A Novel Hidden Danger Prediction Method in Cloud-Based Intelligent Industrial Production Management Using Timeliness Managing Extreme Learning Machine

Xiong Luo [1,2], Xiaona Yang [1,2], Weiping Wang [1,2]*, Xiaohui Chang [1,2], Xinyan Wang [3], Zhigang Zhao [3]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China
[3] Communication and Information Center, State Administration of Work Safety, Beijing 100013, China

Abstract: To prevent possible accidents, the study of data-driven analytics to predict hidden dangers in cloud service-based intelligent industrial production management has been the subject of increasing interest recently. A machine learning algorithm that uses timeliness managing extreme learning machine is utilized in this article to achieve the above prediction. Compared with traditional learning algorithms, extreme learning machine (ELM) exhibits high performance because of its unique feature of a high generalization capability at a fast learning speed. Timeliness managing ELM is proposed by incorporating timeliness management scheme into ELM. When using the timeliness managing ELM scheme to predict hidden dangers, newly incremental data could be added prior to the historical data to maximize the contribution of the newly incremental training data, because the incremental data may be able to contribute reasonable weights to represent the current production situation according to practical analysis of accidents in some industrial productions. Experimental results from a coal mine show that the use of timeliness managing ELM can improve the prediction accuracy of hidden dangers with better stability compared with other similar machine learning methods.

Keywords: prediction; incremental learning; extreme learning machine; cloud service

I. INTRODUCTION

The demand to increase productivity based on information technologies to improve industrial production has been increasing. With the assistance of Internet-based technologies in cloud computing, cloud service-based industrial production is becoming popular. Some computational intelligence-inspired analytical techniques have been successfully proposed in intelligent industrial production management to develop high-quality cloud applications [1-3]. Specifically, in addition to the traditional information network security and management for industrial production process, intelligent security prediction plays an important role in applications of industrial production management. In the available related works, data-driven prediction methods have been used to predict hidden dangers in industrial production and prevent possible accidents. This article focuses on the prediction of hidden dangers in industrial production under a cloud service-based industrial production environment.

In the past decades, especially in recent years, researchers from various disciplines of engineering and science have demonstrated increasing interest in problems related to prediction of the tendency of hidden dangers. Generally speaking, the number of hidden dangers and their influencing factors are nonlinearly related. In consideration of such a nonlinear relation, the neural network (NN), as an effective computational intelligence method, was developed and some learning algorithms that model that nonlinearity were designed to predict the number of hidden dangers [4]. Among the available NN-based machine learning algorithms, extreme learning machine (ELM) for single-hidden layer feedforward network (SLFN) has attracted much attention because of its quickness and simplicity [5-8]. However, the traditional ELM is not an adaptive algorithm and cannot take timeliness problem into consideration [9]. Online sequential ELM (OSELM) was proposed to learn data one by one or chunk by chunk with fixed or varying chunk sizes [10, 11]. OSELM will then discard those data that have already been trained, and only newly arrived single or chunks of samples are addressed and learned. Hence, the system can use incremental data to adjust a training data set to implement online learning. Given its unique feature, OSELM has been widely used in machine learning fields.

During the prediction of hidden dangers, the relation between enterprise management ability and the corresponding staff quality is dynamic. This relation fluctuates with time, and old data then become increasingly unlikely to change. Thus, the influence of old data may weaken. Therefore, timeliness should be taken into consideration for the newly incremental data. Newly incremental data play a critical role in implementing the prediction, where the current collection of data has a higher contribution to the model in accordance with the dynamic changes of the environment. Considering it, a timeliness managing ELM (TMELM) was proposed through the full use of the incremental data [12]. Under the adaptive timeliness weight and iteration schemes in TMELM, the incremental data can contribute reasonable weight to represent the current situation to ensure the stability of the model. Then, employing TMELM for high accuracy prediction in mine production is feasible. Nowadays, cloud computing technology makes processing and storing information in the cloud easy, as well as makes information available over the Web or other terminal ends. The proposed prediction method for hidden dangers can be considered an important scheme to support the development of high-quality cloud applications.

The rest of this article is organized as follows: Section II describes the works of hidden danger prediction and ELM. Section III proposes an implementation of TMELM for the prediction of hidden dangers. Section IV analyzes the performance of the developed algorithm. Section V concludes this article.

II. BACKGROUNDS

2.1 Prediction of hidden dangers

Generally, the prediction of hidden dangers in intelligent industrial production management helps in understanding the future development tendency of the accidents. The influence factors of some industrial safety production processes are complex, thereby presenting challenging obstacles to prediction. To address this critical challenge, the prediction and analysis for hidden dangers that were previously almost based on painstakingly handcrafted models of historical data can now be performed using data-driven advanced machine learning algorithms. Among the available optimization approaches, NN-based learning algorithm has attracted significant attention because of the fast adaptability and approximation capabilities of NN as evidenced by many related research activities, and it has been extensively applied to many prediction problems [13].

This article is concerned with studies related to the influence of enterprise safety management level and personnel quality indicators via NN-based advanced learning. The framework of the data-driven prediction for hidden dangers is shown in Figure 1.

2.2 Extreme learning machine (ELM)

ELM is a learning algorithm for SLFN. For N arbitrary distinct training samples $\{(\mathbf{x}_i, \mathbf{t}_i)\}$ ($i = 1, 2, \ldots, N$), where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^{\mathrm{T}} \in \mathrm{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^{\mathrm{T}} \in \mathrm{R}^m$, the corresponding output of this NN with L hidden neurons is mathematically modeled as (1) [14]

$$\mathbf{o}_i = \sum_{j=1}^{L} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j), \quad i = 1, \ldots, N \qquad (1)$$

where $\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]^{\mathrm{T}}$ is the weight vector that connects the j-th hidden neuron and the input neurons; $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^{\mathrm{T}}$ is the weight vector that connects the j-th hidden neuron and the output neurons; b_j is the threshold of the j-th hidden neuron; and $g(\cdot)$ is the activation function. In addition, $\mathbf{w}_j \cdot \mathbf{x}_i$ denotes the inner product of $\mathbf{w}_j$ and $\mathbf{x}_i$.

While using ELM, this SLFN is designed to approximate these N samples with zero error. Then, we have $\sum_{i=1}^{N} \|\mathbf{o}_i - \mathbf{t}_i\| = 0$, and existing $\mathbf{w}_j$, b_j, and β_j, such that

$$\mathbf{t}_i = \sum_{j=1}^{L} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j), \quad i = 1, \ldots, N \qquad (2)$$

The network structure of SLFN with ELM is illustrated in Figure 2.

The above N equations in (2) can be written compactly as

$$\mathbf{H}\beta = \mathbf{T}, \qquad (3)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ldots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L},$$

$\beta = [\beta_1^{\mathrm{T}}, \ldots, \beta_L^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{T} = [\mathbf{t}_1^{\mathrm{T}}, \ldots, \mathbf{t}_N^{\mathrm{T}}]^{\mathrm{T}}$.

Here, $\mathbf{H}$ is called the hidden layer output matrix of SLFN [15], and the i-th column of $\mathbf{H}$ is the i-th hidden node output with respect to the inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. In addition, $\mathbf{h}(\cdot)$ is called the hidden layer feature mapping. The i-th row of $\mathbf{H}$, i.e., $\mathbf{h}(\mathbf{x}_i)$, is the hidden layer feature mapping with respect to the i-th input $\mathbf{x}_i$.

According to [16], contrary to the common understanding that all the parameters of SLFN need to be adjusted, the input weights $\mathbf{w}_j$ and the first hidden layer biases b_j of ELM are not necessarily tuned, and they can be given randomly. Moreover, the orthogonal projection method can be efficiently used in ELM: $\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}$ if $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is nonsingular, where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of $\mathbf{H}$. Hence, the solution of β is

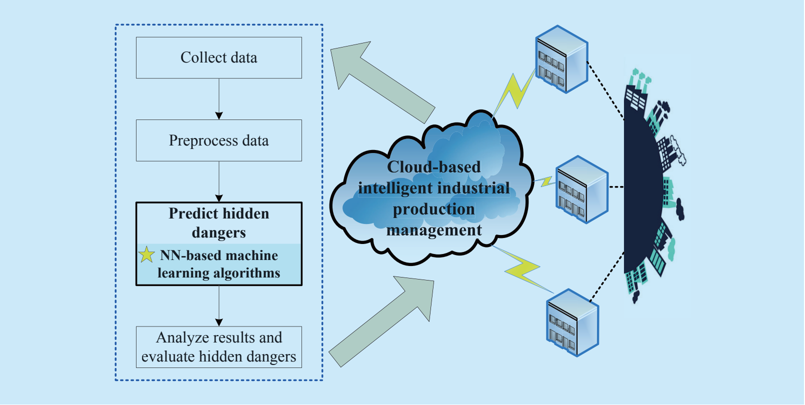

Fig.1 *Framework of prediction for hidden dangers*

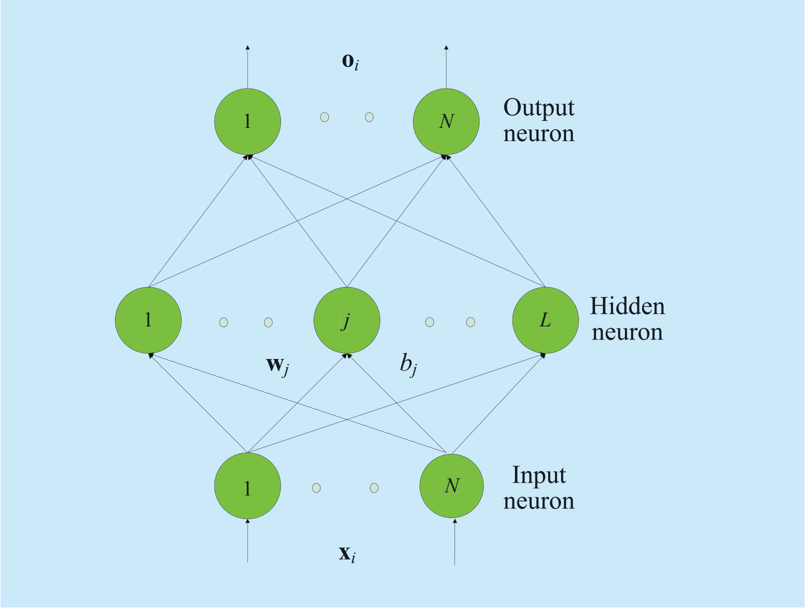

Fig.2 *Network structure of SLFN with ELM*

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T}. \tag{4}$$

The hidden layer matrix $\mathbf{H}$ remains unchanged once random values are assigned at the beginning of learning. Then, the above equation can be viewed as a linear system, in SLFN can be trained by solving this linear system. Training SLFN is simply equivalent to finding a least square solution β of the linear system. The minimum norm least square solution of (4) is unique.

2.3 Timeliness managing extreme learning machine (TMELM)

In ELM, all the samples need to be handled before being trained. However, obtaining whole samples only once in some practical applications is difficult. Thus, OSELM is proposed to deal with this issue [10]. After dividing the matrix into several parts for training in OSELM, OSELM can effectively improve the computational efforts and the learning performance. However, it does not consider the timeliness of the newly incremental data. For the newly incremental data, a timeliness analytical technique should be developed to give the new data higher priority. This real-time update mechanism can better reflect the dynamic changes of the environment, and the current collection of data has higher contribution to the system model. Thus, TMELM may be better than OSELM.

Unlike OSELM, TMELM initializes the learning through the use of a small part of the given training data. After computing the initial weights of NN, the samples are iteratively computed based on the initial weights until the newly incremental samples are collected. Unlike OSELM, under the adaptive timeliness weight and iteration scheme in TMELM, the incremental data can contribute reasonable weights to represent the current situation. TMELM can achieve higher accuracy and better stability than other machine learning methods [12].

III. TMELM-Based Prediction for Hidden Dangers

3.1 Implementation of TMELM

Given a chunk of initial training set of safety data in coal mine $\varkappa_0=\{(\mathbf{x}_i, \mathbf{t}_i)\}$ ($N_0 \geq L$; i=1, 2, …, N_0), under the ELM scheme, we can find that

$$\beta_0 = \mathbf{K}_0^{-1}\mathbf{H}_0^{\mathrm{T}}\mathbf{T}_0, \tag{5}$$

where $\mathbf{K}_0 = \mathbf{H}_0^{\mathrm{T}}\mathbf{H}_0$ and

$$\mathbf{H}_0 = [\mathbf{h}(\mathbf{x}_1), \dots, \mathbf{h}(\mathbf{x}_{N_0})]^{\mathrm{T}},\ \mathbf{T}_0 = \left[\mathbf{t}_1^{\mathrm{T}}, \dots, \mathbf{t}_{N_0}^{\mathrm{T}}\right]^{\mathrm{T}}.$$

$$\mathbf{h}(\mathbf{x}_1) = [g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1), \dots, g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L)]_{1\times L},$$

$$\mathbf{h}(\mathbf{x}_{N_0}) = [g(\mathbf{w}_1 \cdot \mathbf{x}_{N_0} + b_1), \dots, g(\mathbf{w}_L \cdot \mathbf{x}_{N_0} + b_L)]_{N_0\times L}.$$

Then, suppose that we are given another chunk of data set $\varkappa_1=\{(\mathbf{x}_i, \mathbf{t}_i)\}$ (i=N_0+1, …, N_0+N_1), where N_1 denotes the number of new samples in this data set. Considering both training data sets $\varkappa_0$ and $\varkappa_1$, the output weight β_1 becomes [10]

$$\beta_1 = \mathbf{K}_1^{-1}\begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{T}_0\\ \mathbf{T}_1\end{bmatrix}, \tag{6}$$

where

$$\mathbf{K}_1 = \begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\end{bmatrix} = \mathbf{K}_0 + \mathbf{H}_1^{\mathrm{T}}\mathbf{H}_1,$$

$$\mathbf{H}_1 = [\mathbf{h}(\mathbf{x}_{N_0+1}), \dots, \mathbf{h}(\mathbf{x}_{N_0+N_1})]^{\mathrm{T}},$$

$$\mathbf{T}_1 = \left[\mathbf{t}_{N_0+1}^{\mathrm{T}}, \dots, \mathbf{t}_{N_0+N_1}^{\mathrm{T}}\right]^{\mathrm{T}}.$$

$$\mathbf{h}(\mathbf{x}_{N_0+1}) = [g(\mathbf{w}_1 \cdot \mathbf{x}_{N_0+1} + b_1), \dots, g(\mathbf{w}_L \cdot \mathbf{x}_{N_0+1} + b_L)]_{1\times L},$$

$$\mathbf{h}(\mathbf{x}_{N_0+N_1}) = [g(\mathbf{w}_1 \cdot \mathbf{x}_{N_0+N_1} + b_1), \dots, g(\mathbf{w}_L \cdot \mathbf{x}_{N_0+N_1} + b_L)]_{N_0\times L}.$$

For sequential learning, the weight β_1 is a function of β_0, $\mathbf{K}_1$, $\mathbf{H}_1$, and $\mathbf{T}_1$. Hence,

$$\begin{aligned}\begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{T}_0\\ \mathbf{T}_1\end{bmatrix} &= \mathbf{H}_0^{\mathrm{T}}\mathbf{T}_0 + \mathbf{H}_1^{\mathrm{T}}\mathbf{T}_1\\ &= \mathbf{K}_0\mathbf{K}_0^{-1}\mathbf{H}_0^{\mathrm{T}}\mathbf{T}_0 + \mathbf{H}_1^{\mathrm{T}}\mathbf{T}_1\\ &= \mathbf{K}_1\beta_0 - \mathbf{H}_1^{\mathrm{T}}\mathbf{T}_1\beta_0 + \mathbf{H}_1^{\mathrm{T}}\mathbf{T}_1.\end{aligned} \tag{7}$$

Combining (6) and (7), the new model parameter β_1 is given by

$$\begin{aligned}\beta_1 &= \mathbf{K}_1^{-1}\begin{bmatrix}\mathbf{H}_0\\ \mathbf{H}_1\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{T}_0\\ \mathbf{T}_1\end{bmatrix}\\ &= \beta_0 + \mathbf{K}_1^{-1}\mathbf{H}_1^{\mathrm{T}}(\mathbf{T}_1 - \mathbf{H}_1\beta_0).\end{aligned} \tag{8}$$

As can be seen from (8), the calculation of β_1 is based on β_0, which will improve the computational efforts. Then, ($\mathbf{T}_1$-$\mathbf{H}_1\beta_0$) is obtained by using the old model parameters β_0, where $\mathbf{H}_1\beta_0$ is considered the error of predicting the newly added data. Thus, a penalization weight $\boldsymbol{\omega}$ is designed to adjust the contribution of data. Then, (8) can be rewritten as [12]

$$\beta_1 = \beta_0 + \omega \cdot \mathbf{K}_1^{-1}\mathbf{H}_1^{\mathrm{T}}(\mathbf{T}_1 - \mathbf{H}_1\beta_0). \quad (9)$$

The penalization weight $\boldsymbol{\omega}$ reflects the timeliness effect of newly incremental data by strengthening the effect of the new prediction model and decreasing the rate of contribution of the old prediction model. It can be expressed as follows [17]

$$\omega = 1 + 2 \cdot \exp(-|\mathrm{mean}(a_1) - \mathrm{mean}(a_2)|^{|\mathrm{var}(a_1)-\mathrm{var}(a_2)|}), \quad (10)$$

where a_1 is the newly incremental data, a_2 is the historically incremental data that is adjacent to a_1, "mean" is the function used to obtain the mean value, and "var" is the function used to obtain the variance value.

With the increase of iterative number k, for the new chunk of data set $\varkappa_k=\{(\mathbf{x}_i, \mathbf{t}_i)\}$ ($k \geqslant 2$; $i = 1 + \sum_{j=0}^{k-1} N_j, \ldots, \sum_{j=0}^{k} N_j$) where $\varkappa_k$ denotes the number of new samples at the k-th incremental learning, the learning model of (9) is redefined as [17]

$$\beta_{k(j+1)} = \beta_{k(j)} + \omega \cdot \mathbf{K}_{k+1}^{-1}\mathbf{H}_{k+1}^{\mathrm{T}}(\mathbf{T}_{k+1} - \mathbf{H}_{k+1}\beta_k). \quad (11)$$

The final parameter of β_k is obtained until it converges, i.e., $|\beta_{k(j+1)}-\beta_{k(j)}|<\boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a small value defined by users and $\beta_{k(j)}$ is the j-th iteration result for k-th incremental data. If this constraint is satisfied, then $\beta_{k(j+1)}=\beta_{k(j)}$. Finally, after ($k$+1) times incremental learning, the timeliness model parameter becomes

$$\beta_{k+1} = \beta_k + \omega \cdot \mathbf{K}_{k+1}^{-1}\mathbf{H}_{k+1}^{\mathrm{T}}(\mathbf{T}_{k+1} - \mathbf{H}_{k+1}\beta_k). \quad (12)$$

3.2 Algorithm framework

TMELM is employed to predict the tendency of hidden dangers by incorporating timeliness management scheme into OSELM. Its flowchart is shown in Figure 3.

When running the above algorithm, we first choose the number of hidden dangers and its influence factors in industrial production as the input for SLFN. Then, we conduct TMELM to analyze and model the corresponding relationship between the number of hidden dangers and index data. Through the use of TMELM, we can predict the number of hidden dangers in the future with high accuracy.

IV. Experimental Results and Discussions

4.1 Experimental description

The idea of TMELM is to strengthen the recent production data and weaken the older data. To test the effectiveness of this algorithm, we evaluate the performance of TMELM through predicted results and the successful predic-

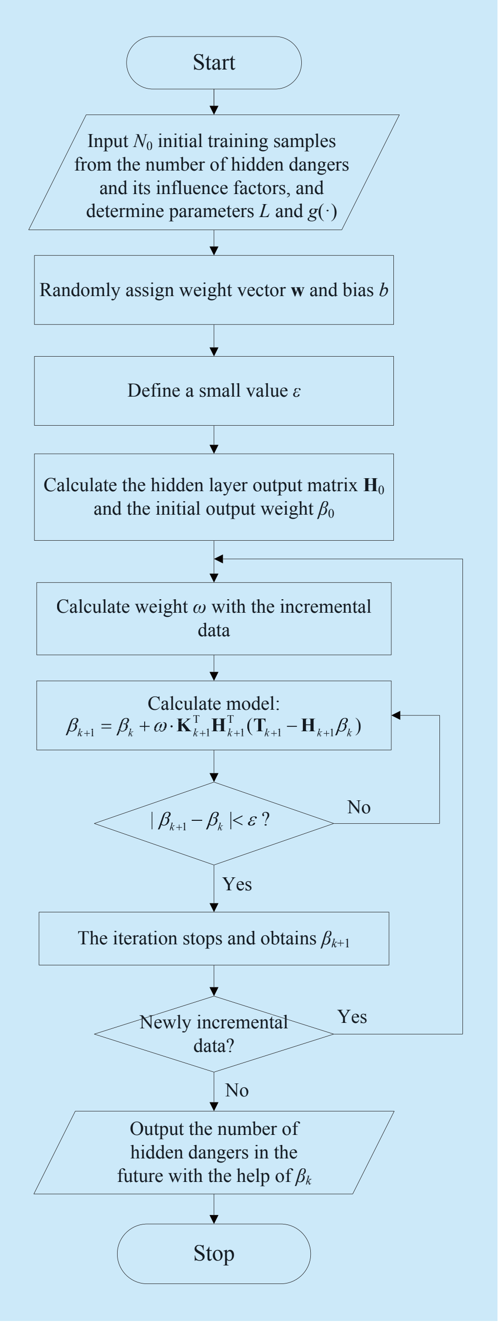

Fig.3 *Flowchart of the algorithm*

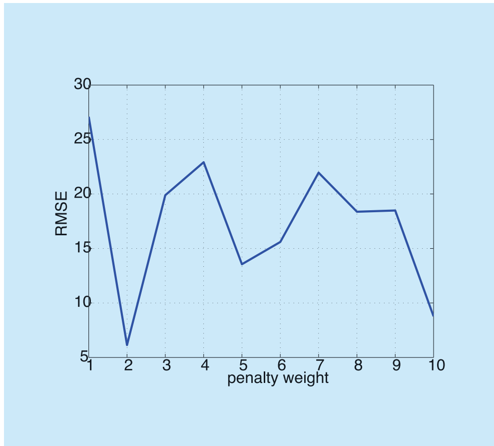

Fig.4 *Experiment of penalty weight*

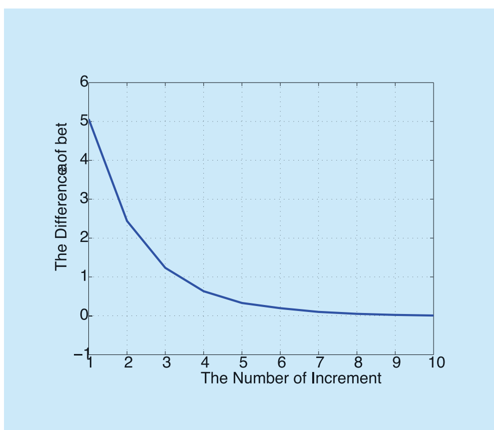

Fig.5 *Difference between model parameters $\beta_{k(j+1)}$ and $\beta_{k(j)}$*

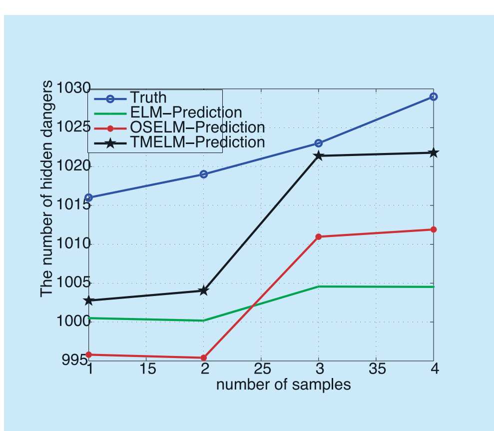

Fig.6 *Comparison among ELM, OSELM, and TMELM*

tion rate. The performance of TMELM is also evaluated by comparing different traditional schemes, including ELM- and OSELM-based methods. All the experiments are conducted in MATLAB R2010b computing environment with Intel® Core^{TM} i3 and 2.13 GHz CPU.

The accuracy is evaluated on the basis of prediction error. We use the root-mean-square error (RMSE) to measure the prediction error between the predicted value of hidden dangers and the actual value recorded in the past years. A large prediction error corresponds to poor estimation accuracy. The RMSE is defined as

$$\mathrm{RMSE}(\mathbf{y}, \mathbf{y}') = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{13}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$, $\mathbf{y}' = [y_1', y_2', \ldots, y_n']^T$, y_i and y_i' denote the actual value and the predicted value, respectively.

The actual data in this experiment were collected in the coal mine of Beijing within 24 months from 2011 to 2012. Details of those data can be found in [13]. The number of hidden dangers are analyzed and predicted to determine the tendency of the hidden dangers to occur. In our experiment, the activation function is set to $g(x)=1/(1+e^{-x})$. The number of nodes in the hidden layer of SLFN is set to L=5. We choose the number of initial samples as 10. Otherwise, the whole data for 24 months are divided into two parts. The first 20 months are used for training, and the rest of the data are subjected to testing. In this article, we set a penalty weight $\boldsymbol{\omega}$ to change the influence of the new data on prediction mechanism. During the iteration, the termination condition threshold is set to $\boldsymbol{\varepsilon}$=0.01.

Theoretically, a large weight corresponds to fast convergence. However, the large weight makes the model suitable for the current incremental data. If the distribution of testing error is different from incremental data, then large testing error will occur. Hence, the weight should be within a certain range. In our experiment, the penalty weight is unchanged. After testing, with the increase of weight $\boldsymbol{\omega}$ in [1, 10] using (10), the RMSE first comes down and then goes up, as shown in Figure 4. Thus, we choose the penalty weight as ω=2.

To confirm the effectiveness of penalty weight ω we set here, we show the difference between model parameters $\beta_{k(j+1)}$ and $\beta_{k(j)}$ when the iteration number increases in Figure 5. This figure shows a difference of model parameters before 6 times iteration, and then it becomes stable with the increase of iteration number. This finding may confirm that with the weight that has been set, the learning scheme for the model parameters in (11) can guarantee the convergence of this ELM-based learning model.

4.2 Experimental results

With the actual mine data, some experiments are conducted to evaluate the performance of TMELM algorithm in predicting hidden dangers. Moreover, experimental comparisons are implemented between TMELM and other similar algorithms, including ELM and OS-ELM. Typical experimental results obtained after running those algorithms many times are shown in Figures 6, 7, and 8.

Figure 6 shows that TMELM using timeliness mechanism can follow the development tendency of the samples more closely than the other two algorithms. This result is obtained because TMELM reinforces the recent new incremental samples and weakens the historical samples. Moreover, its penalty weight scheme reflects the timeliness of samples while ensuring that the new training samples have a greater contribution to the system model. The adaptive iteration scheme guarantees the stability of the system. OSELM produces a large deviation because it imposes the same contribution for all sample data. Thus, OSELM cannot distinguish the importance between new incremental data and historical data. Therefore, TMELM is the best one among the three algorithms in most cases, and it is more stable because more training samples are used in numerous independent tests. Thus, considering prediction accuracy and stability, TMELM may be better than the other two algorithms in tracking the tendency.

In this experiment, we calculate the prediction performance for hidden dangers by using RMSE. Figure 7 shows the testing error with different numbers of incremental samples. Obviously, with the increase of increment numbers, the training data set increases, thereby covering more possible situations. Therefore, the testing errors for all these algorithms are reduced. Unlike OSELM and ELM, TMELM achieves a small testing error quickly and remains stable through the use of penalty weight and iteration schemes. Compared with TMELM, the testing errors of ELM and OS-ELM are slightly poorer. Therefore, from this point of view, TMELM can better predict hidden dangers in practical applications.

Figure 8 shows the training time consumption among ELM, OSELM, and TMELM.

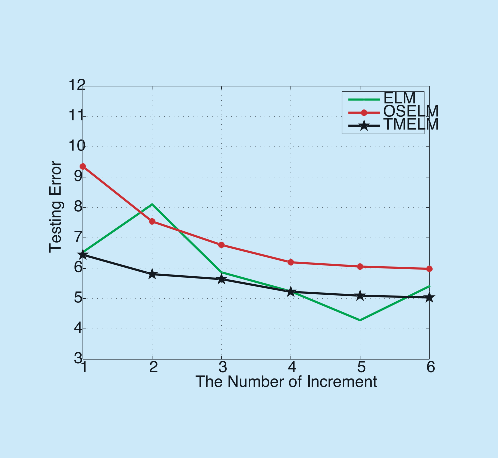

Fig.7 *Performance comparison in testing error*

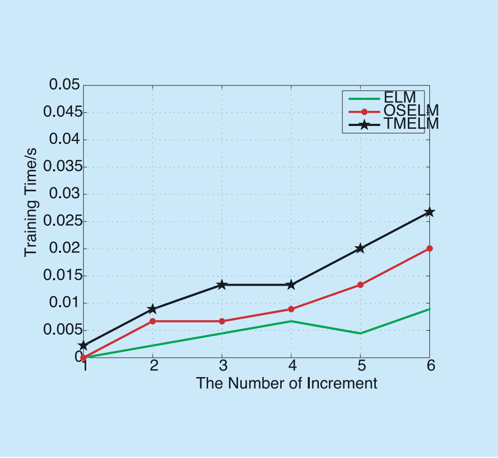

Fig.8 *Performance comparison in training time*

With the increase of the training data, the training time increases slowly. The learning speed of ELM is known to be fast, and ELM can train SLFN much faster than classical learning algorithms. A comparison with OS-ELM and TMELM indicates that the basic ELM outperforms the best. However, given its unique features of penalty weight and adaptive iteration, TMELM consumes more time to guarantee stability and convergence.

V. Conclusion

In cloud-based intelligent industrial production management, the prediction of hidden dangers plays an important role in preventing possible accidents in enterprises. To analyze hidden dangers and related influence factors, a sequential learning algorithm TMELM under the framework of OSELM is developed to predict the number of hidden dangers. Unlike ELM and OSELM, TMELM is designed through the use of timeliness analytical technique in the ELM approach. After adding the penalty weight and adaptive iteration strategies in the scheme of TMELM, the weight can reflect the timeliness effectiveness in accordance with the difference between the newly incremental data and the historical data. Furthermore, the adaptive iteration strategy could improve stability and convergence of the system. Therefore, TMELM may be an effective learning algorithm for SLFN when dealing with the issue of timeliness sequence. Experiments are conducted using actual data obtained from a mine in Beijing in 2011 and 2012. Numerical results indicate that the performance of TMELM is superior to that of other algorithms in terms of prediction accuracy and stability. The TMELM-based prediction algorithm is expected to improve production safety by enabling users to take effective measures to rectify hidden dangers.

Acknowledgement

We gratefully acknowledge the anonymous reviewers who read the drafts and offered many helpful suggestions. This work was partially supported by the National Key Technologies R&D Program of China under Grant No. 2015BAK38B01, the National Natural Science Foundation of China under Grant Nos. 61174103 and 61272357, and the Fundamental Research Funds for the Central Universities under Grant No. 06500025.

References

[1] J. Dong, H. Wang, S. Cheng, "Energy-performance tradeoffs in IaaS cloud with virtual machine scheduling", *China Communications*, vol.12, no.2, pp 155-166, February, 2015.

[2] D. Zhang, S. Zhao, L.T. Yang, M. Chen, Y. Wang, J. Liu, "NextMe: localization using cellular traces in Internet of Things", *IEEE Transactions on Industrial Informatics*, vol.11, no.2, pp 302-312, April, 2015.

[3] C. Wang, K. Ren, J. Wang, "Secure optimization computation outsourcing in cloud computing: A case study of linear programming", *IEEE Transactions on Computers*, vol.65, no.1, pp 216-229, January, 2016.

[4] K.I. Funahashi, "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, vol.2, no.3, pp 183-192, June, 1989.

[5] G. Huang, G.B. Huang, S. Song, K. You, "Trends in extreme learning machines: A review", *Neural Networks*, vol.61, pp 32-48, January, 2015.

[6] X. Luo, X.H. Chang, H. Liu, "A Taylor based localization algorithm for wireless sensor network using extreme learning machine", *IEICE Transactions on Information and Systems*, vol.E97D, no.10, pp 2652-2659, October, 2014.

[7] X. Luo, X.H. Chang, "A novel data fusion scheme using gray model and extreme learning machine in wireless sensor networks", *International Journal of Control, Automation, and Systems*, vol.13, no.3, pp 539-546, June, 2015.

[8] X. Luo, X.H. Chang, X.J. Ban, "Regression and classification using extreme learning machine based on L1-norm and L2-norm", *Neurocomputing*, vol.174, pp 179-186, January, 2016.

[9] S.Y. Xu, J. Lam, "A new approach to exponential stability analysis of neural networks with time-varying delays", *Neural Networks*, vol.19, no.1, pp 76-83, January, 2006.

[10] G.B. Huang, N.Y. Liang, H.J. Rong, P. Saratchandran, N. Sundararajan, "On-line sequential extreme learning machine", *Proceedings of the IASTED International Conference on Computational Intelligence*, Calgary, Canada, pp 232-237, 2005.

[11] Y. Lan, Y.C. Soh, G.B. Huang, "A constructive enhancement for online sequential extreme learn-

ing machine", *Proceedings of the International Joint Conference on Neural Networks*, Atlanta, USA, pp 1708-1713, 2009.

[12] J.F. Liu, Y. Gu, Y.Q. Chen, Y.S. Cao, "Incremental localization in WLAN environment with timeliness management", *Chinese Journal of Computers*, vol.36, no.7, pp 1448-1455, July, 2013.

[13] X.Y. Wang, *The Study of Evaluation and Prediction on the Governance Capability for Safety Hidden Danger of Coal Mine*. Beijing, China: China University of Mining and Technology, 2013.

[14] G.B. Huang, X. Ding, H. Zhou, "Optimization method based extreme learning machine for classification", *Neurocomputing*, vol.74, no.1-3, pp 155-163, December, 2010.

[15] G.B. Huang, H.A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions", *IEEE Transactions on Neural Networks*, vol.9, no.1, pp 224-229, January, 1998.

[16] G.B. Huang, H. Zhou, X. Ding, "Extreme learning machine for regression and multiclass classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.42, no.2, pp 513-529, April, 2012.

[17] Y. Gu, J.F. Liu, Y.Q. Chen, X.L. Jiang, H.C. Yu, "TOSELM: Timeliness online sequential extreme learning machine", *Neurocomputing*, vol.128, pp 119-127, March, 2014.

Biographies

Xiong Luo, received his Ph.D. from Central South University, China, in 2004. He currently works as an associate professor in the University of Science and Technology Beijing. His current research interests include machine learning, cyber-physical systems, and computational intelligence. Email: xluo@ustb.edu.cn.

Xiaona Yang, is currently working toward her master's degree at the University of Science and Technology Beijing. Her research interests include machine learning and cyber-physical systems.

Weiping Wang, the corresponding author, email: shiya666888@126.com, received her Ph.D. from Beijing University of Posts and Telecommunications, China, in 2015. She currently works as an associate professor in the University of Science and Technology Beijing. Her research interests include neural networks and computational intelligence.

Xiaohui Chang, received her master's degree from the University of Science and Technology Beijing, China, in 2016. Her research interests include machine learning and cloud computing.

Xinyan Wang, currently works as an engineer at the Communication and Information Center of the State Administration of Work Safety, China. Her research interests include computational intelligence and machine learning.

Zhigang Zhao, currently works as an engineer at the Communication and Information Center of the State Administration of Work Safety, China. His research interests include cloud computing and communication engineering.