LSTM-based Analysis of Industrial IoT Equipment

Weishan Zhang*, Wuwu Guo*, Xin Liu*, Yan Liu†, Jiehan Zhou§, Bo Li*, Qinghua Lu*, Su Yang‡
*Department of Software Engineering, China University of Petroleum (East China), Qingdao, China.
†Faculty of Engineering and Computer Science, Concordia University, Montreal, Canada
‡College of Computer Science and Technology, Fudan University, Shanghai, China. 200433
§University of Oulu, Finland

Abstract—Industrial Internet of Things (IIoTs) are producing massive data, which are valuable for knowing running status of the underlying equipment. However, these data involve various operation events that span some time, which raise questions on how to model long memory of states, and how to predict the running status based on historical data accurately. This paper aims to develop a method of (1) analyzing equipment working condition based on the sensed data; (2) building a prediction model for working status forecasting, and designing a deep neural network model to predict equipment running data; (3) improving the prediction accuracy by systematic feature engineering and optimal hyperparameter searching. We evaluate our method with real-world monitoring data collected from 33 sensors of a main pump in a power station for 3 months. The model achieves less root mean square error than that of ARIMA. Our method is applicable to general IIoT equipment for analyzing time series data and forecasting operation status.

Index Terms—Time Series Prediction, LSTM model, Power Equipment, Industry Internet of Things.

1 Introduction

Internet of Things (IoTs) are playing important roles in a smart world, like the usage of IoTs for handling emergency situations [1], or for monitoring various industrial equipment, as used in power plants. Industrial Internet of Things (IIoTs) has been heralded as a helpful means of improving operational efficiency. The connected IIoT devices form a new layer that augments equipment intelligence to the edge. The operational status of individual equipment is constantly monitored and informed for their operations.

An example is a power station, which is a complex system equipped with a large number of sensors with diverse types. Figure 1 presents an example of a cooling pump in a power station with 33 sensors. Each sensor is dedicated to a specific measurement, such as pressure, temperature, vibration and so on. These sensors are constantly collecting measurement data in time series. Discovering patterns from these time series allows anomaly detection and trend forecasting to prevent faults and defects in a proactive manner [2] [3].

The original monitoring data is multidimensional with the following characteristics: (1) With the number of sensors increases, processing every time series from each sensor becomes both data intensive and computing intensive; (2) The data has outliers and noises; (3) The time series from different sensors have correlations at various levels, which results in large amount of redundant information. By grouping correlated sequences, it helps to reduce the size of monitoring data to be processed; (4) The running states before downtime usually lasts for a period of time and involves equipment failure states. In addition, after the defected sensor is restored, a grace period also appears in the running state. Therefore, a prediction model should be able to reflect the memory span of such states.

These data characteristics lead to the research questions of

- R1: How to reduce redundant information without degradation of forecasting accuracy?
- R2: How to identify outliers and noises?
- R3: How to model the long memory of states?
- R4: How to predict the running status based on historical data accurately?

The time series forecasting processes such as AR [4], MA [5], ARMA [6], ARIMA[1] depend on parameters resulted from historical time series. These parameters are non- negative integers that refer to the order of the autoregressive part, the degree of first differencing involved, and the order of moving average part. Although these processes can handle non-stationary data, they are limited in memorizing any state for any length of time.

The Long-Short-Term-Memory (LSTM) [7] is one kind of Recurrent Neural Network (RNN). LSTM has long short-term memory blocks that consist of memory cell units. These memory cell units make it possible for LSTM to remember state values for an arbitrary long time. A LSTM block also has three different gate units that learn to keep, utilize, or destroy a state when appropriate. LSTM neural networks have been successfully applied to applications with sequential data such as stock forecasting, natural language processing, and social behavior analysis.

This paper proposes an analysis method based on the LSTM neural network model to predict power station working conditions from data collected from sensors. First, the LSTM model is applied to express the memory span of

1. http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2825538, IEEE Access

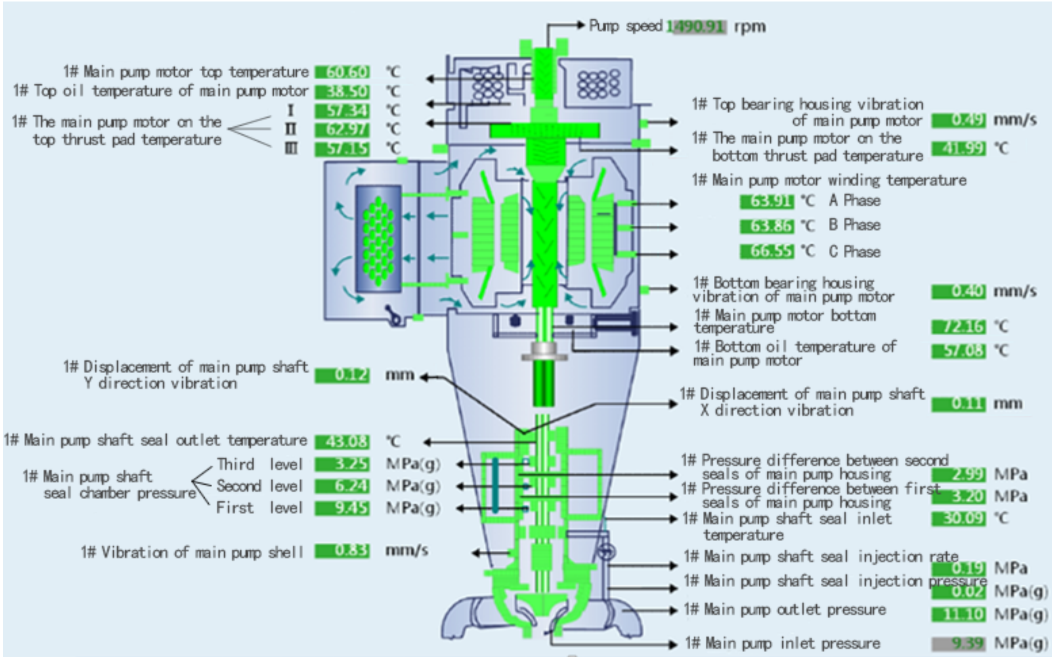


Fig. 1: Measurements and status monitored from a main pump in a power station

power equipment working conditions. We propose to use the orthogonal experimental design method to optimize the LSTM model parameters. Root mean square error is used to evaluate the prediction accuracy.Second, we build a workflow of feature engineering that eliminates the outliers, conducts window normalization of sequence segments, and runs correlation analysis of sequences. This workflow extracts important features as inputs to the LSTM model to forecast faults. To optimize the searching of hyperparameter values that best suit the LSTM model, we apply orthogonal experimental design (OED) to reduce the number of trials. This workflow improves the LSTM model in terms of prediction accuracy as well as the time of training and prediction. We further compare the LSTM model with the ARIMA model. The evaluation shows the improvement of LSTM produce higher prediction accuracy. This paper aims to propose an approach for accurate prediction of equipment running status, which has great significance for reducing the cost of maintenance and safety of equipment on other hand.

The contributions of our research in this paper are three-fold:

- A systematical method that build a prediction model augmented with feature engineering techniques. This method is generally applicable to model the operation states of IIoT equipment and forecast working conditions to improve operation quality.
- The technique of combining correlations of sensors to extract features for inputs to the LSTM model. This technique pertains to intrinsic interactions and patterns of sensors. Thus the technique effectively reduces the complexity of data processing while improve the accuracy of the LSTM model.
- Orthogonal experimental method used to optimize hyperparameters of the LSTM model.

The remaining part of the paper is as follows: Section II reviews the background on the used dataset and also LSTM deep learning model. Section III presents LSTM based time series analysis, with details on the optimization of LSTM model. Section IV presents our experiment to evaluate the proposed solution. Then we discuss the related work in Section V. Section VI gives the conclusion and discuss the future work.

2 Background on the Used Data Set and Deep Learning Model

2.1 Data Set on Cooling Pump

Our work is based on historical monitoring data produced by 33 sensors in the main pump of a power station for three months. As shown in Figure 1, sensors are distributed and measuring a variety of metrics including temperature, pressure, vibration, injection rate and so on. Our study found that the dataset is characterized by redundancy, multiple noise sources, and the correlation between sensor monitoring data. These characteristics affect the accuracy of a forecasting model. Therefore, it is essential to improve the data quality of inputs derived from original data.

2.2 LSTM Neural Network Model

The Long Short Term Memory (LSTM) neural network is a special kind of Reccurent Neural Network (RNN). A LSTM model is capable of learning long-term dependencies. We develop the following steps to build a LSTM model.

The first step in building a LSTM model is to decide on information that is thrown away from cell states. This decision is made by a sigmoid layer called forget gate layer. According to Eq.(1), the LSTM model checks h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. The value of 1 represents the meaning of 'completely keep this' while a value 0 represents the meaning of 'completely get rid of this'.

$$F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

The next step is to decide what information to be stored in the cell state. First, a sigmoid layer called the input gate layer decides which values will be updated (Eq.(2)). Next, a tanh layer creates a vector of new candidate values, $\widetilde{C_t}$, that will be added to the state (Eq.(3)). We combine these two steps to create an update of an old cell state C_{t-1} into a new cell state C_t.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C_t} = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

According to Eq.(4), we multiply the old state by f_t, the ratio of forgetting states; and add $i_t \times \widetilde{C_t}$ to scale each state value to be updated.

$$C_t = f_t \times C_{t_1} + i_t \times \widetilde{C_t} \tag{4}$$

Finally, the output is calculated based on the cell states that are further filtered. A sigmoid layer decides on the cell states to output (Eq.(5)), then the filtered cell states pass through tanh to transform the values in the range of -1 to 1. The final output is the value of tanh that multiplies the output of the sigmoid gate.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = O_t \times tanh(C_t) \tag{6}$$

3 LSTM BASED TIME SERIES ANALYSIS

We propose a LSTM based solution for time series data analysis as shown in Figure 2, which consists of layers of data preprocessing, feature engineering, prediction model development, model optimization, and systematic evaluation.

Data Pre-processing Layer. The data preprocessing layer eliminates outliers and noises for raw data. This stage also carries out normalization within segments of time series sequences. Normalization solves the issue that data values expand overs a large range of time periods and may have different dimensions. This stage ensures that all raw data is converted to the range of [0, 1].

LSTM Input Layer. The working condition analysis of power equipment is based on the long multi-dimensional time series generated by 33 sensors. Correlations be- tween sensors exist at varies levels. We merge time series of sensors with strong correlation to reduce the input di- mensions of the LSTM model. Our purpose is to reduce the computational complexity of the model.

LSTM Embedding Layer. An embedding layer is added to the traditional LSTM network model. Through the mapping relation of this layer, the single dimensional long time series is transformed into multiple fixed short time series. This set of multiple fixed short time sequences is used as the input to the hidden layer for model training.

LSTM Hidden layer. The hidden layer accepts data passed from the previous level. Each hidden layer is actually a feature representation layer. The last hidden layer output becomes the input for the next hidden layer. By iteration, the weight of the hidden layer is continuously adjusted until the network converges.

LSTM Output layer. The output of the LSTM network is the predictive value of each sensors operating conditions.

The network parameters are adjusted and optimized by using the orthogonal experimental design (OED) method. We analyze the influence of the number of neurons, the batch size and the number of iterations on prediction accuracy. The outcomes of the OED method help to find a set of parameter combinations that produce the least prediction errors, which achieve the best accuracy.

Evaluation Layer. Finally, the optimized parameters are applied to the LSTM prediction network, and the root mean square error between the predicted data and the original data is calculated, so that the optimal prediction results can be obtained.

The main improvements to the normal way of using LSTM models are as follows.

- Adding a window normalization to data preprocessing in order to fit for multi-modal data processing as needed in IIOT equipment.
- Adding an Embedding Layer to change mapping in LSTM. An embedding layer is added to the traditional LSTM network model. Through the mapping relation of this layer, the single dimensional long time series is transformed into multiple fixed short time series. This set of multiple fixed short time sequences is used as the input to the hidden layer for model training.
- Using OED experiment to obtain optimized model parameters.

3.1 Data Preprocessing

3.1.1 *Data Cleaning*

The original data contains noise and outliers that are caused from the equipment downtime interference and sensor errors. The left part of Figure 3 shows a sequence of raw data with outliers. To filter these noise and outliers, we first detect the outliers by using k-nearest neighbor clustering. The outlier of a data point in the time series is given by the distance to its k nearest neighbors. The threshold of detecting outliers is set by the average distance of k-nearest neighbors.

3.1.2 *Data Normalization*

Data normalization is used to make sure that all data is in the appropriate scale of values. The traditional normalization method is direct global normalization [8]. For example, the data can be limited by the maximum and minimum

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2825538, IEEE Access

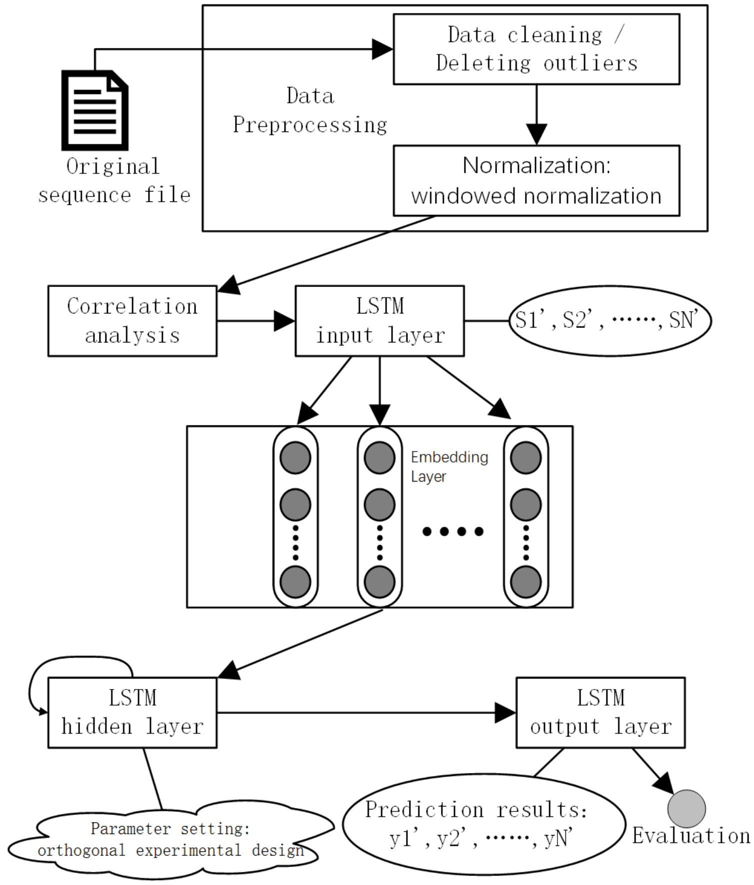


Fig. 2: Workflow of the LSTM-Based time series analysis

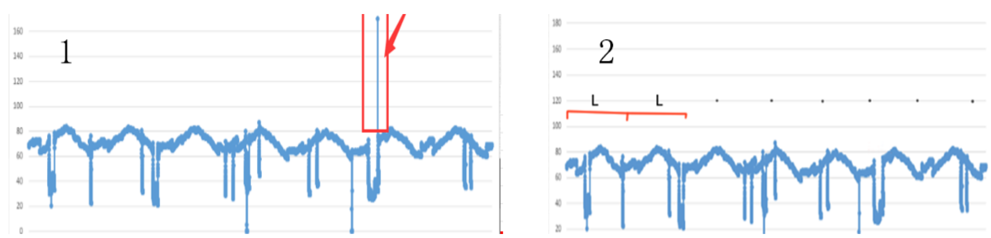


Fig. 3: Left: Locate and Delete Outliers. Right: Segmentation.

values in the global range, using the following formula to calculate.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{7}$$

This method is simple, however it can not handle time series that have various range of scales. In fact, sensors are updated from time to time, and measurement units and scales are different. Therefore, the global normalization can cause local segments deviate dramatically compared to their local scales. Therefore we develop a normalization method that utilizes the windowed wavelet/Fourier transformation to normalize segments of time series, referred as window normalization in this paper. Using window normalization, the time series in each window is normalized by the maximum and minimum values of the segments. This window normalization method is described in details as follows:

Given a time series S windowed and segmented, S is composed of a sequence of n segments and each segment is with a window length of L. The value of L depends on the frequency variation of the time series. The diagram is shown in the right part of Figure 3.

$$S = \{S_1 ... S_i ... S_n\} \tag{8}$$

$$S_i = \{x_1, x_2 ... x_L\} \tag{9}$$

After normalization of Si, the sequence becomes

$$S^{'} = \left\{x_1^{'}, x_2^{'} ... x_{L \cdot i}^{'} ... x_{L \cdot n}^{'}\right\} \tag{10}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2825538, IEEE Access

3.2 Time Series Correlation Analysis

We analyze the correlation matrix between sensors and illustrate the results using scatter plots. Let $(X_1, X_2, ..., X_n)$ be an n-dimensional random variable. If there are correlation coefficient $\rho_{ij}\,(i,j=1,2,...,n)$ with any X_i and X_j, then the n order matrix with ρ_{ij} as the element is called the correlation matrix of the random vector of this dimension, denoted as:

$$R = \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & \cdots\cdots & & \rho_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \rho_{n1} & \rho_{n2} & \cdots & \rho_{nn} \end{bmatrix} \tag{11}$$

And,

$$\rho_{ij} = \frac{E\left(\left(X_i - E\left(X_i\right)\right)\left(X_j - E\left(X_j\right)\right)\right)}{\sqrt{DX_i}\sqrt{DX_j}} \tag{12}$$

These scatter plots compares cross class aggregation data that represent the distribution of data points in a Cartesian coordinate plane in the regression analysis. In this work, the scatter plots display a sequence as a set of points, which are represented by the location of the points in the chart. The categories are represented by different labels in the chart. In this way, we can observe the scatter plot of these data points, and then find the correlations between different sensors.

3.3 Embedding for Model Inputs

After correlation analysis, the 33 sets of time series are simplified. The simplified data are used as inputs to the neural network. Each single dimensional time series of these sets is computed via the embedding layer. We can get a vector sequence as:

$$L = (L_1, L_2, ..., L_t) \tag{13}$$

And,

$$L_1 = (X_1, X_2, ..., X_i)$$
$$L_2 = (X_2, X_3, ..., X_{i+1})$$
$$\cdots$$

$$L_t = (X_t, X_{t+1}, ..., X_{t+i}) \tag{14}$$

Where *i* is the input length of the neural network.

The time series data are used as inputs to the LSTM network, which is linearly connected to the *n* hidden layer network. In Figure 4, only one hidden layer is plotted to simplify the model presentation. The output vector sequence is computed iteratively, represented as $H = (h_1, h_2, ..., h_n)$. Finally, the output vector from the output layer is calculated by the activation function, and the output vector sequence Y is obtained.

$$Y = (Y_1, Y_2, ..., Y_t) \tag{15}$$

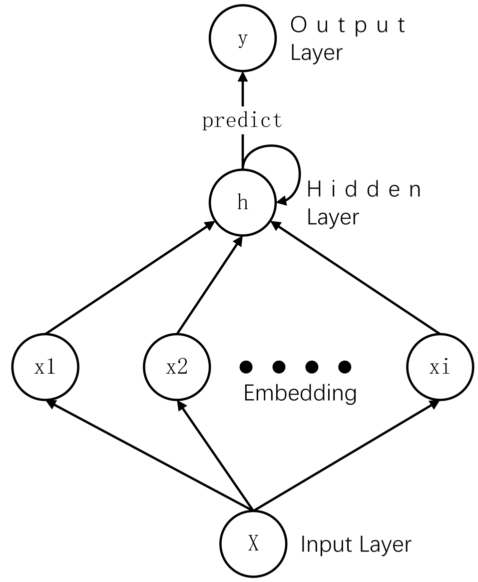


Fig. 4: LSTM Network with Embedding

From the *t=1* moment to *t=T*, the activation function of the hidden layer is computed iteratively by the formula:

$$H_t^1 = H\left(W_{i \cdot h^1} \cdot x_t + W_{h^1 h^1} \cdot h_{t-1}^1 + b_h^1\right) \tag{16}$$

The output form of each y_t is a fixed length time vector. In this paper, the prediction result vector y_t is compared with the corresponding input sequence x_{t+1}.

3.4 Searching Optimal Parameters for LSTM by OED

Orthogonal experimental design is a design method with multi-factors and multi-levels [9], which has the capabilities of partial tests replacing overall tests in order to reduce the costs of conducting all needed experiments. According to the LSTM network model, some network parameters need to be adjusted. With OED, we can determine the network parameters that are best suited for the analysis of device conditions with the least number of tests.

Orthogonal table $L_n(c^t)$ design

In this paper, *L* is the symbol for orthogonal tables, *n* is the number of experiments, *t* is the level, and *c* is the factor number. We need to adjust the number of neurons, the size of the batch, and the number of iterations. We choose $L_4(2^3)$ orthogonal table, which indicated that 4 experiments were needed, 3 factors could be observed (Number of neurons, Batch size, number of iterations), and there was no interaction between any two factors. Each factor has 2 levels. This helps to determine the test plan for OED as shown in Table 1.

The results of the experiment are evaluated by the RMSE (root mean square error). The less the RMSE, the better the prediction, and the score is y_i. Then, the magnitude of different factors is reflected by the range value *R* of the horizontal main effect *k*. By comparing the *k* value, we can also determine the influence of the horizontal change of the factors on the experiment. In the analysis of power

TABLE 1: Levels of the four factors with OED for LSTM model

Factors / Num	Number of neurons	Batch size	number of iterations
1	5	10	3000
2	5	50	5000
3	10	10	5000
4	10	50	3000

TABLE 2: RMSE scores using OED

Factor Level / Num	Number of neurons	Batch size	number of iterations	score $\left(y_i * 10^{-3}\right)$
1	5	10	3000	5.16
2	5	50	5000	4.77
3	10	10	5000	4.97
4	10	50	3000	5.00

equipment working conditions, the smaller the k value is, the better the effect of the set of parameters in the prediction is.

$$k_c = \frac{y_{t1} + y_{t2}}{2} \tag{17}$$

$$R_c = |k_{c1} - k_{c2}| \tag{18}$$

Table 2 shows the RMSE scores obtained from four sets of model parameters. Table 3 shows the k value and R value calculated according to the score in Table 2, and also the analysis of main effects (MF). It can be seen from the analysis results in Table 3, the MF of iteration number is 1, so the maximum affecting factor is the number of iterations, and it should be first controlled at the best level. From the experimental results as in Table 2, when considering 5000 times of the iteration number as the factor, the main effect is greater than that of 3000 times$(4.77 < 4.97 < 5.00 < 5.16)$, so the optimal number of iterations is 5000 times. Similarly, the second important factor is the number of neurons, the best level is 5 neurons; the third is the batch size, and the best level is 50.

3.5 Accuracy Measurement

After the above data preprocessing, we obtain the following multidimensional time series X. There is some correlations between S in each dimension, which may be strong or weak correlations. The single dimension time series is serving as input into the LSTM prediction model, 67% of them are used to train the model, 33% is used to predict, and finally we evaluate the model according to the final prediction results.

TABLE 3: OED analysis for parameter optimization of the LSTM network

Factor		Number of neurons	Batch size	number of iterations
Level	k_1	4.965	5.065	5.08
	k_2	4.985	4.885	4.87
Value of R		0.020	0.180	0.210
Superior level		5	50	5000
MF		2	3	1

$$X = \begin{cases} S_1 \\ S_2 \\ \dots \\ S_t \end{cases} \tag{19}$$

$$S = (x_1, x_2, \dots, x_n) \tag{20}$$

The most commonly used evaluation model is root-mean-square deviation (RMSE). The square term of the MSE can eliminate the positive and negative states between the deviations, and the squared deviation between the real and the predicted values is completely preserved in the index. The square root reduces the final value of the index, allowing the indicator to show more deviations. RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n} (y_i - \widehat{y_i})^2} \tag{21}$$

Where y_i is the real value of the ith sample, $\widehat{y_i}$ is the predictive value of the ith sample, and n is the number of samples.

In our evaluation, the Euclidean distance is used. However, RMSE is sensitive to the outliers. If the regression value of a regression point is very irrational, it will have a great influence on RMSE, that is, the average value is not robust. Therefore in our experiment, we get rid of the outliers first, and then we can predict and calculate the RMSE.

4 Evaluations

4.1 Correlation Results

We use a matrix graph to represent the correlation matrix and find the approximately related sensors. The data of the correlation matrix are arranged in rows and columns and arranged into matrix graphs. The corresponding correlation matrix is shown in Figure 5, which reflects the degree of correlation between sensors. It can be seen that the correlation between the sensors in the initial group is relatively large, and the correlation between the sensors behind them is relatively small in general.it

The scatter plot displays the sequence as a set of points. The value is represented by the position of the point in the chart. Class is represented by different marks in the chart. The coordinate system of two sets of data with different dimensions is used to investigate the distribution of coordinate points, to judge whether there is a correlation between the two variables or to summarize the distribution patterns of coordinate points. The scatter diagram between sensors is schematically illustrated in Figure 6.

We can see from Figure 6:

(1) The correlation of different sensors is related to the location of sensors and the type of sensors. For example, there is a strong correlation between first, second and three sensors. Because these sensors are temperature sensors sensing electrical motor, and located in the same part of the pump. This is shown in the left part of Figure 6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2825538, IEEE Access

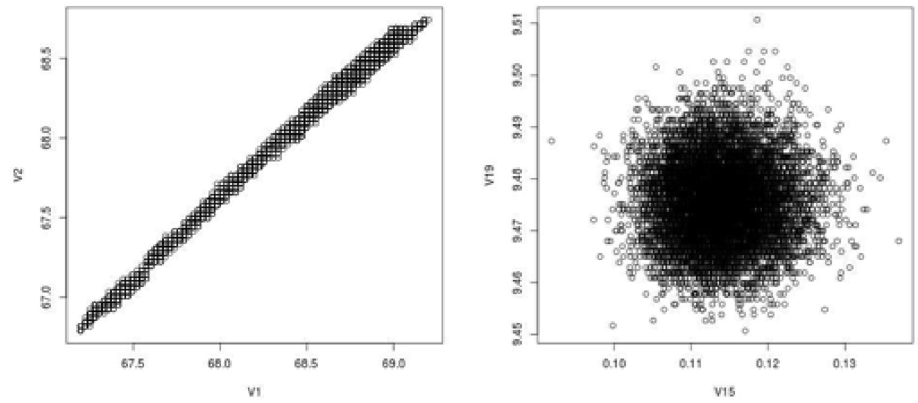


Fig. 6: Scatter diagrams between sensors

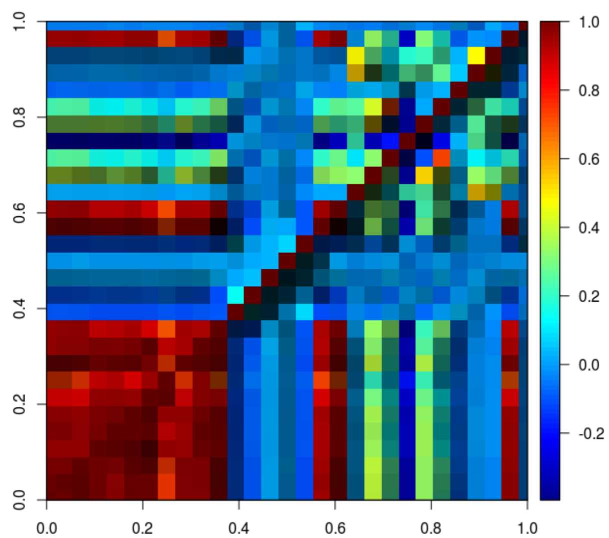


Fig. 5: The degree to which device sensor networks are correlated

(2) A very low correlation can exist between different sensors, for example, the nineteenth and the fifteenth sensors, whose scatter plots exhibit a two-dimensional Gauss noise distribution, as shown in the right part of Figure 6.

According to the above correlation analysis, we can obtain a correlation sequence based on correlation strength to facilitate experimental parameters setting of similar dimensions, in order to reduce the complexity of processing. We don't list all the scatter diagrams between the groups. The results of data classification based on correlation are shown in Table 4.

TABLE 4: The results of data classification based on correlation

Class	Sensor Number
1	1∼11,17,18,28
2	12,14,15,22,25,26,29
3	13,16
4	19∼21,23,24,27,30∼33

Scatter plots are roughly divided into 4 categories, and the number of sensors in each class is shown in the Table 4.

TABLE 5: RMSE of the first 11 sensors with the LSTM prediction model

Sensor number	1	2	3	4	5	6
RMSE	0.005	0.005	0.005	0.029	0.006	0.003
Sensor number	7	8	9	10	11	-
RMSE	0.005	0.006	0.012	0.029	0.003	-

TABLE 6: RMSE of sensors with weak correlations

Sensor number	12	13	16	17	32
RMSE	0.078	0.008	0.006	0.012	0.012

4.2 Prediction Experiment

For the first 11 sensors, by observing Figure 7, the equipment running data for three months is extracted in the data set, and the training and prediction are carried out by using the LSTM network, and the results are visualized. Obviously, the data collected by the first 11 sensors in this experiment are very similar to each other, thus proving that there is a strong correlation between these data. As shown in Table 5, the results of the LSTM network used in this experiment are evaluated by means of root mean square error. It can be seen that the result of time series prediction using LSTM network has stability and reliability, and it can be widely applied to the equipment state prediction.

As shown in Figure 8, although the data curves between the weak correlation points are different, the state prediction with the LSTM network can obtain satisfactory prediction results.

But the prediction results for the 12th sensor have bigger difference. From the actual situation analysis, the equipment vibration frequency from the 12th sensor is a typical irregular data prediction using the LSTM network, where we can see that for such kind of prediction, it is more error-prone. The mean square error is shown in table 6.

4.3 Comparison with ARIMA predictor

To show the effectiveness of the proposed approach, we compare our outcomes with that of an ARIMA predictor [10], which is often a baseline for prediction research.

Comparing Table 7 and Table 5, it can be found that the prediction error using LSTM model is significantly lower than that of the ARIMA prediction model. As shown in

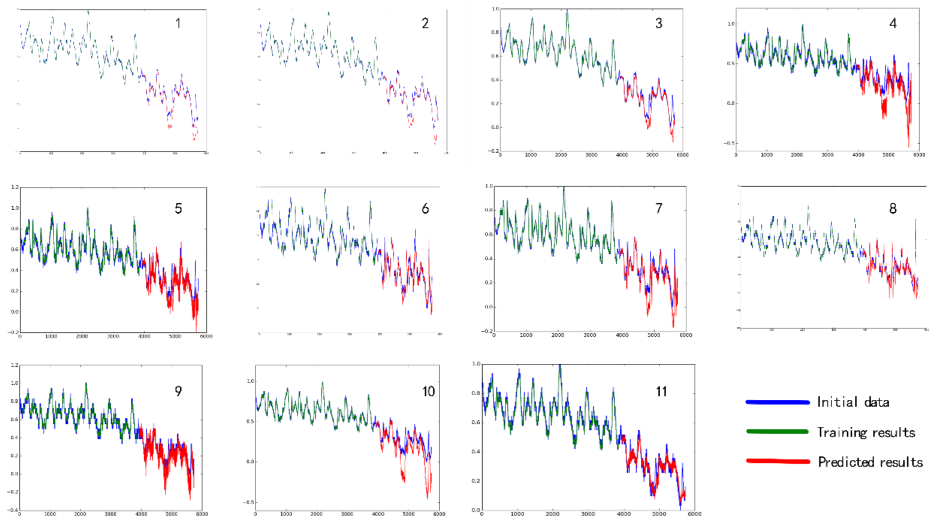


Fig. 7: Prediction Results of the First 11 Sensors

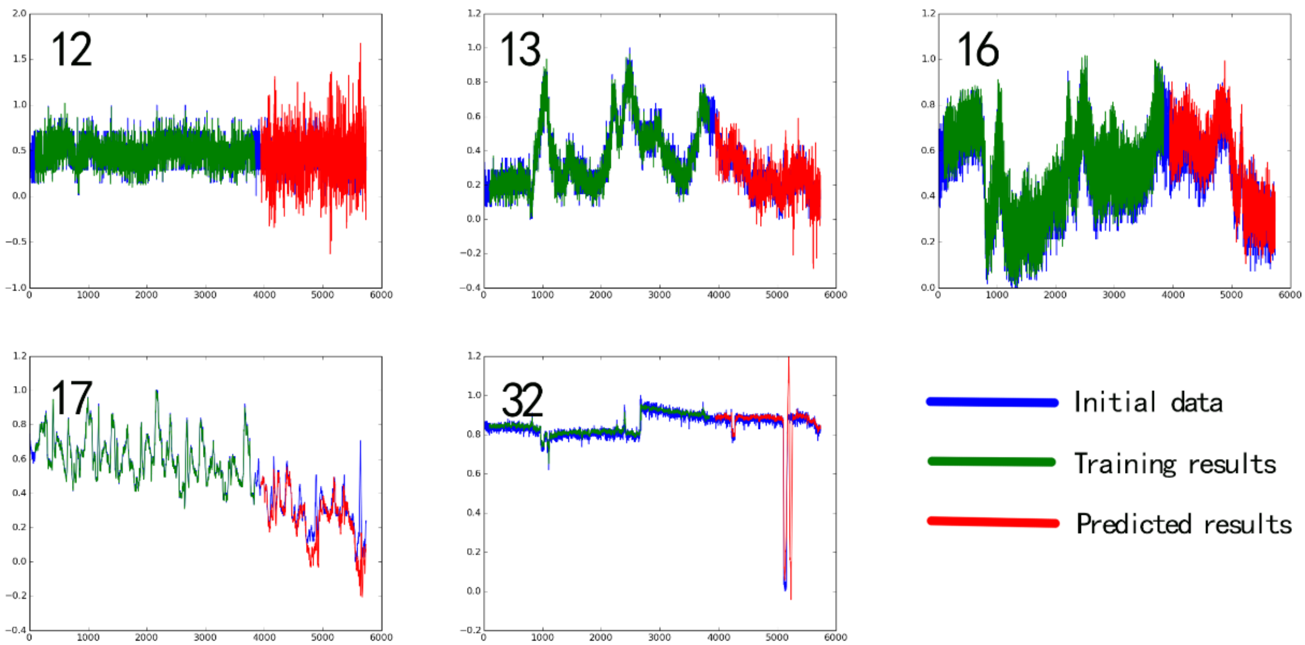


Fig. 8: Prediction results of sensors with different classifications

TABLE 7: RMSE of the first 11 sensors in the ARIMA prediction model

Sensor number	1	2	3	4	5	6
RMSE	0.024	0.025	0.025	0.066	0.099	0.062
Sensor number	7	8	9	10	11	-
RMSE	0.034	0.076	0.086	0.051	0.061	-

TABLE 8: RMSE of No.12,25,28 sensor using two models

Sensor number	12	25	28
RMSE(LSTM)	0.078	0.010	0.004
RMSE(ARIMA)	0.202	0.17	0.06

Figure 9 and Table 8, two sets of sensor data (number 28 and number 25), with weak correlation are selected randomly, their predictions are only 0.004 and 0.010 for RMSE value under the LSTM model, but their RMSE value in the ARIMA model are 0.06 and 0.17 respectively. The RMSE for No.12 sensor is 0.202 under the ARIMA model (greater than 0.078 under the LSTM model).

4.4 Discussion

From the single dimensional time series analysis, we can get the conclusion that the prediction results of strongly correlated sensors are similar. However, it is not always possible to make a high accuracy decision on the overall operation of the equipment. If multi-dimensional fusion analysis can be carried out, the equipment working conditions can better be predicted at each moment. The pre-processed data are

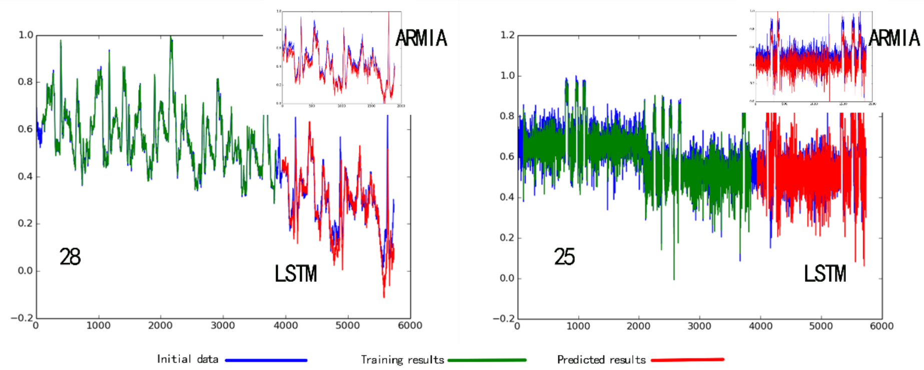


Fig. 9: Comparison of Prediction Results between Two Network Models of LSTM and ARIMA

adjusted according to the requirements of different dimensions, and then used as input to the LSTM network. The neural network will then automatically perform feature extraction, training, learning and prediction, including model fitting and activation function settings. This leads to an improvement of prediction accuracy of working conditions, which include all measuring points and overall prediction results.

5 Related Work

There is some similar work on power equipment condition prediction. Zhang et al. proposed a trend prediction method–full vector LS-SVM, which combines the full vector spectrum technology based on information fusion homologous [11].Compared with the traditional single channel signal extraction method, this method guarantees the integrity of LS-SVM prediction data in feature extraction, thus improving the prediction accuracy.

Zhang et al. studied a prediction method based on discrete process neural networks [12]. In order to avoid choosing a local optimal solution during the training, they introduced the chaotic particle swarm optimization algorithm in the training. Liu et al. trained a model to fit the monitored data [13]. They proposed a framework to predict future peaks. More specifically, based on the peak information in historical data, a data-driven model is trained to predict the next peak. A real case regarding the first seal leak of a reactor coolant pump was considered to validate the effectiveness of the proposed framework. Aydin et al. performed LSTM to predict the current situation of an engine [14]. All those work does not focus on data correlation prediction. These existing work did not use LSTM for predicting equipment running status.

Researchers are also using LSTM for predictions in other areas. For example, Tax et al. investigates Long Short-Term Memory (LSTM) neural networks as an approach to build consistently accurate models for a wide range of predictive process monitoring tasks [15]. Following the recent success of Recurrent Neural Network (RNN) models for sequence prediction tasks. Alahi et al. propose an LSTM model which can learn general human movement and predict their future trajectories [16] . Zhao et al. proposed a new conventional foreast models which used the LSTM network and considers temporal-spatial correlation in traffic system via a two-dimensional network which is composed of many memory units [17]. Fu et al. used GRU to traffic flow prediction,which performed better than auto regressive integrated moving average (ARIMA) model [18].

Some are using LSTM for anomaly detection. Ordez et al. pointed out that the convolutional LSTM architecture achieved significantly better results than traditional ways, even though the classifier was trained on raw input sequences without any further feature engineering [19]. This shows the potential of LSTM for the task of time series analysis. Malhotra et al.learns to reconstruct 'normal' time-series behavior, and thereafter uses reconstruction error to detect anomalies [20]. Wielgosz et al. presents a model based on LSTM and GRU for facilitating an anomaly detection in Large Hadron Collider superconducting magnets [21].

6 Conclusions and Future work

In this paper, we present a systematic LSTM-based method for time series analysis and forecasting of sensor data from IIoT equipment. The LSTM model captures the time span surrounding the faults and defects. Our method is equipped with an analysis workflow that provides a complete solution for data cleaning and feature engineering. The experiment shows that the RMSE for most of the actual power pump state is under 0.010. The prediction results are stable. The techniques we present in this paper can be applied to analyze other time series from IIoT, especially those with sensor correlations. In the future, we will add more dimensions to enhance working condition forecasting in a richer environment. We will also study root causes of faults, and analyze its characteristics by other time series classification techniques, such as the one using deep belief network (DBN) [22].

Acknowledgments

This research is supported by the Key Research Program of Shandong Province (No. 2017GGX10140) and also supported by Fundamental Research Funds for the Central Universities.

REFERENCES

[1] T. Qiu, R. Qiao, and D. O. Wu, "Eabs: An event-aware backpressure scheduling scheme for emergency internet of things," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2018.

[2] C. U. Matsuoka, Yoky (Palo Alto, "Methods and systems for identification and correction of controlled system data," no. 20150161515, June 2015. [Online]. Available: http://www.freepatentsonline.com/y2015/0161515.html

[3] J. Lin, K. S. Aggour, B. S. Courtney, J. A. Interrante, C. A. Lacomb, S. Mathur, C. T. Mcconnell, and Q. Snell, "Method for storage, querying, and analysis of time series data," 2015.

[4] P. M. T. Broersen, "Autoregressive model orders for durbin's ma and arma estimators," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2454–2457, 2017.

[5] W. Maleika, "Moving average optimization in digital terrain model generation based on test multibeam echosounder data," *Geo-Marine Letters*, vol. 35, no. 1, pp. 61–68, 2015.

[6] Y. Min and C. Ao, "A new data mining method improvement of time series based on auto regressive moving average model," *Metallurgical Mining Industry*, 2015.

[7] Y. Duan, Y. Lv, and F. Y. Wang, "Travel time prediction with lstm neural network," in *IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 1053–1058.

[8] H. Adel and H. Schtze, "Global normalization of convolutional neural networks for joint entity and relation classification," *CoRR*, vol. abs/1707.07719, 2017. [Online]. Available: http://arxiv.org/abs/1707.07719

[9] W. Zhang and D. Wang, "Optimization of sheet metal forming process parameters by artificial neural network and orthogonal test method," in *International Conference on Electrical, Mechanical and Industrial Engineering*, 2016.

[10] J. D. Hamilton, *Time series analysis*. Princeton university press Princeton, 1994, vol. 2.

[11] Q. Zhang, J. Han, L. Chen, W. U. Yanzhao, and H. U. Xin, "Equipment condition trend prediction based on full vector least squares support vector machine," *Machine Tool and Hydraulics*, 2016.

[12] Y. H. Zhang, X. H. Han, and S. H. Wang, "Equipment's condition prediction based on the discrete process neural networks," *Journal of University of Electronic Science and Technology of China*, 2016.

[13] J. Liu and E. Zio, "Prediction of peak values in time series data for prognostics of critical components in nuclear power plants," in *Ifac Amest*, 2016.

[14] O. Aydin and S. Guldamlasioglu, "Using lstm networks to predict engine condition on large scale data processing framework," in *International Conference on Electrical and Electronic Engineering*, 2017, pp. 281–285.

[15] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, *Predictive Business Process Monitoring with LSTM Neural Networks*, 2017.

[16] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F. F. Li, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[17] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IEt Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[18] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *Youth Academic Annual Conference of Chinese Association of Automation*, 2017, pp. 324–328.

[19] F. J. Ordez and Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, no. 16 (1), pp. 324–328, 2016.

[20] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv: Artificial Intelligence*, 2016.

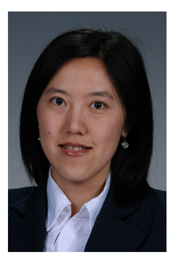
[21] M. Wielgosz, A. Skocze, and M. Mertik, "Recurrent neural networks for anomaly detection in the post-mortem time series of lhc superconducting magnets," 2017.

[22] W. Zhang, P. Duan, L. T. Yang, F. Xia, Z. Li, Q. Lu, W. Gong, and S. Yang, "Resource requests prediction in the cloud computing environment with a deep belief network," *Softw., Pract. Exper.*, vol. 47, no. 3, pp. 473–488, 2017. [Online]. Available: https://doi.org/10.1002/spe.2426

Weishan Zhang Weishan Zhang is a full professor with Department of Software Engineering, China University of Petroleum. He got PhD from Northwestern Polytechnical University, China in 2001. His current research interests are big data platforms, pervasive cloud computing, and service oriented computing. Weishan has published over 100 papers and his current h-index according to Google scholar is 15.

Wuwu Guo Wuwu Guo is a master student in the China University of Petroleum's Department of Software Engineering. His research interests are time series data analysis, big data processing.

Xin Liu Xin Liu is an associate professor at College of Computer and Communication Engineering, China University of Petroleum, Qingdao, China. She received her PhD from Nankai University, China in 2012. Her research interests include social networks, data mining, big data processing.



Yan Liu Yan Liu is an associate professor in the Electrical and Computer Engineering Department at Concordia University, Montreal. Her research interests are data stream processing, cloud software architecture, distributed computing, software performance engineering, and adaptive systems. Liu received a PhD in computer science from the University of Sydney. She's a member of IEEE. Contact her at yan.liu@concordia.ca.

Bo Li Bo Li is a master student in the China University of Petroleum's Department of Software Engineering. His research interests are big data processing, software architecture. Li received a bachelor of software engineering from the China University of Petroleum.

Jiehan Zhou Jiehan Zhou is currently a senior researcher in University of Oulu. He got his PhD in computer engineering from University of Oulu, Finland and PhD in Manufacturing Automation from Huazhong University of Science and Technology, China. His current research interests are big data platforms, Internet of Things, pervasive cloud computing, He has published over 100 papers.

Qinghua Lu Qinghua Lu is an associate professor at Department of Software Engineering, China University of Petroleum, Qingdao, China. She received her PhD from University of New South Wales (UNSW) in 2013. Her research interests include software architecture, dependability of cloud computing, and service engineering.

Su Yang Su Yang is a professor with Dept. of Computer Science and Engineering, Fudan University. His research interests are mainly pattern recognition, social computing, machine vision, and data mining. He is the PI of a number of NSFC projects including Graphical Symbol recognition in natural scenes and Detection of abnormal collective behaviors via movement and communication pattern analysis.