Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-17 (2018) 831–836

Water Quality Prediction Model Combining Sparse Auto-encoder and LSTM Network

Zhenbo Li*, **, *.Fang Peng*. Bingshan Niu*. Guangyao Li*. Jing Wu*. Zheng Miao*.**

** College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China (Tel:18701586839; e-mail: zhenboli@126.com).*

*** Key Laboratory of Agricultural Information Acquisition Technology, Ministry of Agriculture, Beijing 100083, China (e-mail: zhenboli@126.com)*

**** Beijing Engineering and Technology Research Center for Internet of Things in Agriculture, Beijing 100083, China (e-mail: zhenboli@126.com)}*

Abstract: In order to improve the prediction accuracy of dissolved oxygen in aquaculture, a hybrid model based on sparse auto-encoder (SAE) and long-short-term memory network (LSTM) is proposed in this paper. The hidden layer data pre-trained by SAE contains deep latent features of water quality, and then input it into the LSTM to enhance the prediction accuracy. Experimental results show that SAE-LSTM surpasses LSTM through reducing MSE respectively by 23.3%, 53.6%, and 39.2% in the prediction steps of 3, 6, and 12 hours, and surpasses SAE-BPNN by 87.7%, 91.9%, and 90.0%, proving that our hybrid model is more accurate.

Keywords: water quality prediction; dissolved oxygen; sparse auto-encoder; long-short-term memory network

1. INTRODUCTION

The scale of aquaculture in China is growing and intensive farming is gradually becoming the main production mode of aquaculture. The increase of breeding density, and the unscientific utilizing of medicines or feeding can lead to frequent breeding diseases (**Bai, 2014**). On the other hand, aquaculture water have poor self-purification capability, so that long-term accumulation of nitrogen, phosphate, and organic residues in the breeding process cannot be decomposed, which can easily lead to eutrophication and the ecological environment pollution. Water quality is the key of aquaculture. At present, the water quality is judged mostly by experience which is inaccurate and laborious. If data can be collected through the IoT devices, and then predict the change trend of water quality factors by deep learning algorithms, it will help early prevention and control of disasters, which will reduce the economic loss, and protect the ecological environment at the same time.

The water quality factors mainly include temperature, dissolved oxygen, pH value, ammonia nitrogen, nitrites, nitrates, etc., and they have complex interactions with each other, making the water environment complex, changeable, and non-linear (**Chan, 2014**). In recent years, the prediction of water quality time series by machine learning modelling has made some progress (**Chen, 2012**). Such as (**Yan, 2014; Khotimah, 2015**) respectively employed BP neural network and smooth support vector machine (SSVM). (**Liu, 2013**) used real-valued genetic algorithms to search for optimal SVR parameters. (**Liu, 2014; Liu, 2013**) proposed a mixed dissolved oxygen prediction model based on Wavelet Analysis (WA) and Least Squares Support Vector Regression (LSSVR). (**Deng, 2015; Li, 2017**) proposed a hybrid model for water quality prediction.

Most of the above schemes adopted the combined model to improve the traditional BP neural network and support vector machine method, which proves the effectiveness of the combined model to predict water quality. However, they have the bottleneck in mining the hidden information of water quality sequence, and it is difficult to learn the long-term dependence between water quality time series. Currently it is becoming a research hotspot to apply the deep learning algorithms that can automatically extract features. (**Ta, 2018**) proposed a simplified inverse CNN prediction model to solve the problem of dissolved oxygen prediction. (**Luo, 2018**) combined sparse auto-encoder (SAE) and extreme learning machine (ELM), obtaining more sparse and compact feature information, so that the predicting accuracy can be further improve. Therefore, this paper proposes a water quality prediction method based on a fusion model of sparse auto-encoder (SAE) and long-short-term memory networks (LSTM), named SAE-LSTM, to further improve the accuracy and stability of dissolved oxygen prediction.

For decades, auto-encoders have been widely used for dimension reduction or feature learning (**Lu, 2016**). It is an unsupervised pre-training network composed of an input layer, a coding layer (hidden layer), and a decoding layer. It can learn a characteristic transformation $h = f(Wx + b)$ by making up a three-layer network of x to h to x (**Erhan, 2010**). (**Wang, 2014**) verified the dimension reduction effect of the auto-encoder and the stability of its expression capability. In particular, it is possible to apply sparsity restriction to the hidden units to produce a variant of the auto-encoder, the sparse auto-encoder (**Bengio, 2012**). Sparse auto-encoder (SAE) can automatically learn features from unlabeled data,

10.1016/j.ifacol.2018.08.091

giving better feature description than the original data. In the actual application, if the features found by SAE can replace the original data, they can lead to better results.

Long-short-term memory network (LSTM) is a special kind of recurrent neural network that can memorize the previous information and applies it to the calculation of the current output (**Graves, 2012**). It solves the problem of gradient disappearance in traditional recurrent neural networks by selectively memorizing or forgetting some data (**Lecun, 2015**), which has long-term memory capability and is suitable for processing water quality time series data.

2. MATERIALS AND METHODOLOGIES

2.1 Sparse Auto-Encoder (SAE)

The specific steps of SAE are as follows:

Step.1 Determine the number of hidden layer neuron nodes M, and randomly generate the connection weights between the input layer and the hidden layer, $w_i=[w_{1i}, w_{2i}, \ldots, w_{mi}]^T$, $(i=1, 2, \ldots, M)$, and the threshold value of the hidden layer neurons, $b=[b_1, b_2, \ldots, b_M]^T$. Then we use the original input x_k to train the SAE, which can learn the feature representation h_k of the original input, as shown in **Fig.1**.

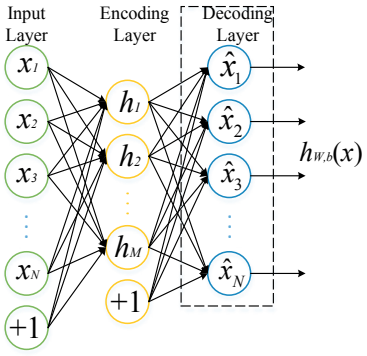

Fig.1. The network structure of sparse auto-encoder

Step.2 Define $a_j^{(2)}(x)$ as the activation value of element j in the hidden layer when given a specific input x. The output of sigmoid activation function values in [0, 1], 0 represents suppression, 1 represents activation. The sparsity parameter is defined as the average activation value of the hidden unit j on the training set. The constraint $\hat{\rho}_j=\rho$ is as shown in (1).

$$\hat{\rho}_j=\frac{1}{N}\sum_{i=1}^{N}[a_j^{(2)}(x^{(i)})]=\rho \qquad (1)$$

In the formula (1), ρ is a sparsity parameter, which is generally a number close to 0, such as 0.05; N is the number of neurons in the input layer.

Step.3 The KL divergence is added as a penalty term to the loss function, which guides the entire network to learn sparse feature expressions. The characteristic of KL divergence is that the closer $\hat{\rho}_j$ get to ρ, the smaller KL is, which can be used to punish those $\hat{\rho}_j$ that deviate too much and make $\hat{\rho}_j$ closer to ρ in the training process. This is shown in (2).

$$\sum_{j=1}^{M}KL(\rho\,\|\,\hat{\rho}_j)=\sum_{j=1}^{M}\rho log\frac{\rho}{\hat{\rho}_j}+(1-\rho)log\frac{1-\rho}{1-\hat{\rho}_j} \qquad (2)$$

So the loss function of sparse auto-encoder is:

$$J_{sparse}(W,b)=J(W,b)+\beta\sum_{j=1}^{M}KL(\rho\,\|\,\hat{\rho}_j) \qquad (3)$$

In (3), β controls the weight of the sparse penalty term, and $J(W, b)$ is a 1/2 square error cost function, which contains the weight decay term; W and b represent the connection weights and bias from the input layer to the hidden layer respectively. M is the number of neurons in the hidden layer. The training objective is to minimize the target loss function $J_{sparse}(W,b)$.

The partial derivatives of the loss function after adding KL divergence are calculated as (4), where δ is an error term.

$$\delta_i^{(2)}=\left(\sum_{j=1}^{M}W_{ji}^{(3)}\delta_j^{(3)}\right)f'\left(z_i^{(2)}\right)+\beta\left(-\frac{\rho}{\hat{\rho}_j}+\frac{1-\rho}{1-\hat{\rho}_j}\right) \qquad (4)$$

Step.4 After having new partial derivatives, adopt back propagation to optimize parameters of the entire network.

$$W_{ij}^{(2)}=W_{ij}^{(2)}-\alpha\frac{\partial}{\partial W_{ij}^{(2)}}J_{Sparse}(W,\,b)=W_{ij}^{(2)}-\alpha a_j^{(2)}\delta_i^{(3)} \qquad (5)$$

$$b_i^{(2)}=b_i^{(2)}-\alpha\frac{\partial}{\partial b_i^{(2)}}J_{Sparse}(W,\,b)=b_i^{(2)}-\alpha\delta_i^{(3)} \qquad (6)$$

In (5) and (6), α is the learning rate, W_{ij} represents the connection weight between the ith neuron of the input layer and the jth neuron of the hidden layer, and b_i represents the bias. After normalizing the initial data and iterative pre-training by the sparse auto-encoder, the next-step prediction model can be established.

2.2 Long-short-term Memory Network(LSTM)

As shown in **Fig.2**, the LSTM network has the form of a repeating block chain for learning the time series information. The key of it is the cell state, which is the horizontal line above the graph (from C_{t-1} to C_t). And the cell state is somewhat like a conveyor belt, running directly through the entire chain, with only minor linear interactions, which makes information can easily be kept in a constant state and keep flowing above it (**Wang, 2017**).

LSTM network has three important "gate", respectively named input gate, output gate, and forget gate. One gate is usually composed of a sigmoid layer and a pointwise multiplication, allowing information to be selectively passed (**Graves, 2012**). They are functioned as a multi-level feature selector to either increase (memorize) or remove (forget) information to the cell state. When the peephole connections are selectively added to some gates, such as the connection line between the three sigmoid layers and the cell state in **Fig.2**, they can enhance the performance of LSTM. That means the "gate" also sees the cell state and read the value of C_t. (**Lipton, 2015**) proved that the LSTM network enhanced by the peephole connections could separate the subtle differences between sequences from 50 or 49 discrete-time

steps without the help of any short-term training samples. So we adopt peepholes in our SAE-LSTM model to help learn highly nonlinear sequences stably.

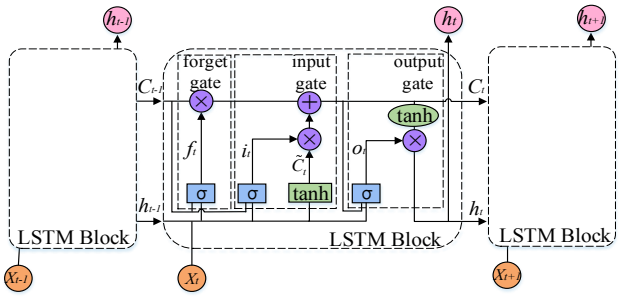

Fig.2. The block of LSTM

The information in the LSTM flows as follows:

Step.1 Determine what information to discard from the cell state by the forget gate, which reads h_{t-1}, x_t and C_{t-1}, outputs a number between 0 and 1 for each cell in C_{t-1}. 1 means "completely reserved", 0 means "completely discarded." At each time t, x_t is the input of the current moment, h_{t-1} is the output of the hidden layer at the last moment, and C_{t-1}, the cell state at the last moment is also read because of the added peephole connections. In (7), f_t denotes the output of the forget gate at the current time, σ is the abbreviation of the sigmoid function, W_f and b_f respectively denote the weight and bias from the input layer to the forget gate.

$$f_t = \sigma(\mathrm{W}_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f) \qquad (7)$$

Step.2 Decide what new information to store in the cell state. This consists of two parts: firstly, the sigmoid layer of the input gate filters the information from h_{t-1}, x_t and C_{t-1}, secondly, the tanh layer creates a vector containing all the candidate values added to the cell state with a value of -1 to 1. In (8), i_t denotes the output of the input gate, W_i and b_i respectively denote the weight and bias of the input gate. And W_C and b_C in (9) respectively represent the weight and bias of the cell state.

$$i_t = \sigma(\mathrm{W}_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i) \qquad (8)$$

$$\tilde{C}_t = \tanh(\mathrm{W}_C \cdot [h_{t-1}, x_t] + b_C) \qquad (9)$$

Step.3 Update the old cell state, namely C_{t-1} is updated to C_t. Multiply the old state by f_t and discard the information that you decided to forget. Then add $i_t * \tilde{C}_t$, the new information that you decide to remember. After completing the above steps, we basically ensure that the information added to the cell state is important and not redundant as (10) shows, where C_t represents the cell state at the current moment.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \qquad (10)$$

Step.4 Decide what value to output that is the filtered information of the cell state. Firstly, run a sigmoid layer that determines which parts of the cell state need to be exported. Then make the value of the cell state between -1 and 1 by the tanh layer and multiply it by the sigmoid layer output. In (11) and (12), o_t denotes the output of the output gate, W_o and b_o respectively represent the weight and bias from the input gate to the output gate, and h_t denotes the output of the hidden layer at the current moment.

$$o_t = \sigma(\mathrm{W}_o \cdot [C_t, h_{t-1}, x_t] + b_o) \qquad (11)$$

$$h_t = o_t * \tanh(C_t) \qquad (12)$$

2.3 The Combined Model SAE-LSTM

The dissolved oxygen prediction framework based on SAE-LSTM is shown in **Fig.3**. And the detailed calculation process is described as follow steps.

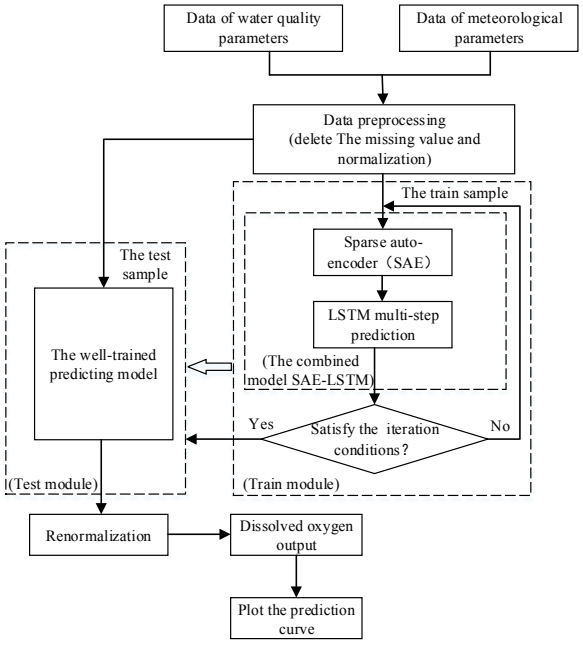

Fig.3. Flow chart of our water quality prediction model

Step.1 Data pre-processing. Including the interpolation of the missing values of dataset, and the normalization in order to eliminate the dimension influence between water quality factors. Data normalization can speed up the speed of gradient descending to find the optimal solution and possibly improve the prediction accuracy.

Step.2 Input the normalized data from the Step.1 into a sparsely auto-encoder network for pre-training, where the network parameters are randomly initialized, and the iteration of errors is performed through backward propagation.

Step.3 Input the multi-dimensional and sparse activation value of the hidden layer learned in the Step.2 into the LSTM network with parameters randomly initialized. And then set the prediction step length, train errors by batching stochastic gradient descent and backpropagation. After that, compare the predicted value with the real value to calculate whether the iteration condition is satisfied. That is the error should descend to a certain degree or the number of iterations be reached.

Step.4 Use the train sample to rain the model through step 2 and step 3, then input the test sample into the well-trained

model to output the renormalized predicted dissolved oxygen value of the test sample. Finally, and plot the prediction curve.

With the SAE-LSTM combination method, the ability of LSTM to memorize long-term sequence data can be utilized to learn the temporal characteristics and cumulative effects of data. At the same time, sparse auto-encoders are adopted to mine the hidden features in the data, and the multi-dimensional and sparse features are better to improve the shortcomings of LSTM which is easy to forget recent data, so as to further improve the accuracy of LSTM model prediction.

3. EXPERIMENT AND ANALYSIS

3.1 Experimental Data

The experimental dataset was measured in a shrimp pond by every 10 minutes one set for 20 consecutive days, including four water quality factors and four meteorological factors: dissolved oxygen (DO), water temperature (W_T), ammonia nitrogen content (Am), pH and atmospheric temperature (A_T), air humidity (Hu), atmospheric pressure (AP), and wind speed (WS), a total of 2880 sets of data. The program execution environment is keras+theano+cudNN, based on the Windows 10 pro 64-bit operating system. The original time series for each water quality factor is shown in **Fig.4**. We select dissolved oxygen (DO) as the predicted factor.

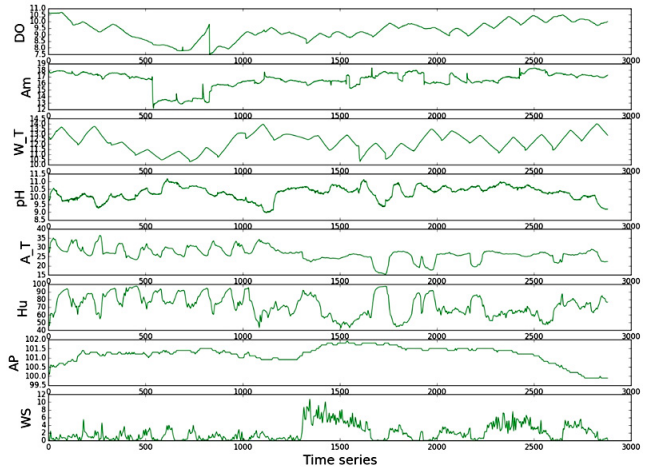

Fig.4. Time series of water quality and meteorological factors

3.2 Experiment and results

The experimental part of this paper compares the results of the DO prediction values of the SAE-LSTM model with that of the SAE-BPNN model, the single LSTM model, the single BPNN model and the actual values. Where the SAE parameters used in SAE-BPNN are consistent with those in SAE-LSTM, and SAE-BPNN is a combined method resemble this paper, but use BPNN to replace LSTM.

We set 100 hidden neurons in the SAE algorithm, set 50 hidden neurons in the LSTM algorithm, and select 0.1 as the learning rate. In the LSTM network, we train the network with batch stochastic gradient descent (batch_size=72) as an optimizer in order to equilibrate the convergence speed and accuracy. The last layer of the LSTM is a dense layer and the loss function is the mean square error function. We also adopt early stopping to prevent overfitting. As for the BPNN algorithm, we randomize the initial weights and optimize the number of hidden neurons. The ratio of the training sample to the test sample is set to 3:1. And we set a random seed to replace repetitive experiments.

We respectively select the prediction step lengths as 18, 36, and 72 to predict the dissolved oxygen content in the following 3 hours, 6 hours, and 72 hours. The input of model is the eight factors above, and the dissolved oxygen value after the prediction step length represent the output. When the prediction step length is 18, that is, the prediction duration is 3 hours, the result of our combined prediction model is shown in **Fig.5** where the blue curve represents the ground truth. The first sub-figure represents the training error decreasing with the epochs, the second sub-figure describes the fitting degree of the model output value and the ground truth, and the third sub-figure describes the fitting degree of the predicting value and the ground truth.

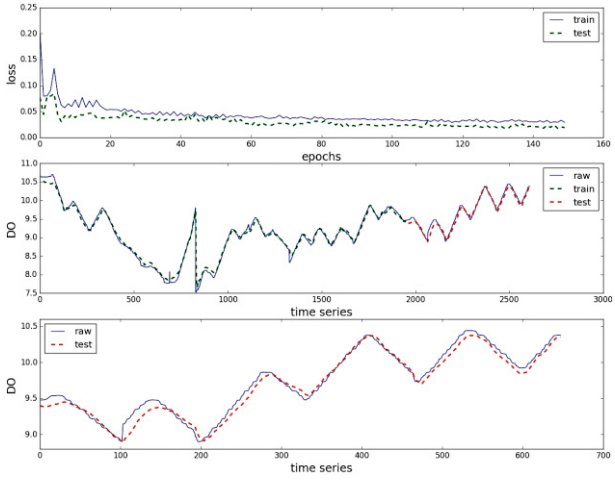

Fig.5. The prediction of SAE-LSTM in 3 hours

When the prediction duration is 6 hours, our combined model performs as shown in **Fig.6**.

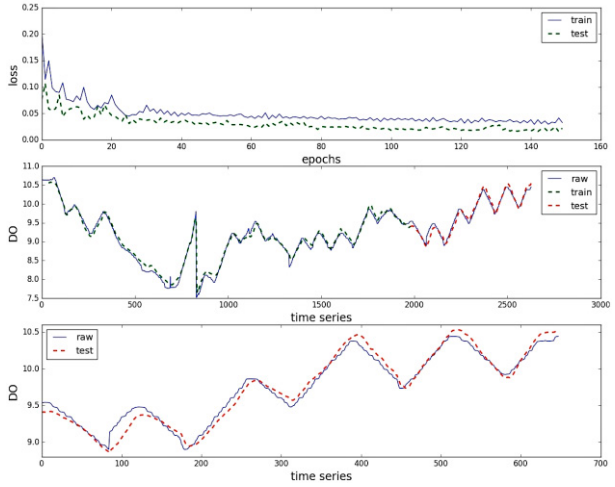

Fig.6. The prediction of SAE-LSTM in 6 hours

When the prediction step length is 72, namely the prediction duration is 12 hours, our combined model performs as shown in **Fig.7**.

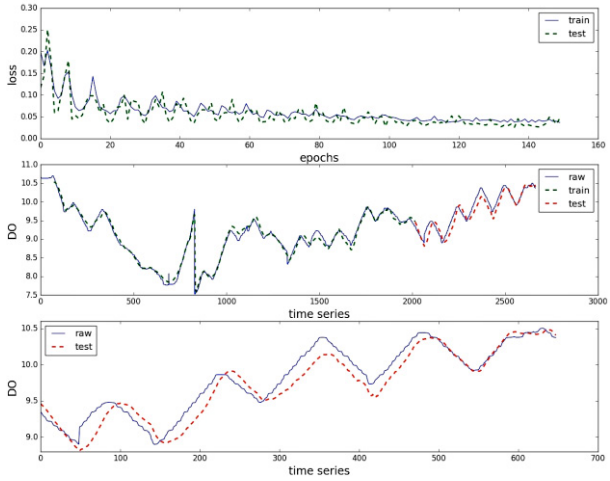

Fig.7. The prediction of SAE-LSTM in 12 hours

Comparing **Fig.5**, **Fig.6** and **Fig.7**, it can be seen that the prediction accuracy of SAE-LSTM decreases with the prediction step length increasing. This is consistent with the common sense that the longer the predicting duration, the more difficult it is to predict accurately. The loss error converges with the number of iterations at the early time, which proves that the convergence speed of the model is fast. At the same time, as the prediction step length increases, the convergence speed decreases, proving the calculation cost and model complexity both increase with prediction step length. Remarkably, the model prediction results are very similar when the prediction step lengths are 18 and 36, indicating the stability of our model.

Compare the four models we mentioned above as shown in **Fig.8**.

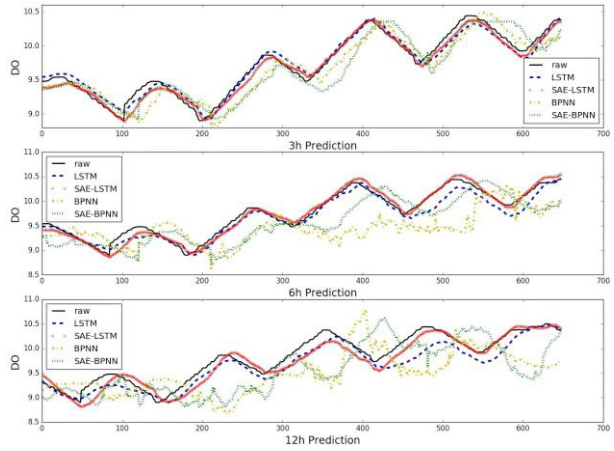

Fig.8. Comparison of the prediction results of four models

From **Fig.8,** it can be seen that SAE-LSTM and LSTM can better fit the sequence with the increase of prediction step, but SAE-BPNN and BPNN have shown large error fluctuation, especially the prediction of dissolved oxygen after 12 hours. It has proven the ability of LSTM to predict for long periods of time. In comparison, SAE-LSTM is better than LSTM, and SAE-BPNN is better than BPNN, indicating that SAE used as a feature extraction pre-training network has enhanced the performance of LSTM and BPNN. Section 3.3 will use quantitative indicators to analyse the forecast results.

4.1 Quantitative Evaluation and Analysis

Finally, the prediction effect of the model is evaluated by adopting Mean Absolute Error (MAE), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). In order to reduce the randomness error of the algorithm, a random seed is set during the experiment to ensure the consistency of each operation result. Then we show the four evaluation indicators and running times of models in **Table 1** and **Fig.9**.

Table 1. Comparison of the four models

	Model Name	MAE	MSE	MAPE	RMSE	Run Time (s)
3h-Prediction	SAE-LSTM	0.059	**0.0056**	0.006	0.075	25.8
	SAE-BPNN	0.180	0.0457	0.018	0.214	8.3
	LSTM	0.075	0.0073	0.008	0.085	19.9
	BPNN	0.165	0.0359	0.017	0.189	2.3
6h-Prediction	SAE-LSTM	0.069	**0.0077**	0.007	0.088	28.2
	SAE-BPNN	0.258	0.0953	0.027	0.309	9.1
	LSTM	0.100	0.0166	0.010	0.129	22.0
	BPNN	0.3	0.1352	0.031	0.368	3.4
12h-Prediction	SAE-LSTM	0.126	**0.0242**	0.013	0.155	29.6
	SAE-BPNN	0.406	0.2428	0.041	0.493	9.1
	LSTM	0.159	0.0398	0.016	0.199	23.2
	BPNN	0.453	0.3013	0.045	0.549	3.6

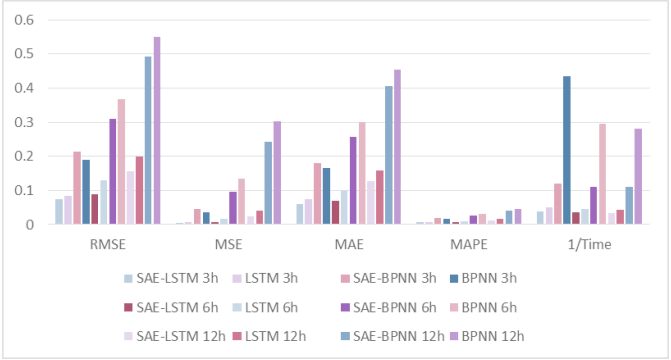

Fig.9. Error and run time comparison histograms for the four models

MSE can evaluate the variation degree of data. The smaller the value of MSE is, the better the prediction accuracy of the model to describe the experimental data. And MAPE is the mean absolute percentage error. From **Table 1**, it can be seen that, the prediction accuracy of SAE-LSTM decreases as the prediction step increases, and the RMSE change from 0.075 to 0.088 and to 0.155, the other models are the same.

According to **Table 1** and **Fig.9**, the combined model SAE-LSTM is slightly slower than the single LSTM model, but the error is smaller than it, and the MSE is reduced respectively by 23.3%, 53.6%, and39.2% in the prediction steps of 3, 6, and 12 hours. The combined model SAE-BPNN respectively decreases MSE in the prediction steps of 3, 6, and 12 hours, by -27.3%, 29.5% and 19.4% compared with BPNN. All above proves that SAE as feature extraction pre-training network can enhance the performance of LSTM and BPNN, but SAE's support to BPNN is not very good in a few cases.

In pairwise comparison with SAE-BPNN, the SAE-LSTM reduces MSE than by 87.7%, 91.9%, and 90.0% in the prediction steps of 3, 6, and 12 hours respectively, and LSTM surpasses BPNN through reducing the MSE by 79.7%, 87.7%, and 86.8% respectively in the prediction steps of 3, 6, and 12 hours, indicating that LSTM is more accurate than the standard BPNN model in predicting water quality factors.

4. CONCLUSIONS AND DISCUSSIONS

Deep learning methods such as AE and LSTM have natural advantages in extracting features and processing sequence data, which provide new ideas for research on water quality prediction. This paper verified that SAE as a feature extraction pre-training network can enhance the prediction accuracy of LSTM. However, there are still some theoretical and technical issues in the field of aquaculture water quality prediction:

1) The future work can explore a multivariate output that only trains the model once, so as to realize more comprehensive prediction and analysis of the water body.

2) To explore the effect of increasing the network layers of SAE on the prediction accuracy, it is possible to establish a deep network through layer-wise pre-training to obtain deeper latent features.

ACKNOWLEDGEMENT

This research was supported by International Science & Technology Cooperation Program of China (No.2015DFA00530), Key Research and Development Research Project of Shandong Province (No.2016CYJS03A02) and National Nature Science Foundation of China Project (No.61471133，61472172).

REFERENCES

Bai, Z. (2014). Sparse extreme learning machine for classification. *IEEE Transactions on Cybernetics*, 44(10), 1858–1870.

Bengio, Y. (2012). Deep Learning of Representations for Unsupervised and Transfer Learning, *PMLR*, 27, 17-36.

Chan, J., H. Ma, T. Saha., and C. Ekanayake. (2014). Self-adaptive partial discharge signal de-noising based on ensemble empirical mode decomposition and automatic morphological thresholding. *IEEE Transactions on Dielectrics and Electrical Insulation*, 21(1), 294–303.

Chen, C. F., M. C. Lai, and C. C. Yeh. (2012). Forecasting tourism demand based on empirical mode decomposition and neural network. *Knowledge-Based Systems*, 26, 281–287.

Deng, W., et al. (2015). A novel hybrid water quality time series prediction method based on cloud model and fuzzy forecasting. *Chemometrics and Intelligent Laboratory Systems*. 149, 39-49.

Erhan, D., et al. (2010). Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11, 625-660.

Graves. A. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. *Studies in Computational Intelligence*, 1735-1780. Springer, Berlin Heidelberg.

Khotimah, W. N. (2015). Aquaculture water quality prediction using smooth SVM. *Iptek J Proc*, 1(1), 342–5.

LeCun, Y., Bengio Y., and Hinton. G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Li, C., et al. (2018). A hybrid model for dissolved oxygen prediction in aquaculture based on multi-scale features. *Information Processing in Agriculture*, 5(1), 11-20.

Lipton, Z., Berkowitz, J., and Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning.

Liu, S., et al. (2013). A hybrid approach of support vector regression with genetic algorithm optimization for aquaculture water quality prediction. *Mathematical and Computer Modelling*, 58(3), 458-465.

Liu, S., et al. (2013). Prediction of dissolved oxygen content in river crab culture based on least squares support vector regression optimized by improved particle swarm optimization. *Computers and Electronics in Agriculture*, 95, 82-91.

Liu, S., et al. (2014). A hybrid WA–CPSO-LSSVR model for dissolved oxygen content prediction in crab culture. *Engineering Applications of Artificial Intelligence*. 29, 114-124.

Lu, Y. (2016). Research on auto-encoder for deep network. Soochow University.

Luo, X. et al. (2018). Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy. *Journal of the Franklin Institute-Engineering and Applied Mathematics*, 355(4), 1945-1966.

Peng, F., et al. (2017). Multi-scale combined prediction model of dissolved oxygen based on EEMD and ELM. *IAEJ*, 26(4), 283-292.

Ta, X. and Wei. Y. (2018). Research on a dissolved oxygen prediction method for recirculating aquaculture systems based on a convolution neural network. *Computers and Electronics in Agriculture*. 145, 302-310.

Wang Y. (2014). Research on expressiveness of auto-encoder in deep learning. Harbin Institute of Technology.

Wang, H. and Bhiksha, R. (2017). On the Origin of Deep Learning.

Yan, J. X., Yu, L. J., Mao, W. W., and Cao, S. Q. (2014). Study on prediction model of dissolved oxygen about water quality monitoring system based on BP neural network. *Advanced Materials Research*. 912–914, 1407–11.