International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 519-531

https://doi.org/10.70517/ijhsa46337

Algorithm Design for Production Data Timing Calculation Analysis and Real-time Business Indicator Recommendation System

Wen Ma[1], Xinyang Zhang[2], Min Xu[2], Mei Zhang[3,*] and Shenzhang Li[3]
[1] Production and Operation Monitoring Center, Yunnan Power Grid Co., Ltd, Kunming, Yunnan, 650000, China
[2] Electric Power Research Institute, Yunnan Power Grid Co., Ltd, Kunming, Yunnan, 650000, China
[3] Information Center, Yunnan Power Grid Co., Ltd, Kunming, Yunnan, 650000, China
Corresponding authors: (e-mail: zhangmei2357@126.com).

Abstract In the context of the digital era, the use of intelligent algorithms to improve production efficiency and decision support is receiving attention from many production enterprises. In this paper, an AttLR-LSTM-based time series model of production data is proposed based on time series data features, combined with LSTM network and attention mechanism to realize accurate prediction of key indexes of production devices, and an intelligent recommender system based on collaborative filtering algorithm is designed to improve the decision support capability and efficiency in the production process. In the comparison experiments with different machine learning models, the prediction effect of this paper's method is improved by 88.91% and 60.92% compared with the LSTM with long-term time series data as input and the LSTM with short-term time series data as input, which fully proves the validity and stability of this paper's method in the prediction of the operating state of production devices. Meanwhile, the real-time business indicator recommendation system designed in this paper not only has high satisfaction, but also receives unanimous praise for its accuracy and confidence.

Index Terms time series, LSTM, attention mechanism, recommender systems, real-time business metrics

I. Introduction

Digital production is an important research direction in the field of intelligent manufacturing nowadays, which refers to the advanced technologies such as big data, cloud computing and artificial intelligence in order to realize the optimization and intelligent management of the production process [1], [2]. The characteristics of digital production data present significant features such as high dimensionality, real-time and dynamics, and the data come from a variety of sources, including sensors, real-time monitoring systems, historical production records and equipment operation information [3]-[5]. The rapid growth and complexity characteristics of these data bring great challenges to traditional data processing and production decision support systems, and there is an urgent need to build an efficient real-time business indicator recommendation system to explore the potential value in production data and support production management decisions [6]-[9].

Digital production involves multiple data sources, and the diversity of these data sources makes the data format and content show a high degree of heterogeneity, which brings challenges to the integration and analysis of data [10]. At the same time, there is a real-time requirement for production data, which requires the system to be able to respond quickly to changes in the production process [11]. For example, monitoring data transmitted by sensors in real time needs to be processed quickly for immediate decision support [12], [13]. Further, digitized production data usually exhibit high dimensionality and contain multiple variables and parameters, such as temperature, pressure, flow rate, chemical composition, etc. The complex interactions of these dimensions require advanced data mining and machine learning techniques to reveal potential patterns and trends [14]-[16].

The article first elaborates the time series model and its representation, commonly used similarity measures. Then it combines the LSTM network for production data time series modeling, and embeds the attention mechanism to enhance the model performance, which enables the model to obtain a fairly accurate prediction result. Subsequently, the diesel production part of a catalytic cracking unit as well as other key indicators are taken as examples to verify the effectiveness of the method proposed in this paper by comparing it with commonly used regression prediction methods. Further, a real-time business indicator intelligent recommender system based on collaborative filtering algorithm is designed, and finally, the proposed recommender system is subjected to performance tests and user satisfaction experiments.

II. Research on modeling and analysis of production timing data

II. A. Time series modeling and characterization

A time series model is a statistical model used to analyze time series data in which the data points are arranged in chronological order. It describes a stochastic process that varies over time, with observations at each moment t combined as time series data. Time series models can be used to predict future trends, detect cyclical changes, and explore relationships with other variables. A time series S is defined as:

$$S=\left\{\left(x_1,x_2,x_3,\cdots,x_t,\cdots,x_n\right)\right\} \tag{1}$$

In Eq. (1), x_t denotes the observation corresponding to time point t.

II. A. 1) Time series representation

Time series representation, i.e., the method of time series dimensionality reduction. Time series dimensionality reduction is to represent the original time series in another space by transforming the original time series to a lower dimensional space or by feature extraction [17]. In time series clustering, dimensionality reduction has an important role. Time series are usually represented in four ways: data-adaptive representation, non-data-adaptive representation, model-based representation and data-based representation methods, the representation of time series is shown in Figure 1.

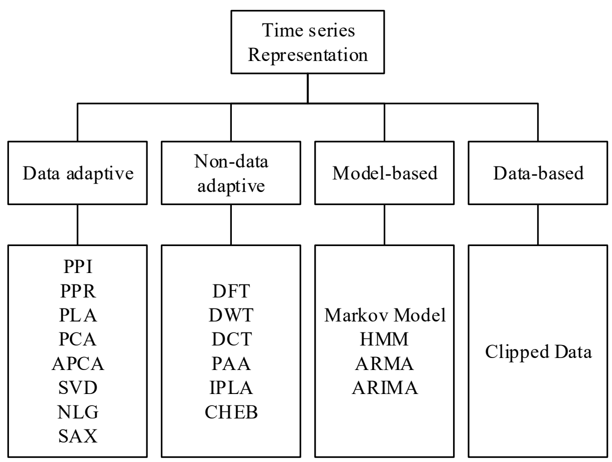

Figure 1: The representation of the time series

The data adaptive representation method performs on all time series in the dataset and tries to minimize the global reconstruction error using arbitrary length (unequal) segments. This technique has been applied to different methods such as segmented polynomial interpolation (PPI), segmented polynomial regression (PPR), segmented linear approximation (PLA), segmented constant approximation (PCA), adaptive segmented constant approximation (APCA), singular value decomposition (SVD), natural language, symbolic natural language (NLG), symbolic aggregation approximation (SAX), and iSAX.The data adaptive representation can approximate each series better, but comparison of multiple time series is more difficult.

Non-data-adaptive methods are a suitable representation for fixed-length (equal-length) segmented time series, and their comparison is usually relatively straightforward. These methods include wavelet transform, discrete wavelet transform (DWT), Chebyshev spectral polynomials, discrete Fourier transform, stochastic mapping, segmented aggregation approximation (PAA), and transposable segmented linear approximation (IPLA).

Model-based methods are a stochastic way of representing time series, which include Markov models, Hidden Markov Models (HMM), statistical models, time series bitmaps, and Autoregressive Moving Average (ARMA). These methods can be data-adaptive, non-data-adaptive, and model-based compression depending on the requirements of specific application scenarios. Users can define the compression ratio according to their applications for more efficient time series storage and processing.

In the data-based approach, the compression ratio is automatically defined based on the original time series. The basic idea is to preserve as much important information in the time series as possible by compressing the time series. These methods usually rely on the characteristics of the data, such as the smoothness, periodicity and trend of the data, as well as the spectral characteristics and autocorrelation of the time series.

II. A. 2) Similarity measures

In the field of time series, the computation of distance is based on approximate matching rather than exact matching. Commonly used similarity measures include Euclidean distance, dynamic time regularization, Manhattan distance, Chebyshev distance, and longest common subsequence. Among them, Euclidean distance and dynamic time regularization are the most commonly used similarity measures in time series clustering. Euclidean distance is a simple and commonly used distance measure for most data types. It can directly calculate the distance between vectors and is easy to understand and implement. Dynamic time regularization, on the other hand, is a more flexible method that can deal with the presence of temporal offsets and distortions in time series. Compared with Euclidean distance, dynamic time regularization is more suitable for dealing with time series data.

(1) Euclidean distance

Euclidean distance is a commonly used distance metric that measures the spatial distance between two points, which is the true distance between two points in Euclidean space. Euclidean distance can be calculated by the following formula:

$$d(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2} \tag{2}$$

where $x=(x_1,x_2,\cdots,x_n)$ and $y=(y_1,y_2,\cdots,y_n)$ denote the coordinates of the two points, respectively.

(2) Dynamic Time Warping

Dynamic Time Warping (DTW) is a time series similarity measure based on dynamic programming, which measures the similarity between two time series that may have different lengths or speeds.DTW matches the corresponding elements in two sequences by bending and stretching the two sequences on their time axes, and calculates their distance or similarity. In DTW, each time series is viewed as a sequence of time points and corresponding eigenvalues.The main idea of the DTW algorithm is to minimize the distance between the corresponding points of the two time series by stretching or compressing the time axis of the sequences and to find an optimal correspondence.The computational process of DTW can be divided into the following steps:

a. Calculate the distance matrix: for two time series X and Y, first calculate the distance matrix between them D, where $D(i,j)$ represents the distance between the i th point in X and the j th point in Y.

b. Calculate the cumulative distance matrix: Define a cumulative distance matrix C, where $C(i,j)$ denotes the minimum distance from the 1st point in X to the i th point, and from the 1st point in Y to the j th point.

c. Initialization: initialize the first row and column of C to positive infinity.

d. Recursive Calculation: Calculate the value of $C(i,j)$ in order from the upper left corner to the lower right corner. The specific calculation is:

$$C(i,j)=D(i,j)+\min\left(C(i-1,j-1),C(i-1,j),C(i,j-1)\right) \tag{3}$$

That is, from the three positions $(i-1,j-1)$, $(i-1,j)$, $(i,j-1)$, choose a cumulative distance of the smallest value, and then add $D(i,j)$ to get the value of $C(i,j)$.

e. Return result: the final $C\left(N,M\right)$ obtained is the DTW distance of sequence X and sequence Y.

Where N and M are the lengths of sequence X and sequence Y respectively. The process of recursive computation can be realized by dynamic programming.

II. B.Relevant Grounded Theory

II. B. 1) LSTM networks

Although RNN will encounter the problem of gradient disappearance in the process of propagation, however, the Long Short-Term Memory Network (LSTM) solves this problem very well, by adding the input gate, forgetting gate and output gate on top of the RNN to make the information selectively pass through, and the gradient is propagated in a linear way after the derivation, so the gradient disappearance problem is avoided, and the contextual information can be retained for a longer period of time. In the internal structure of LSTM, the propagation of information is controlled by three gates i.e., input gate, forget gate and output gate, the input gate mainly allows information to enter selectively, i.e., it decides how much new information enters into the cell's state.

At moment t, the input gate is calculated as shown in equations (4) and (5):

$$\alpha_i^t=\sum_{i=1}^{I}w_{ii}x_i^t+\sum_{h=1}^{H}w_{hi}b_h^{t-1}+\sum_{c=1}^{C}w_{ci}s_c^{t-1} \tag{4}$$

$$b_i^t=f(\alpha_i^t) \tag{5}$$

The forgetting gate mainly determines what kind of information needs to be discarded from the cell in the process of information flow, because new information has to come in and old information naturally has to be forgotten, and the forgetting gate is calculated as shown in equations (6) and (7):

$$\alpha_{\phi}^{t} = \sum_{i=1}^{I} w_{i\phi} x_{i}^{t} + \sum_{h=1}^{H} w_{h\phi} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_{c}^{t-1} \tag{6}$$

$$b_{\phi}^{t} = f(\alpha_{\phi}^{t}) \tag{7}$$

After the above operation, the state of the cell needs to be updated in the way shown in equations (8) and (9), and the output gate is mainly gated to finally decide what information will be output. The output gate is calculated as shown in equations (10) and (11), and based on the filtering of the output gate, the final output result is shown in equation (12):

$$a_{c}^{t} = \sum_{i=1}^{I} w_{ic} x_{i}^{t} + \sum_{h=1}^{H} w_{hc} b_{h}^{t-1} \tag{8}$$

$$s_{c}^{t} = b_{\phi}^{t} s_{c}^{t-1} + b_{i}^{t} g(a_{c}^{t}) \tag{9}$$

$$\alpha_{w}^{t} = \sum_{i=1}^{I} w_{iw} x_{i}^{t} + \sum_{h=1}^{H} w_{hw} b_{h}^{t-1} + \sum_{c=1}^{C} w_{cw} s_{c}^{t} \tag{10}$$

$$b_{w}^{t} = f(\alpha_{w}^{t}) \tag{11}$$

$$b_{c}^{t} = b_{w}^{t} h(s_{c}^{t}) \tag{12}$$

In the equation, the weight matrices (w_{h^*} or w_{*h}) with the symbol h both denote a generalization, representing any line pointing from the previous moment to the current moment. All the variables with symbol a on the left side of the equation represent the results calculated by weighting, while the variables with symbol b represent the values calculated by the activation function. In the above weights, w_{il} represents the weight between the input data x_i^t and the input gate, w_{cl} represents the weight between the cell state of the previous moment and the input gate, $w_{i\phi}$ represents the weight between the input data x_i^t and the oblivion gate, $w_{c\phi}$ represents the weight between the cell state of the previous moment and the oblivion gate, w_{iw} represents the weight between the input data x_i^t and the output gate, and w_{cw} represents the weight between the cell state with the weight between the output gate, w_{ic} denotes the weight of the input layer of the LSTM network [18].

II. B. 2) Attention mechanisms

(1) Structure and Principle of Attention Mechanism

The attention mechanism is an approach proposed to mimic the ability of a person to focus on part of the information while acquiring it. ·

Attention mechanism is equivalent to a weighted calculation for the results of different moments, which establishes a correlation between the current results and the outputs of the previous period of time, making the outputs of different moments have different impacts on the current results, which is also more in line with the actual situation of industrial production, and is calculated as shown in equations (13)-(15):

$$u_t = \tanh(w_w h_t + b_w) \tag{13}$$

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_{k=1}^{T} \exp(u_t^T u_w)} \tag{14}$$

$$s = \sum_{t} \alpha_t h_t \tag{15}$$

where w_w and u_w are the weighting matrices of the attention mechanism, indicating which information needs to be reinforced. u_t is the result of the first weighting calculation, and b_w represents the bias term. h_t represents the input of the attention mechanism which is the output of the LSTM model at different moments, and α_t is the weight that finally acts on the output h_t at different moments.

(2) Improved learning method of attention mechanism

Since the ordinary ATTENTION calculation is relatively simple, it can't get the dependency relationship between temporal data well, especially for the more temporal data such as PTA and ethylene. Therefore, this paper proposes a new calculation method of the attention mechanism namely the maximum attention mechanism. The calculation is shown in Eqs. (16)-(21):

$$att_1 = hh^T \tag{16}$$

$$att_1 = soft\max(att_1) = \frac{\exp(att_1^i)}{\sum_i \exp(att_1^i)} \tag{17}$$

$$att_2 = hHM \tag{18}$$

$$att_2 = soft\max(att_2) = \frac{\exp(att_2^i)}{\sum_i \exp(att_2^i)} \tag{19}$$

$$att_2 = \exp and(att_2) \tag{20}$$

$$att = \max(att_1, att_2) \tag{21}$$

As can be seen from the above formula, where h is the intermediate output of the model, after the softmax calculation of the formula an attention weight of att_1 will be obtained, and then through the calculation of the formula as well as the expansion will also obtain an attention weight of att_2, making the dimensions of att_1 and att_2 are the same. Finally, the maximum value is taken from the corresponding position of the weight matrix of the formula. Compared with the ordinary attention calculation, the attention calculation method designed in this paper can more accurately obtain the dependency relationship between time series data.

II. C.Production data time series modeling

Modeling and analysis of production is a set of processes from data processing to result analysis [19]. The flow of time series modeling and analysis is shown in Fig. 2.

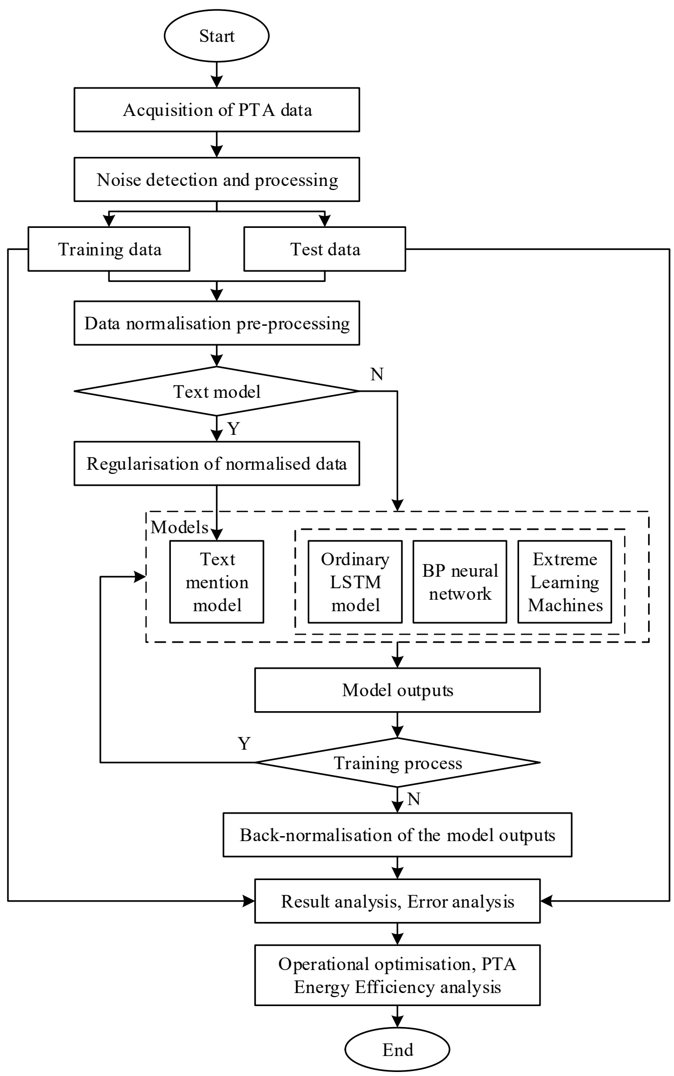

Figure 2: Sequential modeling analysis process

According to the figure, the operational steps for modeling and analyzing the production data are shown below:

Step1 Acquire the data collected by DCSPTA device for 253 time points and divide the training set and test set, in which the training set is 173 entries and the test set is 80 entries.

Step2 Perform the normalization operation on the training set and test set, keeping the data values of each dimension in the range of 0-1.

Step3 For the model AttLR-LSTM proposed in this paper, the data obtained in Step 2 needs to be regularized.

Step4 After the completion of data regularization, the design of AttLR-LSTM algorithm is carried out, and the algorithm is implemented and trained using the deep learning framework.

Step5 Set the termination condition of training, observe the change of error in the process of training, and decide whether to carry out human intervention.

Step6 After the training is completed, the above proposed algorithm outputs the running results of the training data and test data respectively, and performs the inverse normalization process according to the formula to obtain the output data of the same scale as the real data.

Step7 Comparative error analysis of each algorithm according to the output obtained in Step 6, to prove the effectiveness and superiority of the proposed method in PTA data modeling.

Step8 Conduct production analysis based on the results of the model proposed in this paper to determine the optimization scheme of production operation and verify the effectiveness of the optimization. · ·

III. Experiments on time sequence analysis of the operating status of production units

III. A. Data description

The experiments in this section use the diesel part of the input and output data of a catalytic cracking unit as a sample for the study. In this section, single-step predictions of several FCC unit key metrics are performed using the methodology of this paper. The key indicator data are real data collected from the decentralized control system of the FCC unit. The bit numbers of the key indicators to be predicted are TIC2108, TI2074, TI2102, TI3056, and TIC1005, and the related indicators of the key indicators are screened by experts based on the reaction mechanism, with a sampling interval of 3 minutes. In order to cross-sectionally compare the effect of MST-LSTM among different data, the last 3000 data of each indicator are used as the dataset.

Among them, FIC2021 includes 6 relevant indicators such as wax oil consumption, etc. The relevant indicators of TIC2108 include 20 indicators such as light diesel oil out of the device, cold oil slurry out of the device and the top of the fractionating tower, etc. The relevant indicators of TI2074 include 17 indicators such as the pump outlet, the steam inlet, the residue into the raw material oil buffer tank, and the inlet of the steam bag of the slurry steam generator, etc. The relevant indicators of TI2102 include 8 indicators of data of the external dumping of oil slurry, the pump outlet, and the bottom of the fractionating tower. The related indicators of TI3056 include 8 indicators such as supplemental absorbent inlet, hot water exchange buffer tank, etc. The related indicators of TIC1005 include 25 indicators such as the control level of the reactor vapor extraction section, the pressure drop of the reactor to-be-generated ramps slide valve, and the steam outlet, etc. The related indicators of TIC1005 include the control level of the reactor vapor extraction section, the pressure drop of the reactor to-be-generated ramps slide valve, and the steam outlet.

III. B. Performance evaluation

The experiments in this section take FIC2021 as an example for experimental testing, and the prediction results of FIC2021 of this paper's method after 100 iterations of training are shown in Fig. 3, respectively. It can be intuitively found that the prediction results of this paper's method are very accurate, in most cases the model can effectively predict the target value more accurately, and the error rate is 5%, in other words, with an accuracy of about 95%, which is in full compliance with the industry standard of 95% accuracy.

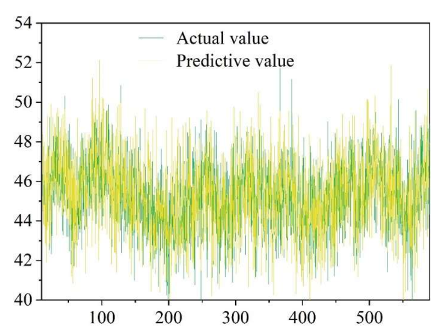

Figure 3: Predictive result

III. C. Time scale impact analysis

In order to further explore the impact of different time scales on the prediction performance of the method in this paper, experiments were conducted in this section with time series of different lengths as inputs to the long time series processing module and short time series processing module, respectively. In conducting the comparison experiments, several regression evaluation metrics are used to ensure the validity of the comparison experiments, which are MAE, Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (MSE).

First, the performance of the model when different length time series are used as long time series processing modules is investigated. The model of different length long time series processing module is defined as LSTM-l-n, and n is the input length of long time series processing module. The length of the short time series processing module is fixed to 10, and time series of lengths 30-80 are selected as inputs to the long time series processing module. The effect of the long time series on the predictive indicator TI2102 is shown in Table 1. Taking point TI2102 as an example, with the growth of the input time series in the long time series processing module, the predictive ability of the model first gradually increases and then gradually decreases, and the performance gap reaches 66.77% at most.

Table 1: Influence of long-term time series on TI2102

Model	MAE	MAPE	RMSE
AttLR-LSTM-l-30	0.5825	0.005	0.7989
AttLR-LSTM-l-40	0.2872	0.0113	0.3912
AttLR-LSTM-l-50	0.2333	0.0015	0.3613
AttLR-LSTM-l-60	0.4304	0.0042	0.5427
AttLR-LSTM-l-70	0.6419	0.0066	0.871
AttLR-LSTM-l-80	0.7021	0.0156	0.9717

Then, this section experimentally analyzes the prediction performance of the model when different lengths of time series are used as the short time series processing module.The model of the short time series processing module with different lengths is defined as LSTM-s-n, and n is the length of the input of the short time series processing module.The LSTM-s-n fixes the length of the long time series processing module as 60, and time series with lengths of 6-15 are selected as the input to the long time series processing module. The effect of short-term time series on the predictor TI2102 is shown in Table 2. Taking point TI2102 as an example, as the input time series in the short time series processing module grows to about 9, the model's prediction remains relatively stable, after which the model's prediction ability decreases with the growth of the input time series. The gap in the predictive ability of the model with the length of the short time series input can vary up to 61.53%.

Table 2: The effect of the short time series on the prediction index TI2102

Model	MAE	MAPE	RMSE
AttLR-LSTM-s-6	0.1787	0.0018	0.3436
AttLR-LSTM-s-7	0.1809	0.0017	0.3012
AttLR-LSTM-s-8	0.2006	0.0324	0.3521
AttLR-LSTM-s-9	0.1582	0.0017	0.2764
AttLR-LSTM-s-10	0.4298	0.0046	0.529
AttLR-LSTM-s-11	0.4414	0.0386	0.5154

III. D. Optimization of model time scale parameters

The input time series scales of the long time series processing module and the short time series processing module have a large impact on the prediction performance of the LSTM model. In order to maximize the performance of the model, Bayesian optimization method is used in this section for hyper-parameter tuning of the long time series processing module and short time series processing module of the method in this paper. The Bayesian optimization results of the prediction model are shown in Table 3.

III. E. Comparative Experimental Results

In order to verify the effectiveness of the model proposed in this paper in predicting the key indicators of catalytic cracking, the method of this paper is compared with Support Vector Regression (SVR), Decision Tree (DT), Random Forest RF, ADABOOST, Multi-Layer Perceptron (MLP), Bootstrap Aggregate Regression, and LSTM. In order to ensure the validity of the comparison experiments, this chapter calculates the regression prediction indexes of input key indicators and their related indicators long time series LSTM and input key indicators short time series LSTM respectively.The prediction results of FIC2021, TIC2108, TI2074, TI2102, TI3056 and TIC1005 are shown in Tables 4 to 9, respectively. As shown in the table, the MAE, MAPE and RMSE of this paper's method show the best performance in several different indicator prediction experiments, which fully proves the prediction performance of this paper's method on the key points of FCC. In the TI2074 indicator dataset, the prediction effect of this paper's method is improved by 88.91% and 60.92%, respectively, compared with the LSTM with long-term time series data as input and the LSTM with short-term time series data as input.

Table 3: Prediction model beeses optimization results

Point bit	Model	MAE	MAPE	RMSE
FIC2021	MST-LSTM-60-10	0.3669	0.0086	0.3548
	MST-LSTM-45-14	0.3567	0.0078	0.3384
TIC2108	MST-LSTM-60-10	0.1786	0.0031	0.2394
	MST-LSTM-81-4	0.1723	0.0049	0.2346
TI2074	MST-LSTM-60-10	0.122	0.0004	0.1504
	MST-LSTM-30-7	0.0783	0.0002	0.1004
TI2102	MST-LSTM-60-10	0.4291	0.0032	0.5255
	MST-LSTM-43-6	0.1968	0.0019	0.3435
TI3056	MST-LSTM-60-10	0.1852	0.0034	0.2578
	MST-LSTM-81-4	0.0778	0.0022	0.101
TIC1005	MST-LSTM-60-10	0.1848	0.0004	0.2305
	MST-LSTM-81-4	0.1607	0.0002	0.2033

Table 4: Predicted results on FIC2021

Model	MAE	MAPE	RMSE
SVR	6.34	0.2	9.006
DT	2.98	0.14	6.128
Bagging regression	1.71	0.1691	3.886
RF	1.83	0.14	5.364
Adaboost	1.95	0.15	4.055
MLP	2.13	0.17	2.599
Long input LSTM	5.6032	0.1183	5.896
Short input LSTM	3.1759	0.1046	4.455
Ours	0.4236	0.0089	0.3511

Table 5: Predicted results on TIC2108

Model	MAE	MAPE	RMSE
SVR	2.0782	0.04	2.713
DT	0.3087	0.0402	0.4332
Bagging regression	0.1153	0.045	0.4958
RF	0.3619	0.05	0.4977
Adaboost	0.3775	0.0406	0.6345
MLP	6.8186	0.1545	8.3337
Long input LSTM	0.2503	0.0055	0.3687
Short input LSTM	0.1554	0.0086	0.5803
Ours	0.1755	0.00336	0.3557

Table 6: Predicted results on TI2074

Model	MAE	MAPE	RMSE
SVR	0.7668	0.0306	0.9205
DT	0.1341	0.1073	0.2056
Bagging regression	0.2544	0.0022	0.2846
RF	0.1142	0.015	0.1436
Adaboost	0.1376	0.0023	0.2256
MLP	1.3738	0.0137	1.8826
Long input LSTM	0.681	0.0022	0.8747
Short input LSTM	0.1932	0.0005	0.2294
Ours	0.0755	0.0002	0.0983

Table 7: Predicted results on TI2102

Model	MAE	MAPE	RMSE
SVR	3.1903	0.0222	3.8404
DT	0.8219	0.0156	1.1264
Bagging regression	1.4973	0.0256	2.011
RF	0.6286	0.0886	0.8489
Adaboost	0.7881	0.0414	1.0446
MLP	0.5867	0.019	0.6743
Long input LSTM	4.4167	0.0223	5.6599
Short input LSTM	0.302	0.0243	0.4374
Ours	0.1982	0.003	0.3422

Table 8: Predicted results on TI3056

Model	MAE	MAPE	RMSE
SVR	0.2518	0.237	0.0481
DT	0.0945	0.1241	0.0501
Bagging regression	0.141	0.1574	0.0353
RF	0.0844	0.1108	0.0315
Adaboost	0.1715	0.1598	0.0621
MLP	0.129	0.0953	0.0171
Long input LSTM	0.1978	0.0177	0.2386
Short input LSTM	0.1703	0.0133	0.2196
Ours	0.1058	0.0334	0.1355

Table 9: Predicted results on TIC1005

Model	MAE	MAPE	RMSE
SVR	1.4102	-0.0016	1.6207
DT	0.6899	0.0025	0.9237
Bagging regression	0.6797	0.0035	0.9359
RF	0.7176	0.0026	0.9751
Adaboost	0.865	0.0018	1.1397
MLP	0.9914	0.0042	1.4462
Long input LSTM	0.6508	0.0009	0.5853
Short input LSTM	0.283	0.0026	0.2341
Ours	0.1616	0.0003	0.2056

IV. Real-time business indicator recommendation system design and experiments

IV. A. Intelligent Recommendation System Design

IV. A. 1) System architecture design

The overall architecture of the business indicator recommendation system can be divided into four main components: data acquisition module, data storage and management module, recommendation algorithm module and user interaction interface module. The data collection module is responsible for collecting real-time production data from multiple data sources, including sensor data, historical production records and external environmental data, etc. Through efficient interfaces and data processing technologies, it realizes rapid acquisition and preliminary cleaning of data. The collected data will be transferred to the data storage and management module. This module adopts distributed database technology to support massive data storage and efficient retrieval, and at the same time, it needs to realize data version control and permission management to ensure data security and consistency. After data storage, the recommendation algorithm module is the core part of the system, based on collaborative filtering algorithm for data analysis and recommendation [20]. The module first performs feature extraction and similarity calculation on the data. Then, the user behavior data is used to generate personalized recommendations to ensure the accuracy and relevance of the recommendations. Finally, the user interaction interface module is responsible for presenting the recommendation results to the user in an intuitive way and supporting the user to make personalized settings and feedback to enhance the user experience of the system. The system design also needs to consider the logical relationship between modules and data flow, through the feedback mechanism of information flow, forming a closed-loop system, so that the recommendation algorithm can be continuously optimized according to the user feedback to improve the recommendation quality and system performance. The basic framework of the system is shown in Figure 4.

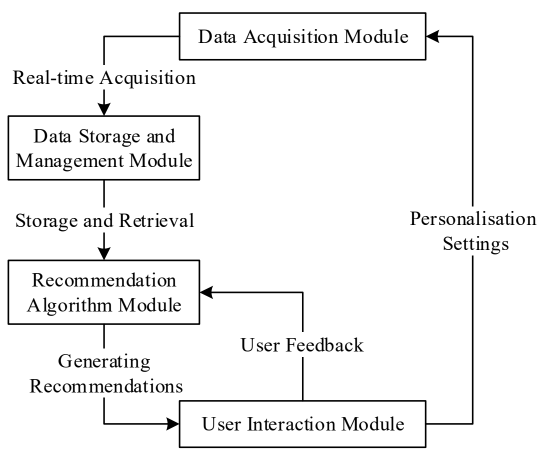

Figure 4: System basic framework

IV. A. 2) Data acquisition and pre-processing

The data acquisition module requires real-time data from multiple sensors and data sources. After data acquisition, data preprocessing must be performed to improve the quality and usability of the data. Data preprocessing includes steps such as data cleaning, data integration, and data conversion. First, data cleaning is used to remove errors and missing values. Assuming that in the collected pressure data, some sensors record abnormal values due to malfunctions, a threshold range, such as 0 kPa to 5000 kPa, needs to be set, and any value outside the range will be labeled as anomalous and deleted or replaced. Second, data integration involves combining data from different data sources into a unified format. For example, a temperature sensor may have data in degrees Celsius while a flow sensor has data in cubic meters per hour, so it is important to standardize them to the same unit for subsequent analysis.

IV. A. 3) Collaborative Filtering Algorithm Implementation

Collaborative filtering algorithms are divided into two main categories: user-based collaborative filtering and item-based collaborative filtering. User-based collaborative filtering relies on the similarity between users, while item-based collaborative filtering focuses on the similarity between items. In this system, item-based collaborative filtering algorithm is chosen, which has the advantage of effectively handling sparse data and improving the accuracy of recommendation.

The first step in the item-based collaborative filtering algorithm is to construct the user-item rating matrix, denoted as R. Where, $R_{u,i}$ denotes the rating of item i by user u. If the user has not rated the item, the value is 0. In order to calculate the similarity between items, it is first necessary to define the similarity measure. The commonly used cosine similarity can be expressed as equation (22):

$$S_{i,j}=\frac{\sum_{u\in U} R_{u,i}\cdot R_{u,j}}{\sqrt{\sum_{u\in U} R_{u,i}^2}\cdot\sqrt{\sum_{u\in U} R_{u,j}^2}} \tag{22}$$

where $S_{i,j}$ denotes the similarity between item i and item j, and U is the set of all users. The formula derives the degree of similarity between items by calculating the correlation between the user ratings of two items.

After that, a recommendation list can be generated for the user by combining the calculated similarity matrix. The recommendation algorithm is mainly based on the user's historical ratings and the similarity of the items, and the formula is shown in equation (23):

$$\hat{R}_{u,j}=\frac{\sum_{i\in I} S_{j,i}\cdot R_{u,i}}{\sum_{i\in I}\left|S_{j,i}\right|} \tag{23}$$

In Eq. (23), $\hat{R}_{u,j}$ represents the predicted rating of item j by user u, and I is the set of items that have been rated by user u.

This formula generates a personalized recommendation for the user by taking into account the user's ratings of similar items in a weighted summation manner.

IV. A. 4) Training and Optimization of Recommendation Models

After the collaborative filtering algorithm is implemented, targeted training and tuning are also required to ensure the optimization of the recommendation effect. In the model training phase, data selection and processing are crucial. Using the user's historical behavior data as the training set, relevant features are extracted and normalized so that the model can better learn the user's preferences. The normalization process usually adopts the Z-score normalization method, and the formula is shown in equation (24):

$$X'=\frac{X-\mu}{\sigma} \tag{24}$$

Eq. (3), X is the original data, μ is the mean of the data, σ is the standard deviation, and X' is the normalized data. This treatment can effectively reduce the influence of the magnitude between different features and improve the training efficiency of the model.

During the training process, a stochastic gradient descent optimization algorithm is used, where the parameters are continuously adjusted through iterations to reduce the loss of the model. An improved loss function is introduced during the training process by integrating the cross-entropy loss function, which is particularly effective in dealing with implicit feedback from users (e.g., clicking, buying behavior). The expression of the cross-entropy loss function is shown in equation (25):

$$CE=-\sum_{(u,j)\in D}\left[y_{u,j}\log\left(\hat{y}_{u,j}\right)+\left(1-y_{u,j}\right)\log\left(1-\hat{y}_{u,j}\right)\right] \tag{25}$$

In Eq. (25), $y_{u,j}$ represents the actual situation (1 or 0) of whether user u has interaction with item j, and $\hat{y}_{u,j}$ is the probability that the model predicts that user u has interaction with item j.

IV. B. Experiments and results

In order to verify the accuracy and convergence of the personalized recommendation algorithm. According to the simulation experiment parameters are set, Q = 41 for the clustering center, fuzzy control parameters are set to b1 = 124, b2 = 364, simulation time is 2min (120s), 3166 running iterations, set the number of 1040 in the space of W individual dimensions, the population size of 10, set the experimental parameters, and begin to carry out the simulation and analysis, and the original dataset sampling is shown in Figure 5.

Test the sample information collected on the user, and compare the convergence and accuracy of the traditional method with the experimental process, to get the comparison curve diagram, personalized recommendation curve comparison shown in Figure 6. Analysis of the comparison results show that the personalized recommendation algorithm proposed in this paper is very good response in terms of user recommendation satisfaction and information accuracy, but also fully reflects the personalized recommendation algorithm proposed in this paper has a better performance.

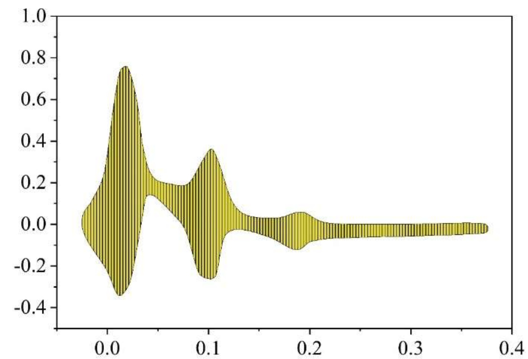

Figure 5: Raw data collection

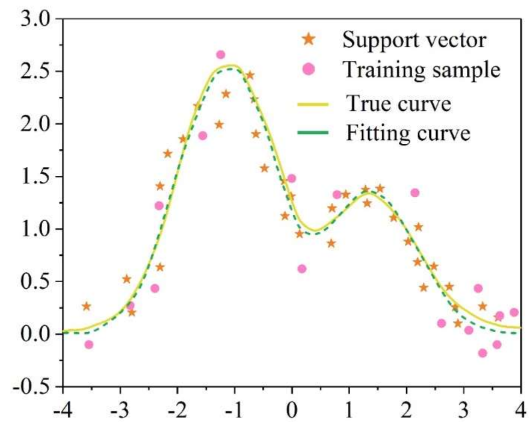

Figure 6: Comparison of personalized recommendation curves

The recommendation method designed in this paper is analyzed and compared with the traditional method, and the PCA algorithm is compared with PSO, and the personalized recommendation satisfaction comparison results are shown in Figure 7. The above experimental results show that the experimental process compared with the traditional method, this paper's algorithm has the highest average satisfaction value of 0.962, the user accuracy information recommendation, to promote the transaction reached, the efficiency has been greatly improved.

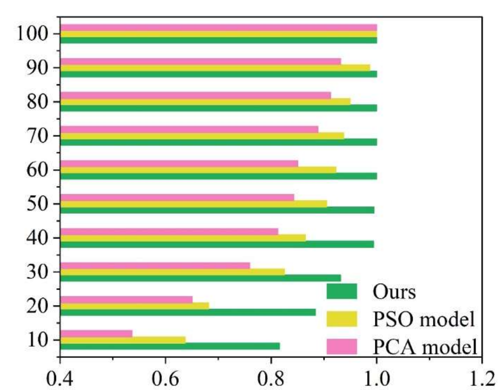

Figure 7: Personalized recommendation satisfaction comparison results

V. Conclusion

The article firstly combines LSTM network and attention mechanism for production time sequence data modeling, and researches a real-time business index intelligent recommendation system based on collaborative filtering algorithm, realizes more efficient and accurate identification of production equipment operation state, and improves the decision support ability and efficiency in the production process, and draws the following conclusions after the test:

(1) After 100 iterations of training, the prediction results of this paper's method on the FIC2021 metrics are very accurate, with about 95% accuracy, indicating that this paper's method fully meets the industry standard of 95% accuracy.

(2) This paper's method shows good performance in comparison experiments. In the TI2074 indicator dataset, the prediction effect of this paper's method is improved by 88.91% and 60.92% over the LSTM with long term time series data as input and the LSTM with short term time series data as input, which proves the efficient prediction performance of this paper's method.

(3) In the satisfaction comparison experiment of intelligent recommender system for time business indicators. Compared with the traditional method, this paper's algorithm has the highest satisfaction, with a mean value of 0.962, thus indicating that this paper's method has good working efficiency.

References

[1] Brauner, P., Dalibor, M., Jarke, M., Kunze, I., Koren, I., Lakemeyer, G., ... & Ziefle, M. (2022). A computer science perspective on digital transformation in production. ACM Transactions on Internet of Things, 3(2), 1-32.

[2] Zhang, Q., Shen, S., Li, H., Cao, W., Tang, W., Jiang, J., ... & Liu, S. (2022). Digital twin-driven intelligent production line for automotive MEMS pressure sensors. Advanced Engineering Informatics, 54, 101779.

[3] Zhang, Y., Zhang, R., Wang, Y., Guo, H., Zhong, R. Y., Qu, T., & Li, Z. (2019). Big data driven decision-making for batch-based production systems. Procedia CIRP, 83, 814-818.

[4] Fani, V., Antomarioni, S., Bandinelli, R., & Bevilacqua, M. (2023). Data-driven decision support tool for production planning: a framework combining association rules and simulation. Computers in Industry, 144, 103800.

[5] Windmann, S., Maier, A., Niggemann, O., Frey, C., Bernardi, A., Gu, Y., ... & Kraus, R. (2015, November). Big data analysis of manufacturing processes. In Journal of physics: Conference series (Vol. 659, No. 1, p. 012055). IOP Publishing.

[6] Guo, Y., Wang, N., Xu, Z. Y., & Wu, K. (2020). The internet of things-based decision support system for information processing in intelligent manufacturing using data mining technology. Mechanical Systems and Signal Processing, 142, 106630.

[7] Gödri, I., Kardos, C., Pfeiffer, A., & Váncza, J. (2019). Data analytics-based decision support workflow for high-mix low-volume production systems. CIRP annals, 68(1), 471-474.

[8] Bakhrankova, K. (2010). Decision support system for continuous production. Industrial Management & Data Systems, 110(4), 591-610.

[9] Kuo, Y. H., & Kusiak, A. (2019). From data to big data in production research: the past and future trends. International Journal of Production Research, 57(15-16), 4828-4853.

[10] Zhou, J., Hong, X., & Jin, P. (2019). Information fusion for multi-source material data: progress and challenges. Applied Sciences, 9(17), 3473.

[11] Chen, L., Peng, C., Yang, C., Peng, H., & Hao, K. (2023). Domain adversarial-based multi-source deep transfer network for cross-production-line time series forecasting. Applied Intelligence, 53(19), 22803-22817.

[12] Fani, V., Antomarioni, S., Bandinelli, R., & Bevilacqua, M. (2023). Data-driven decision support tool for production planning: a framework combining association rules and simulation. Computers in Industry, 144, 103800.

[13] Bousdekis, A., Lepenioti, K., Apostolou, D., & Mentzas, G. (2021). A review of data-driven decision-making methods for industry 4.0 maintenance applications. Electronics, 10(7), 828.

[14] Spanos, A. C., Gayialis, S. P., Kechagias, E. P., & Papadopoulos, G. A. (2022). An application of a decision support system enabled by a hybrid algorithmic framework for production scheduling in an SME manufacturer. Algorithms, 15(10), 372.

[15] Budde, L., Liao, S., Haenggi, R., & Friedli, T. (2022). Use of DES to develop a decision support system for lot size decision-making in manufacturing companies. Production & Manufacturing Research, 10(1), 494-518.

[16] Hansson, L. J., Forsmark, V., Flisberg, P., Rönnqvist, M., Mörk, A., & Jönsson, P. (2022). A decision support tool for forwarding operations with sequence-dependent loading. Canadian Journal of Forest Research, 52(12), 1513-1526.

[17] Shoukun Jiao & Wuyi Ye. (2025). Joint value-at-risk and expected shortfall regression for location-scale time series models. Communications in Statistics - Theory and Methods,54(10),2945-2958.

[18] Wei Zhang,Keke Zhang,Xingxing Li,Jiande Huang,Jiaqi Wu & Yongqiang Yuan. (2025). Clock bias prediction for low Earth orbit satellites with LSTM neural network: method and verification. GPS Solutions,29(3),92-92.

[19] Yucheng Zhan,Yijun Pan,Siying Wu,Yueyi Zhang & Xiaoyan Sun. (2025). Hierarchical Task-aware Temporal Modeling and Matching for few-shot action recognition. Neurocomputing,624,129467-129467.

[20] Tianyu Wang & Dong Ge. (2025). Research on Recommendation System of Online Chinese Learning Resources Based on Multiple Collaborative Filtering Algorithms (RSOCLR). International Journal of Human–Computer Interaction,41(3),1771-1781.