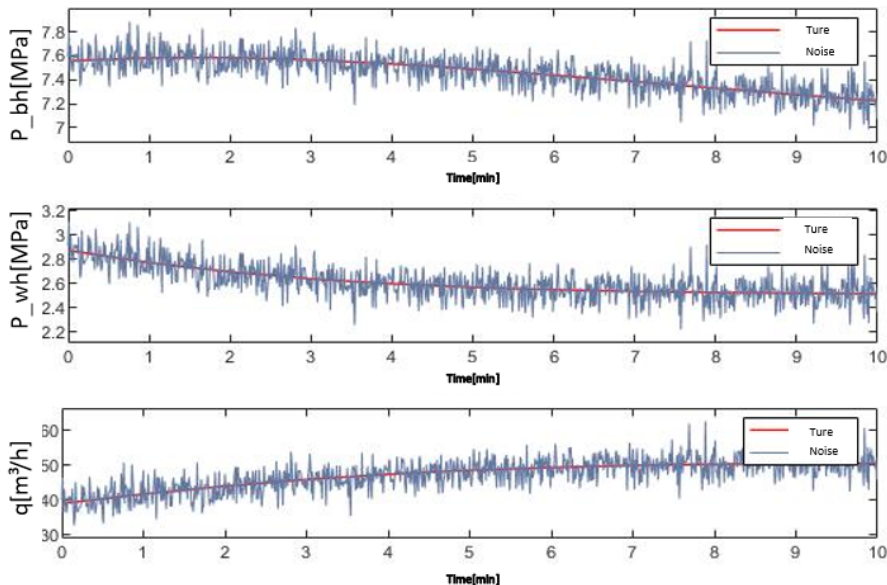


Figure 2-2. Schematic diagram of downhole data (including noise)

In the ESN network model of the electric submersible pump, the noise reduction process usually includes the following aspects: removing abnormal values due to measurement errors or interference during data transmission; a filtering technique that removes the unwanted noise components based on the frequency characteristics of the signal and an adaptive filtering technique that can automatically adjust its parameters according to the statistical characteristics of the signal; or by wavelet transform, the signal can be decomposed into components of different scales and effectively suppress the noise components. In this study, the measured values of wellhead pressure (p_{bh}), well bottom pressure (p_{wh}) and flow rate q , combined with the electric submersible pump system model and the filtering results will be used for the prediction system for system state control. The spatial equation of the state of the system is generally formulated as a discrete nonlinearity, as shown in the following equation:

$$\dot{x} = f(x, u, t) + w \quad (2-3)$$

$$y = g(x, u, t) + v \quad (2-4)$$

In the formula: x — State vector;

u — System input variable;
 w, v — System process interference, and output interference;

The system state variable was defined in this study as:

$$x = [p_{bh}, p_{wh}, q] \quad (2-5)$$

The main state of the system can be described using the following equation:

$$\begin{aligned} \frac{dp_{bh}}{dt} &= \frac{V_1}{\beta} (q_r - q) + w_1(t) \\ \frac{dp_{wh}}{dt} &= \frac{V_2}{\beta} (q - q_c) + w_2(t) \end{aligned} \quad (2-6)$$

$$\frac{dq}{dt} = \frac{1}{M} (p_{bh} - p_{wh} - \rho g h_w - \Delta p_f - \Delta P_p) + w_3(t)$$

In the formula: $w_i(t)$ — Internal interference of the system;

Kalman filtering is an efficient recursive data fusion method to estimate the state of the model by minimizing the error variance. Here the extended Kalman filter is used as a systematic observation strategy, and the flow chart is shown in Figure 2-3.

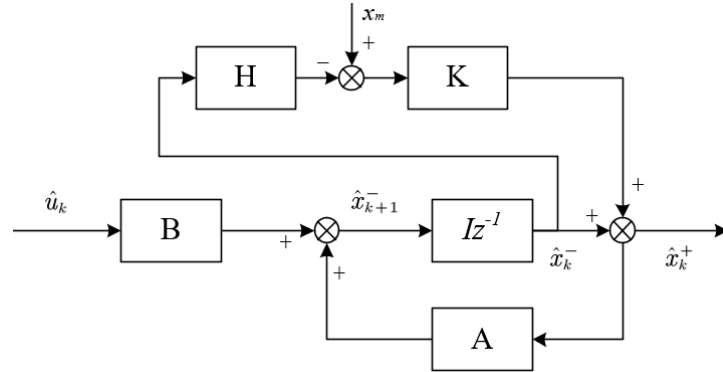


Figure 2-3. Flow chart of extended Kalman filter observations

Extended Kalman filtering involves a Taylor series expansion of the model of nonlinear systems and a truncation treatment in the expanded equations, usually choosing the first order truncation. This truncation method helps to reduce the computational burden of the system. When a nonlinear system is approximately linear, the system can be described as:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_k + w_k \\ y &= Hx_k + v_k \end{aligned} \quad (2-7)$$

According to the system status $x = [p_{bh}, p_{wh}, q]$, Combined equation of state of the system (2-8). The extended Kalman algorithm is:

$$\begin{aligned} \dot{p}_{bh} &= \frac{V_1}{\beta} (q_r - q) \\ \dot{p}_{wh} &= \frac{V_2}{\beta} (q - q_c) \end{aligned} \quad (2-8)$$

$$\dot{q} = \frac{1}{M} (p_{bh} - p_{wh} - \rho g h_w - \Delta p_f - \Delta P_p)$$

Step1 Calculate the Jacobian matrix F_{k-1} , namely, the first-order Taylor expansion at the current computation point;

$$F_{k-1} = \left. \frac{\partial f_{k-1}}{\partial x} \right|_{\hat{x}_{k-1}^+}$$

In this study, the nonlinear model of the system is described by equation (3-10). The state of the system in this paper is a 3D vector. The linearization of the model will be the first order linear expansion here to obtain the Jacobian matrix of

the current expansion point F_{k-1} .

Step2 Estimation error matrix and state estimation;

Let the system state external interference be w_k , Mark its covariance matrix as Q , The error matrix can be iterated as follows.

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1}$$

$$\hat{x}_k^- = A \hat{x}_{k-1}^+ + B u_{k-1}$$

Step3 Measurement update for state estimation and estimated error covariance.

$$H_k = \frac{\partial h_k}{\partial x} \Big|_{\hat{x}_k^-}$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k [y_k - H_k x_k^-]$$

$$P_k^+ = (I - K_k H_k) P_k^-$$

By extending the method of Kalman filter, the data of the downhole data of the electric submersible pump after noise reduction is shown in Figure 2-4, and the signal to noise ratio is greatly improved. The data is obvious, which an ideal reference for subsequent feature extraction.

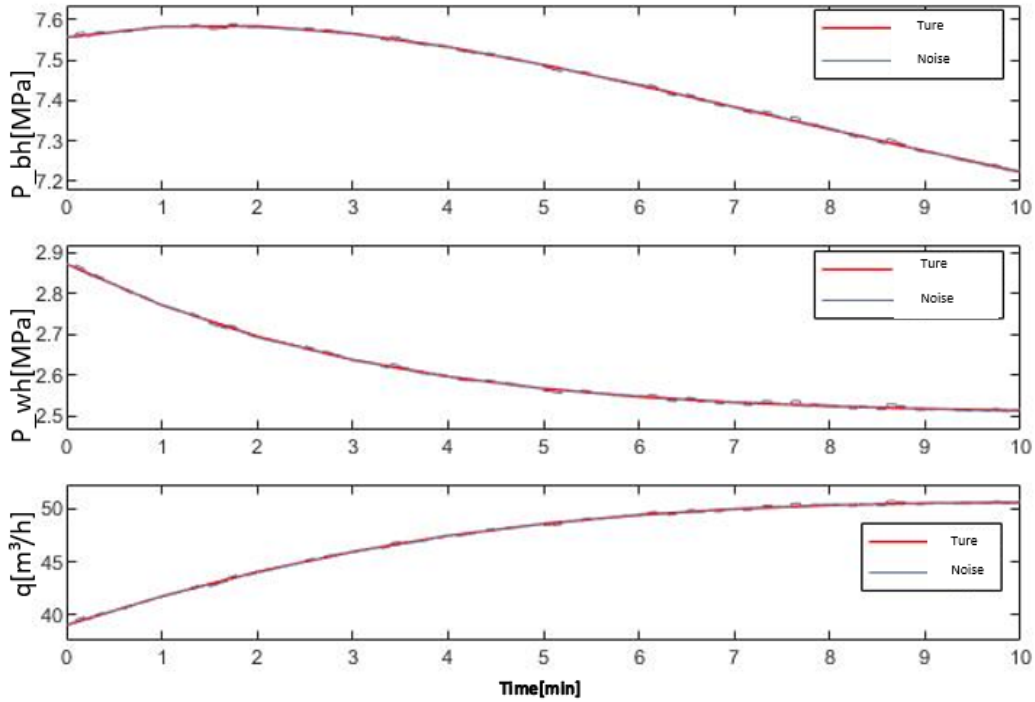


Figure 2-4. Data signals after Kalman filtering

3.2. Feature extraction

Feature extraction plays a crucial role in the applications of machine learning. When dealing with practical problems, large amounts of raw data are often encountered, which are often complex and high-dimensional collections of information. Despite the wealth of information contained in these data, machine learning algorithms do not directly understand the implications of these data. The aim is to extract key information from the raw data and retain important features of the data while reducing redundant information. This process not only helps to translate the data into algorithmically interpretable forms, but also provides statistical and physically significant features for machine learning models. Through feature extraction, we can effectively represent the features of the original data to improve the performance of the model. Moreover, feature extraction also helps to reduce the computational amount of [55]. Through feature extraction, we can significantly reduce

the dimensionality of the data, thus reducing the computational resources required for model training and prediction. This not only improves computational efficiency, but also reduces computational cost. Feature extraction is used as a means to reduce the noise contained in the sample data, and the model can better capture the key information in the data and improve the accuracy and robustness of its prediction.

The specific parameters of the electric submersible pump are formed N samples, form the matrix. For each sample $X = [x_1, x_2, \dots, x_n]$, We extracted individual features using the statistical methods listed in Table 2-1 below. Finally, each N samples Converted to a matrix of $11 \times N$. A complete data collection matrix can be denoted as $R^{M \times n_p}$, inside $M * n_p$ time samples of the resulting parameters.

Table 2-1. Statistics of the characteristics

Feature	Formula	Statistical significance
Average	$X = \frac{1}{N} \sum_{i=1}^N x_i$	Trends in the dataset
Maximum	$X_{\max} = \max\{x_i\}$	Maximum boundary of the data distribution
Minimum	$X_{\min} = \min\{x_i\}$	Minimum boundary of the data distribution
Variance	$D^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{X})^2$	The degree of deviation between the feature variable and the mean
Standard deviation	$D = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{X})^2}$	Reflect the degree of dispersion of the data
Mean Square Error	$MSE = \frac{1}{n} \sum_{i=0}^n (Y_{Ture}[i] - Y_{Pred}[i])^2$	Reflect the prediction accuracy of the model
Root-Mean-Square-Error	$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^n (Y_{Ture}[i] - Y_{Pred}[i])^2}$	
Standard Root Mean Square Error	$NRMSE = \frac{\sqrt{\frac{1}{n} \sum_{i=0}^n (Y_{Ture}[i] - Y_{Pred}[i])^2}}{Y_{\max} - Y_{\min}}$	
Absolute Error	$IAE = \sum_{i=0}^n (Y_{Ture}[i] - Y_{Pred}[i])$	

3.3. Network parameter determination

The choice of hyperparameters of ESN network is important for involving optimal network solving, and a good way to find these parameters is to grid search for different parameters. The grid search runs the network with different hyperparameter values and compares the results to find the best hyperparameter with minimal generalization error. Here, the ESN network in this paper uses the particle swarm optimization algorithm (PSO) [57] when selecting parameters, which can efficiently optimize the leakage rate of ESN network; the main structure of the algorithm is the number of particles, dimension, iteration number and the initial value of position and speed. The principle is to find the optimal solution by simulating particles cooperating in the search space. The algorithm process is as follows:

Let each particle randomly initialize the position and velocity in the hyperparameter space (within the range of the corresponding leakage rate or the range of the value of the number of neurons), $x_i(0)$ and $v_i(0)$ the initial position p_i and the global optimal position of the particle g_i (initially set to infinity).

The corresponding speed update formula is:

$$v_i^{new} = w \cdot v_i^{old} + c_1 \cdot r_1 \cdot (p_i - x_i) + c_2 \cdot r_2 \cdot (g_i - x_i) \quad (2-9)$$

w is the inertial weights that control the influence on past historical speed; c_1 and c_2 are the learning factor, usually set to 2, adjusting the attraction to the current best and global best positions; r_1 and r_2 are the random numbers in the [0,1] range.

The location update formula is:

$$x_i^{new} = x_i^{old} + v_i^{new} \quad (2-10)$$

For each particle, the ESN network is trained using its location, usually using the mean square error (MSE / NRMSE, see Table 2-1) to indicate fitness in compliance with:

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2 \quad (2-11)$$

For each iteration, repeated fitness evaluation and particle information are updated until the preset number of iterations or fitness convergence is reached, and the final output optimized leakage rate records the corresponding best MSE for training the final ESN network model.

The PSO algorithm used in this paper, and the defined parameters used are shown in Table 2-2.

Table 2-2. Definition parameters of the PSO algorithm

Parameter Name	
Particle populations	50
Dimensions	2
Iterations	30
Initial values for the position and velocity (leakage rate)	(0.2, 0.02)
Initial values of position and velocity (number of neurons)	(50, 50)

(1) Disclosure rate

The value range of the leakage rate is between (0,1), so when looking for the best leakage rate of the ESN network

model, the leakage rate is searched through a large gap, and the corresponding fitness is recorded. As shown in Figure 2-5, in the line diagram of the leakage rate and fitness (MSE) of large gaps, the optimal leakage rate of the network is around 0.3.

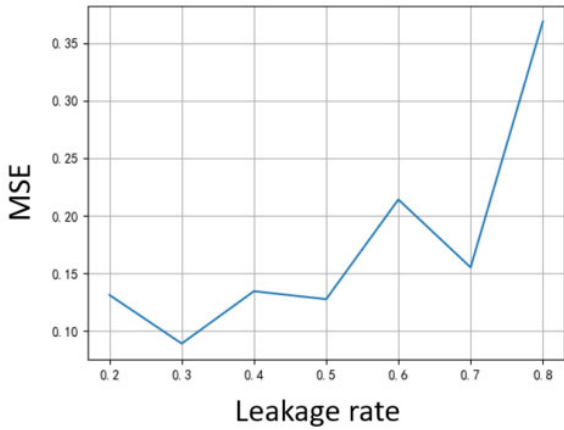


Figure 2-5. leak rate search

Next, narrow the value gap of the leakage rate, run the code again, and search the area near the leakage rate of 0.3.

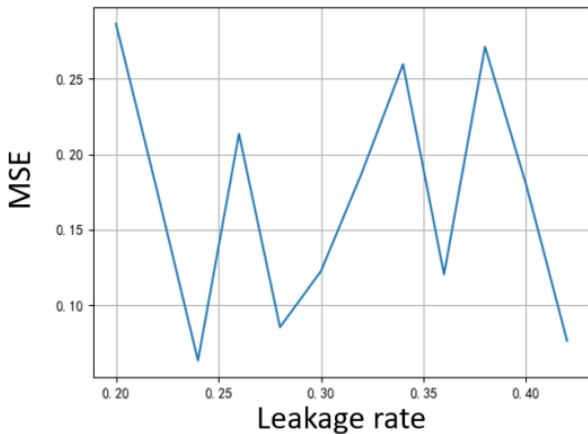


Figure 2-6. A leak rate search for ordinary methods

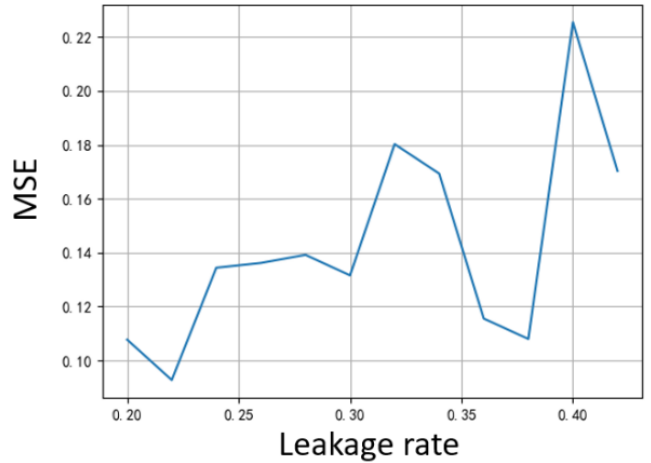


Figure 2-7. A leak rate search using the PSO algorithm

When the value gap of the leakage rate search is taken to 0.02, it can be observed that the best leakage rate obtained by the conventional method is 0.24, while the corresponding MSE value is 0.0794, and the best leakage rate optimized by PSO algorithm is 0.22, and the corresponding MSE value is 0.0898. In order to provide better ESN dynamic expression characteristics, it is more appropriate to choose 0.22 as the leakage rate of ESN network with a small MSE gap.

(2) The number of neurons

The choice of the number of neurons in the hidden layer of an ESN network includes finding the minimum number because it can directly affect the computational efficiency of the entire network while minimizing the error of the network prediction. As shown in Figures 3-10, a bar plot of the running time of the ESN network as a function of the number of neurons. As the number of neurons increases, the running time of the network increases accordingly. This is because more neurons mean that more computational resources and time are spent on processing and transmitting information. However, we cannot simply pursue the minimum number of neurons. In practice, we need to find a balance between computational efficiency and predictive accuracy.

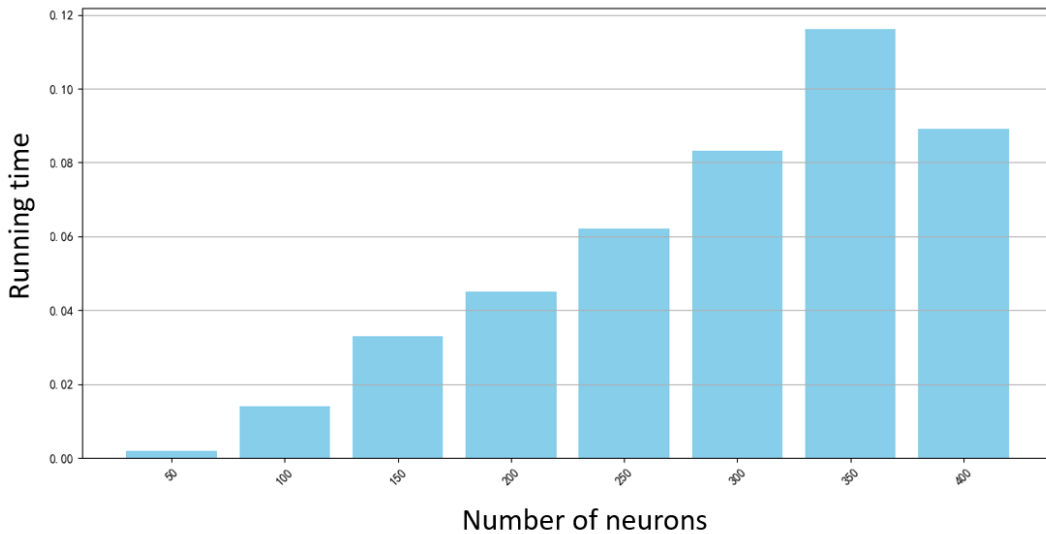


Figure 2-8. Comparison of network operation data under different number of neurons

The Particle Swarm algorithm (PSO) considers the balance between the complexity of the network and the prediction

performance when searching for the optimal number of neurons. With multiple iterations and repeated testing, we

finally determined the best value of 300. This value ensures the prediction accuracy of the model while effectively avoiding overfitting and waste of computational resources. As shown in Table 2-3, the standard root mean square error (NRMSE) values of the network after operation with different numbers of neurons demonstrate this optimization process. By comparing the network performance with different

numbers of neurons, when the number of neurons is 300. The network achieves the best balance between prediction accuracy and computational efficiency. This result not only verifies the effectiveness of the PSO algorithm in the optimization of neural network structure, but also provides ideas for subsequent studies.

Table 2-3. Network parameters of the ESN

Number of neurons	NRMSE
50	0.1772
100	0.1751
150	0.1750
200	0.1786
250	0.1768
300	0.1727
350	0.1789
400	0.1731

(3) Sampling rate

It is crucial to ensure the accuracy of sampling rate, which is directly related to whether the dynamic behavior of the system can be effectively captured effectively. Especially for the sensor inside the electric submersible pump, a reasonable sampling frequency can make it have a higher work efficiency. If the sampling rate is too low, it may cause some important dynamic changes not to be captured, because the interval between the data points is too large to capture those subtle changes. In Figure 2-9, we can see the change of the well

bottom pressure data of the electric submersible pump at different sampling frequencies. By comparing the cases of 12 sampling and 6 sampling per minute, we can clearly observe that the data image slope change of 12 sampling per minute shows a higher degree of nonlinearity. This further demonstrates the importance of a higher sampling rate in capturing the dynamic changes of the system. Instead, a lower sampling rate may cause some important dynamic changes to be ignored, thus affecting an accurate assessment of the system state.

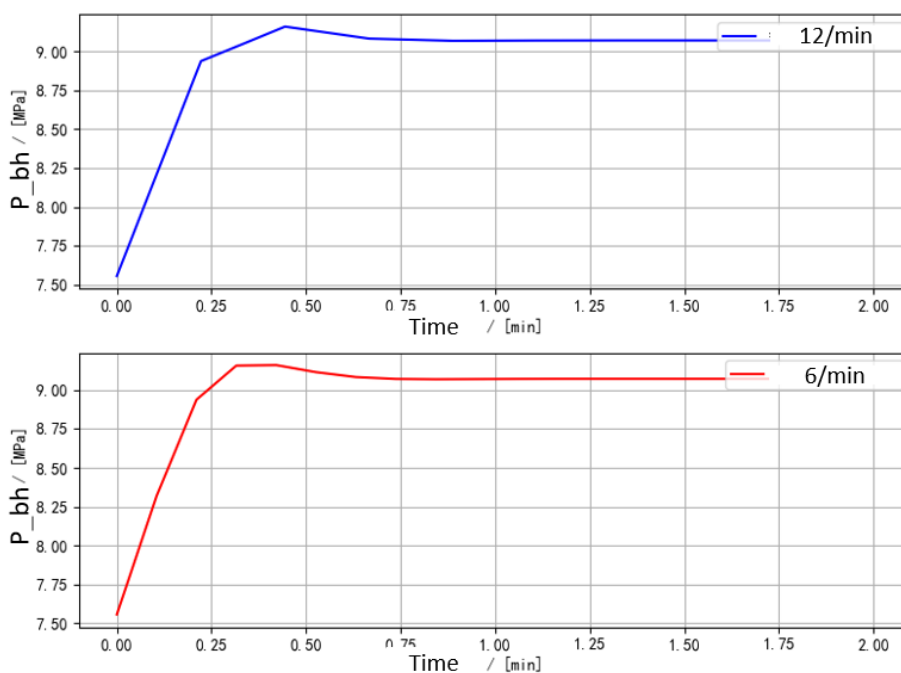


Figure 2-9. Comparison plots of the different sampling rates

(4) Enter the scale factor

ESN networks use the hyperbolic tangent function as an activation function and handle complex nonlinear relationships. When processing a large amount of fluctuating data, using small input scaling factor can avoid gradient disappearance and ensure stable network performance. The ESN parameters are shown in Table 3-4, which are gradually adjusted to accommodate the data complexity, and the slow fitting process improves the model stability and accuracy, leading to better performance in practice.

Table 2-4. Network parameters of the ESN

Parameter name	
Sampling Rate	12times/min
Enter the scaling factor	0.1
Disclosure rate	0.22
Number of neurons	400
Spectral Radius	0.99
The offset	0.1

4. Model Training and Validation

4.1. Network parameter determination

The parameter configuration of the network and the implementation of the specific algorithm are constructed through python3.9. Based on this, the corresponding training dataset was created according to the actual type of submersible pump wells. The data were generated using the NumPy and CasADi open-source libraries in the PyCharm development environment. One of the control system design objectives of the ESN network is to enable the system to reach a specific reference value after multiple training, and to be stably maintained at this reference value. However, in practice, there is an unavoidable problem: when the system reaches a steady state, there will be a certain deviation between the predicted value of the ESN network and the device itself. This situation typically occurs when the input in the training set changes too frequently. To make the ESN network better adaptable, the datasets are created according to the different dynamic properties of the input signals, and the stochastic input signal equations are shown in equations 3-1. In this way, the ESN network can learn both fast and slow dynamics, thus achieving a low error steady state relative to the target object and effectively tracking rapidly changing signals. In further improving the generalization ability and accuracy of the ESN network models, each dataset contains a larger training set and a smaller validation set. Ensure that the model can fully learn the characteristics of the data during the training process, and can verify the performance of the model during the test stage.

$$x_t = x_{\min} + (x_{\max} - x_{\min}) \times \tau \quad (3-1)$$

Where, x_{\min} , x_{\max} is the minimum and maximum values

of the control input respectively; τ is the random number generated through the NumPy library; x_t is the next input.

The initial values that can be set according to the actual situation of the electric submersible pump well are shown in Table 3-1. The random sequence can adjust the amount of change according to different nodes, that is, the value of each input is not fixed, but it can be changed according to the initial value, so as to be closer to the actual working environment in the training process.

Table 3-1. Initial quantities of the training data

Input Data	Initial quantity
p_{wh0} /Mpa	7.4
p_{bh0} /Mpa	3.0
q /(m ³ /h)	30

The change frequency control of the values of the training dataset is shown in Table 3-2, which can intuitively adjust the change pattern of the training data.

Table 3-2. Initial quantities of the training data

	The ttle valve control z /%	Current frequency control f /Hz
x_{\min}	0.1	40
x_{\max}	1	60

The generation effect of the training data set after creation is shown in Figure 3-1 and 3-2. The training data set in the figure below contains 1200 data, and each data can make the system reach the steady state. Figure 3-1 shows that the training dataset has a faster change frequency for every 20 data samples; Figure 3-2 shows that the training dataset has a slower change frequency for every 50 data samples.

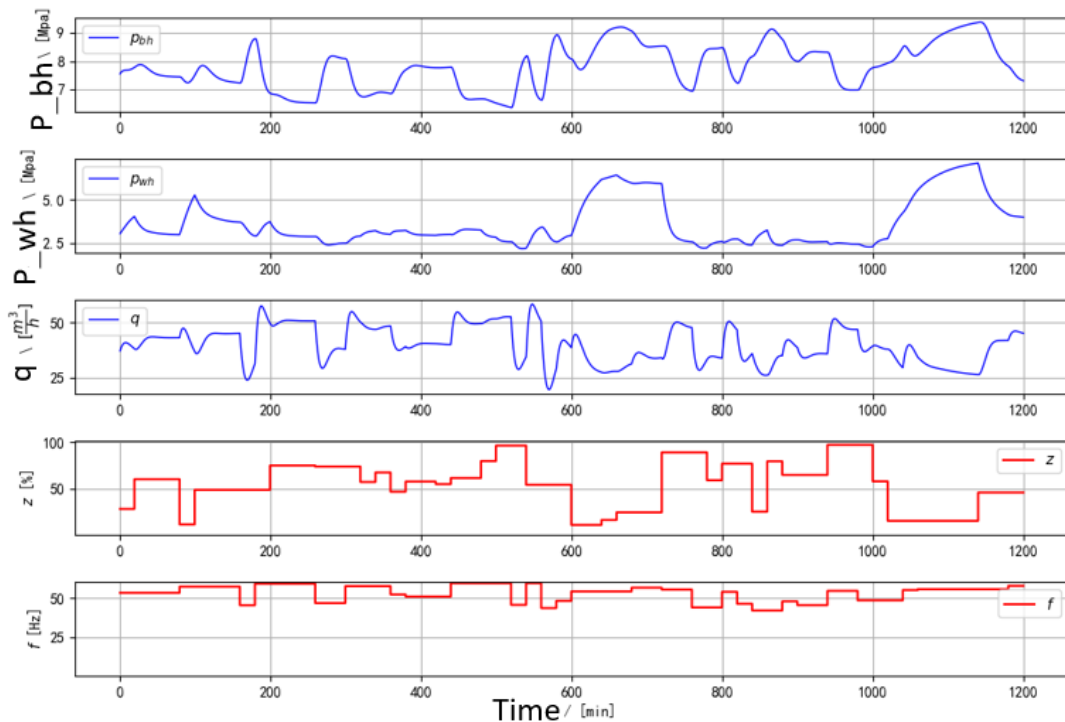


Figure 3-1. Dataset plots of the faster change frequencies

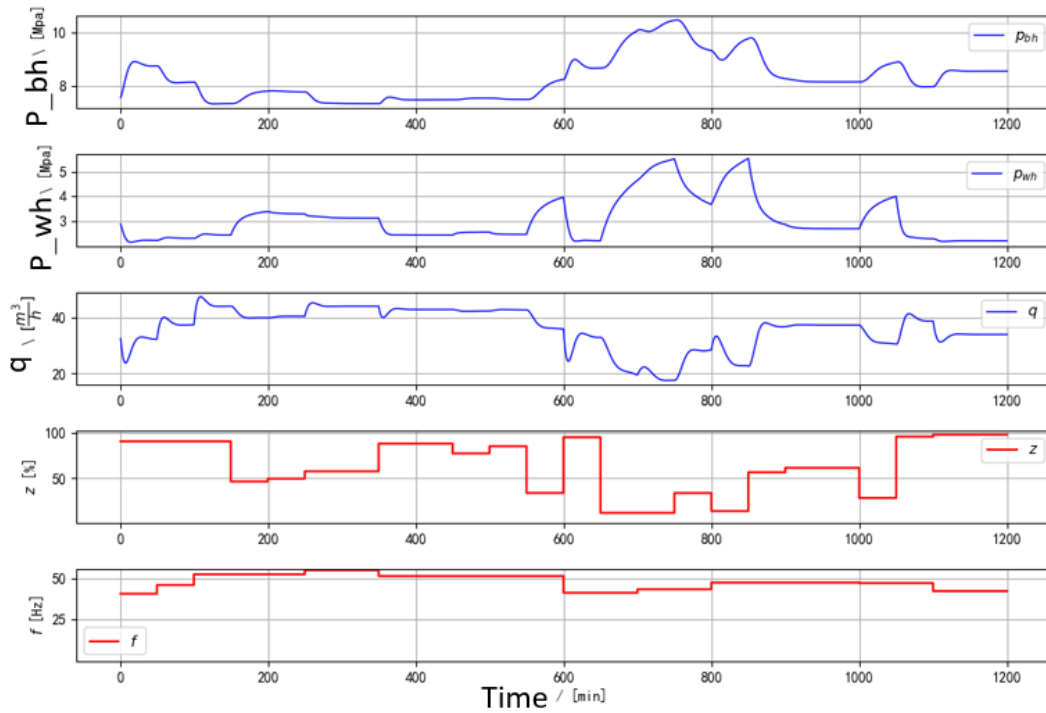


Figure 3-2. Dataset plots of the slower change frequency

To verify the rapid response capability of the ESN network, data samples containing different change frequencies were designed when generating the training dataset. Half of the data samples in this dataset change once for every 20 data samples, and the other half change once for every 50 data

samples. This design aims to simulate data variation at different speeds in order to more comprehensively evaluate the performance of the ESN network. As can be seen in Figures 3-3, the ESN network still has the ability to respond quickly when receiving faster or slower changes in the data.

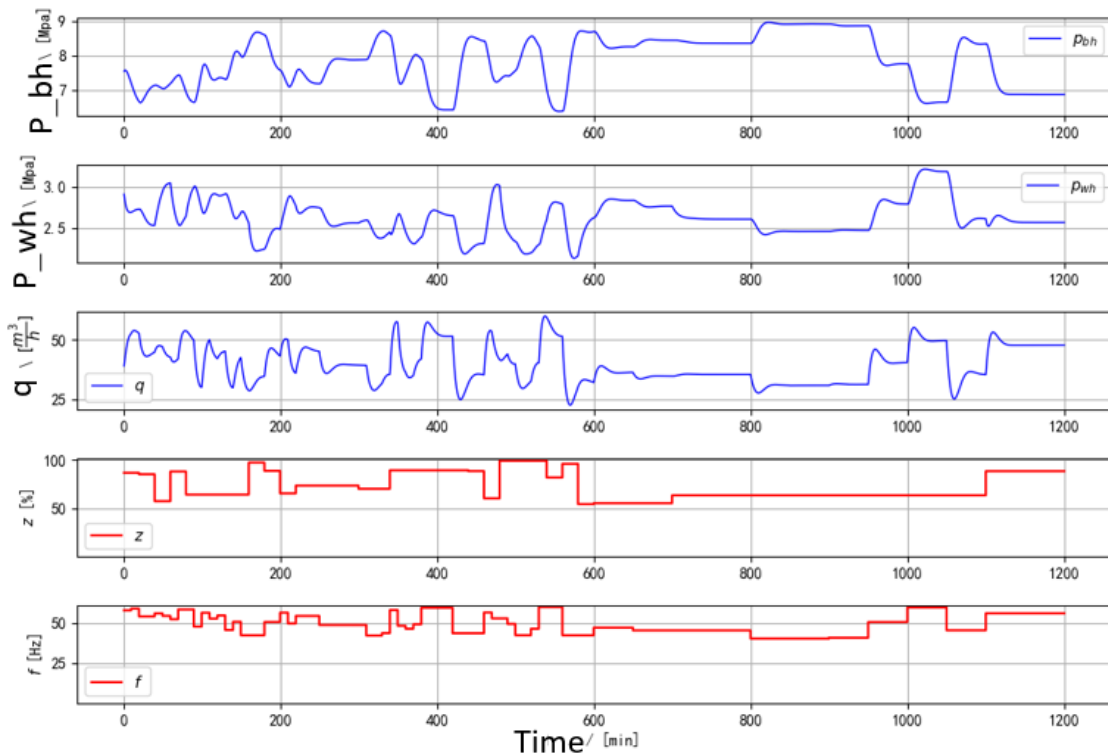


Figure 3-3. Data figures combining changes in speed and speed

4.2. Contrast validation with other neural network methods

In order to ensure that the ESN network model can provide real and reliable prediction results in the prediction task, the

long and short memory network model (LSTM) and 48 gated cycle unit (GRU) network model are written and verified. In this way, we can evaluate whether the prediction performance of ESN model can be better than other recurrent neural network models. These comparison results provide strong

support for the application of ESN network in electric submersible pump model prediction control, ensure the authenticity and credibility of the prediction results, and lay a foundation for the future application in other industrial process control.

Long short memory network model (LSTM) is a special kind of recurrent neural network structure, it by introducing the gating mechanism to solve the gradient in traditional loop neural network, LSTM network through three door unit: forgotten door, input and output door to control the flow of information, makes the network to learn long-term dependence. The forgetting gate is responsible for determining the discard state of the information in the cell; the input gate controls the addition of new input learning; and the output gate determines the output of the next hidden state. This structure allows LSTM to perform relatively well when

handling long sequence data, especially for tasks that need to capture long-term dependencies in time series data.

The gating cycle unit (GRU) model reduces the complexity of the model by simplifying the gating structure of the LSTM model. The GRU model consists of two gating units: reset door and update door. The reset door is responsible for determining how much past information needs to be forgotten, while the update door controls how much new information is combined with information. This design allows the GRU to effectively balance computational efficiency and performance when capturing long-term dependencies in time-series data.

After the construction of the above model, the imported data set is the actual data collected in the first 120 minutes of a random set after the submersible pump well. The comparison results of ESN model, LSTM model and GRU model are shown in Figure 3-4.

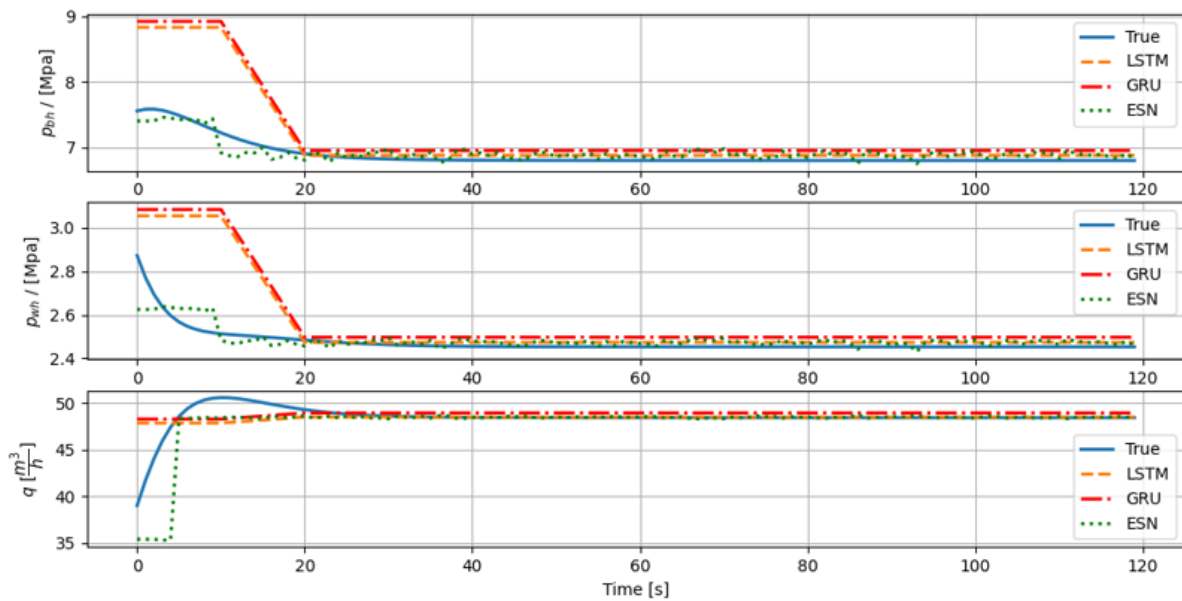


Figure 3-4. Schematic of the prediction comparison between three network models and true values

The results of MSE, NRMSE as standard quantities according on the comparison results are summarized in Table 3-3.

Table 3-3. Table of results between ESN and LSTM and GRU networks

Network type	MSE	NRMSE	Training times	Training duration(s)
ESN	3.90 e 9	0.0082	-	21.5
LSTM	9.19 e 10	0.0203	100	197.5
GRU	9.06 e 10	0.0397	160	218.7

As can be clearly seen from the data results and images of Figure 3-14 and Table 3-3, it can be seen that the ESN model has certain advantages over LSTM model and GRU model in terms of prediction accuracy. The ESN network outperforms LSTM and GRU model in terms of mean square error (MSE) and standard root mean square error (NRMSE), especially when handling electric submersible pump data with complex dynamic characteristics. In addition, ESN model also shows obvious advantages in computational efficiency, and its training time is much lower than LSTM model and GRU model, which is particularly important in real-time control and online optimization scenarios. Through these comparisons, the ESN model has a better application prospect in the predictive control of the electric submersible pump model.

5. Model Performance Assessment

5.1. Model evaluation

In the process of model adjustment and optimization, this paper focuses on the generalization ability of the model, namely the adaptability under different working conditions and environments. By adjusting network parameters such as leakage rate, number of neurons, sampling rate, and input scaling factors, we find that the resulting prediction accuracy and stability of the model are significantly improved. The training process of the model is monitored in real time, which ensures that the convergence of the training process and the risk of overfitting of the model are minimized. Next is the model performance evaluation phase:

The performance of the model on unknown data is evaluated by dividing the training data into training set and test set; as shown in Figure 4-1,4-2,4-3. According to the

prediction results of the test set, the ESN model performs well in the state prediction of fast and slow input changes, and the deviation from the predicted value is as expected.

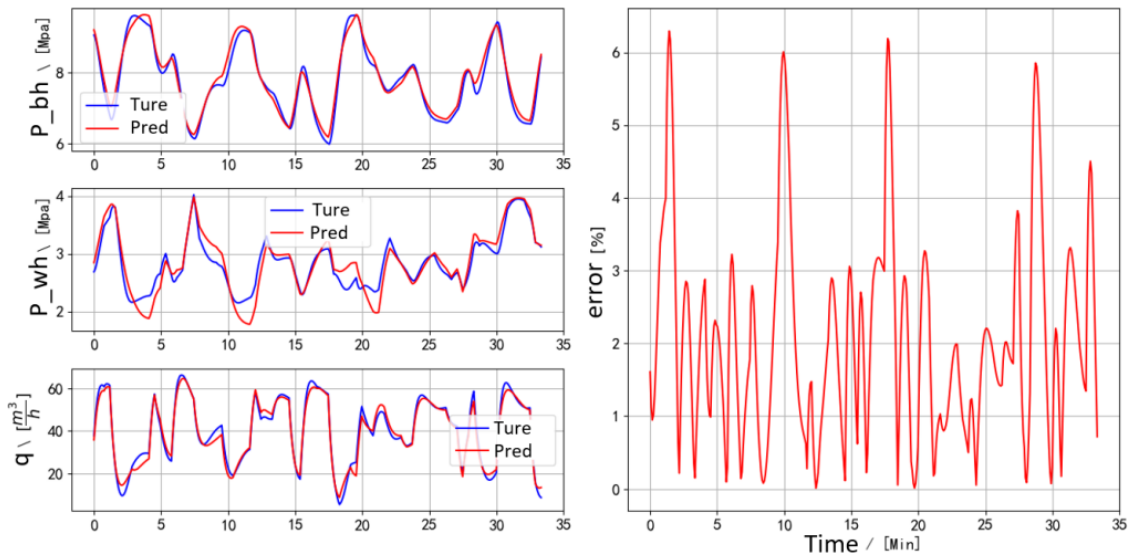


Figure 4-1. Comparison of predicted value and true value and error rate of rapid input data

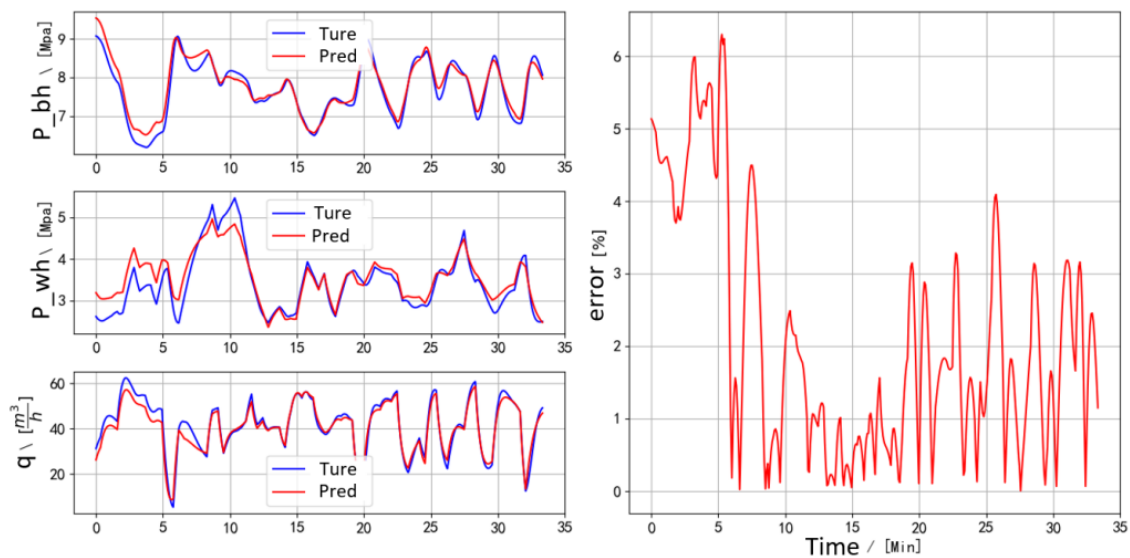


Figure 4-2. Comparison of predicted and true values and error rate when the slow input data changes

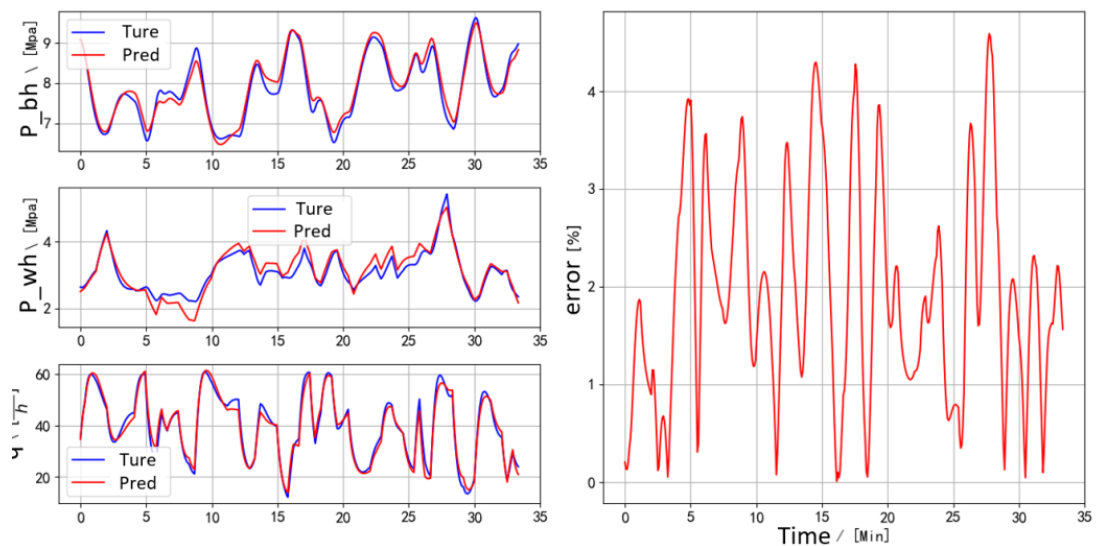


Figure 4-3. Comparison of predicted and true values and error rate between fast and slow input data

When comparing the different results in this subsection, the best results were obtained when using the training set of fast and slow changes combined with the input. In Table 4-1, the MSE is used to measure the error. From the above comparison,

the training set selecting fast and slow change combined with the input was determined as the most appropriate training set for this ESN network.

Table 4-1. Comparison of the effect of input data change speed on prediction accuracy

MSE	P_{bh}	P_{wh}	q
Quick input	1.81×10^{-11}	4.14×10^{-11}	8.99×10^{-7}
Slow input	1.42×10^{-10}	1.88×10^{-11}	1.38×10^{-7}
Mixed input	3.93×10^{-9}	1.72×10^{-10}	3.15×10^{-9}

5.2. Experimental results based on the field data

After some training, the actual data of the electric

submersible pump well is introduced into the ESN network. The true value and predicted value and the corresponding mean error rate image are obtained in Figure 4-4.

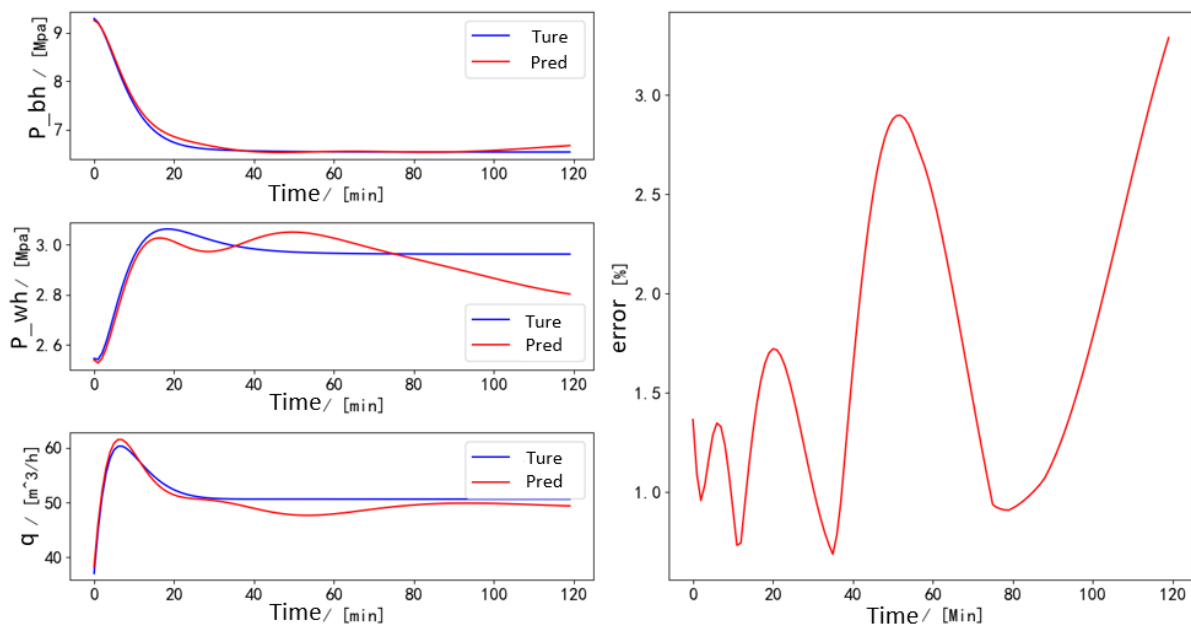


Figure 4-4. Comparison plots of real data and predicted values and the corresponding error plots

Looking at Figure 4-4, we can find that the ESN network has a high matching degree between the actual data and the real value. Even when the data fluctuates greatly, the tracking ability of the model is still excellent. This indicates that the ESN network performs well in dealing with the non-linear and time-varying characteristics of the electric submersible pump system. The average error rate combined with the MSE

data collected in Table 4-2 also confirmed the accuracy of the model, with the error rate maintained at a low level, which further supports the effectiveness and stability of ESN network in the prediction of electric submersible pump well data. In conclusion, the ESN network model shows its significant advantages and reliability in the prediction application of submersible pump well data.

Table 4-2. Comparison table of the influence of input data change speed on prediction accuracy

MSE	P_{bh}	P_{wh}	q
	3.47×10^{-9}	4.91×10^{-9}	1.87×10^{-7}

6. Conclusion

By comparative analysis of ESN network model with other neural network methods, we can conclude that ESN has obvious advantages in handling the electric submersible pump well data prediction task. ESN network is not only more efficient in model construction and training, but also shows good generalization ability and stability in practical application. Moreover, the online learning ability of the ESN network enables it to adapt to the dynamic changing

characteristics of the electric submersible pump system, which has important practical significance for monitoring and controlling the production process of the electric submersible pump well in real time. Therefore, ESN network model has broad application prospect and popularization value in the field of submersible pump well data prediction.

References

- [1] Ma, Q., Sun, S., & Li, Y. (2019). Echo state network for multi-step ahead wind speed forecasting[J]. *Energy Conversion and Management*, 180, 783-795.
- [2] Jean P. Jordanou, Iver Osnes, Sondre B. Hernes, Eduardo Camponogara, Eric Aislan Antonelo, Lars Imsland, Nonlinear Model Predictive Control of Electrical Submersible Pumps based on Echo State Networks[J], *Advanced Engineering Informatics*, Volume 52,2022,101553,ISSN 1474-0346.
- [3] Binder, B. J. T. Kufolalor, D. K. M. Pavlov, A. & Johansen, T. A. Embedded Model Predictive Control for an Electric Submersible Pump on a Programmable Logic Controller[J], 2014 IEEE Conference on Control Applications (CCA), Juan Les Antibes, France, 2014, pp. 579-585.
- [4] Strand S. and Sagli, J. R. MPC in Statoil – advantages with in-housetechnology, in *Int. Symp[J]. Advanced Control Chemical Processes(ADCHEM)*, Hong Kong, 2003, pp. 97–103.
- [5] ABB Automation products AC500, CP400, CP600, DigiVis 500, Wireless. [Online]. Available: www.abb.com
- [6] Delou, P. de A. Azevedo, Julia P. A. de, Dinesh Krishnamoorthy, Maurício B. de Souza, Argimiro R. Secchi, Model Predictive Control with Adaptive Strategy Applied to an Electric Submersible Pump in a Subsea Environment[J], *IFAC-Papers Online*, Volume 52, Issue 1,2019:784-789.
- [7] Fontes, R. M. Santana, Daniel D. & Martins, M A. F. (2022). An MPC auto-tuning framework for tracking economic goals of an ESP-lifted oil well[J]. *Journal of Petroleum Science and Engineering*, Volume 217, 2022, 110867, ISSN 0920-4105.
- [8] Santana, B. A. Matos VS, Santana DD, Martins M. A. F. Embedded MPC Strategies for ESP-Lifted Oil Wells: Hardware-in-the-Loop Performance Analysis of Nonlinear and Robust Techniques[J].*Processes*. 2023; 11(5):1354.
- [9] Lorenz T. Biegler. Efficient Solution of Dynamic Optimization and NMPC Problems[M].*Nonlinear Model Predictive Control*, 2000, Volume 26, ISBN: 978-3-0348-9554-5.
- [10] Victor M. Zavala, & Lorenz T. Biegler. (2009). The advanced-step NMPC controller: Optimality, stability and robustness[J]. *Automatica*, 45(1), 86-93. ISSN 0005-1098.
- [11] Inga J. Wolf, & Wolfgang Marquardt. (2016). Fast NMPC schemes for regulatory and economic NMPC – A review[J]. *Journal of Process Control*, 44, 162-183. ISSN 0959-1524.
- [12] S. Gros and M. Zanon, Data-Driven Economic NMPC Using Reinforcement Learning[J], *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636-648, Feb. 2020, doi: 10.1109/TAC.2019.2913768.
- [13] Rauber T W, de Assis Boldt F, Varejao F M. Heterogeneous feature models and feature selection applied to bearing fault diagnosis[J]. *IEEE Transactions on Industrial Electronics*, 2014, 62(1): 637-646.
- [14] Sofia Eriksson, Jona Nordqvist. The Sherman–Morrison–Woodbury formula for the Moore–Penrose metric generalized inverse[J].(2024)Inverting the sum of two singular matrices. *Results in Applied Mathematics* 22:100463.
- [15] Richalet J,Rault A.Testud et al.Model predictive heuristic control:application to industrial process [J]. *Automatica*, 1978, 14(5): 413-428.
- [16] M. Morari, J. H. Lee.Applications of Nonlinear Model Predictive Control in Complex Industrial [J].*Computers & Chemical Engineering*,1999.
- [17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert. *Constrained Control and Estimation: An Optimized Approach* [J]. IEEE, 2000.